基于多种支撑点的度量空间离群检测算法

许红龙[1),2)] 唐 颂[3)] 毛 睿[2)] 沈 婧[2)] 刘 刚[2)] 陈国良[2)]

[1)](佛山科学技术学院数学与大数据学院 广东 佛山 528000)

[2)](深圳大学计算机与软件学院广东省普及型高性能计算机重点实验室 广东 深圳 518060)

[3)](南开大学化学学院 天津 300071)

摘 要 大数据的价值实现,归根到底还是依赖于数据挖掘技术.而在很多领域中,海量数据的非常规模式往往更具分析价值.离群检测,也叫异常检测,是用于挖掘海量数据中非常规模式的一项关键技术,广泛应用于网络入侵检测、公共卫生、医疗监控等领域.基于索引的离群检测算法通常具有较高的检测速度,然而现有的大多数基于索引的检测算法并非完全基于距离,导致通用性降低.较高的抽象能力使得度量空间具有比多维空间更广泛的适用范围,在其基础上设计的算法具有更高的通用性.而最新的度量空间基于索引的离群检测算法 iORCA 算法通过随机选取支撑点,基于数据到单支撑点的距离建立索引,并应用终止规则(Stopping rule)以期提前结束离群检测并得到正确的结果,多数情况下该机制起到加快检测速度的重要作用.然而 iORCA 算法未提供支撑点选取算法导致检测结果不稳定,且未能充分利用距离三角不等性减少距离计算次数.针对这些问题,文中指出基于距离的离群点定义应结合使用完全基于距离的离群检测算法,以确保算法的通用性,由此提出了度量空间离群检测的概念.在此基础上明确了支撑点选取的两大目标,即边缘支撑点和密集支撑点,并提出基于多种支撑点的度量空间离群检测算法 VPOD.考虑到两个支撑点选取目标难以同时达到,VPOD 算法分别予以选取,在近似的密集区域选取支撑点,即密集支撑点,对应使用终止规则,然后用 FFT(Farthest-First Traversal)算法另选取若干支撑点,即边缘支撑点,与数据集计算距离而形成支撑点空间,利用距离三角不等性,使距离计算次数显著减少,从而提高检测速度.实验表明该算法能在可接受的时间范围内建立索引,并能高效检测离群点,加速比达 2.05,最高达 3.54,距离计算次数平均减少 51.14%,最高达 89.46%,同时保持对多种常见的基于距离的离群点定义的兼容.

关键词 离群检测;度量空间;索引;支撑点选取;三角不等性

中图法分类号 TP311 **DOI 号** 10.11897/SP.J.1016.2017.02839

Various Pivots Based Outlier Detection Algorithm in Metric Space

XU Hong-Long[1),2)] TANG Song[3)] MAO Rui[2)] SHEN Jing[2)] LIU Gang[2)] CHEN Guo-Liang[2)]

[1)](*School of Mathematics and Big Data, Foshan University, Foshan, Guangdong* 528000)

[2)](*Guangdong Province Key Laboratory of Popular High Performance Computers, College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, Guangdong* 518060)

[3)](*College of Chemistry, Nankai University, Tianjin* 300071)

Abstract The realization of the value of big data, is still dependent on data mining technology in the final analysis. In many areas, the unconventional model of massive data is usually more valuable for analysis. Outlier Detection, also known as Anomaly Detection, is a key technique to discover abnormal patterns from mass data. Outlier detection techniques have been widely applied

收稿日期:2016-12-13;在线出版日期:2017-04-18.本课题得到国家"八六三"高技术研究发展计划项目基金(2015AA015305)、国家自然科学基金委-广东联合项目(U1301252,U1501254)、广东省重点实验室建设情况考评项目(2017B030314073)、广东省自然科学基金(2015A030313636)、深圳市科技计划项目(CXZZ20140418182638764)资助.**许红龙**,男,1986 年生,博士,讲师,中国计算机学会(CCF)会员,主要研究方向为数据挖掘、大数据.E-mail:longer597@163.com.**唐 颂**,男,1993 年生,双学士,主要研究方向为常微分方程、度量空间.**毛 睿**(通信作者),男,1975 年生,博士,教授,主要研究领域为度量空间索引、大数据管理分析.E-mail:mao@szu.edu.cn.**沈 婧**,女,1993 年生,硕士研究生,主要研究方向为复杂网络.**刘 刚**(通信作者),男,1977 年生,博士,讲师,中国计算机学会(CCF)会员,主要研究方向为并行算法、片上网络.E-mail:gliu@szu.edu.cn.**陈国良**,男,1938 年生,教授,博士生导师,中国科学院院士,主要研究领域为高性能计算.

in many fields, such as network intrusion detection, public health and medical monitoring. Index based outlier detection algorithm usually has higher detection speed. However, most existing index based outlier detection algorithms are not completely based on distance, resulting in the weakening of universal property. In other words, these algorithms can be only applied in multidimensional dataset. Metric space is a set with a distance function which satisfies the distance triangle inequality. Instead of domain-specific information, the requirement to apply metric space is so simple that only need to define the distance function. Because of better universal abstraction, metric space has a wider range of application than the multidimensional space, and the algorithm designed on the basis of it is more universal. The latest index-based outlier detection algorithm in metric space, namely iORCA, randomly selects a single pivot and builds index upon the distances from data to it, then the algorithm can terminate ahead of time with the correct result in use of stopping rule. In most cases, this mechanism is effective and can save detection time. However, the detection result of iORCA is not stable because of its lack of pivot selection method. Further, it does not exploit Triangle Inequality to reduce distance calculation times. Focusing on these problems, in this paper, we pointed out that the distance based outlier definition should be applied together with completely distance based outlier detection algorithm, in order to guarantee the universal property, and proposed the definition of Outlier Detection in Metric Space. Further, we well defined the two goals of pivot selection, which are border pivot and dense pivot. Based on these goals, Various Pivots based Outlier Detection (VPOD) algorithm is proposed. In consideration of difficulty to achieve the two goals of pivot selection, VPOD selects the two kinds of pivots separately. On one hand, VPOD selects a single pivot in approximately dense region, which is dense pivot, related to the application of stopping rule. On the other hand, several pivots will be selected by Farthest-First Traversal algorithm, which are border pivots. Then VPOD will calculate the distance of all the objects of dataset to these pivots, in order to converts the dataset from metric space to a pivot space. With the help of distance triangle inequality, the distance calculation times can be significantly reduced, with the result of higher detection speed. Experimental results show that VPOD can build the metric space index in acceptable time, and achieves a 2.05 speed up over iORCA on average, and in certain cases, up to 3.54. The distance calculation times are reduced by 51.14% on average, and up to 89.46%. In addition, VPOD has not lost the compatibility to the several most popular distance based outlier definitions.

Keywords outlier detection; metric space; index; pivot selection; triangle inequality

1 引 言

“数据丰富,但信息贫乏(Data Rich & Information Poor)”,体量日益膨胀的大数据尤甚.人类的各种社会活动和各类设备采集、产生的海量数据,其中有用的信息可能微乎其微.这些数据若未能得到及时处理,更会逐步累积而占用大量的存储设备,久而久之成为“食之无肉,弃之可惜”的鸡肋.数据挖掘技术的出现,使这一问题迎刃而解.中国科学院院士梅宏教授指出,真正的大数据应用应该体现在数据挖掘的深度.聚类、分类、关联分析等技术使人们得以获取隐含在海量数据背后的常规模式.然而,“一个人的噪声可能是另一个人的信号”[1],海量数据中的非常规模式,有时往往具有更为惊人的价值.

离群检测,就是用于从海量数据中发现非常规模式的数据挖掘技术.学术界影响最为深远的定义源自 Hawkins,“离群点是数据集中与众不同的点,其表现与其它点如此不同,以至于使人怀疑这些并非随机的偏差,而是由另外一种完全不同的机制所产生的”[2].换言之,离群点就是海量数据中极少数与主流数据显著不同的数据.在很多领域中,这些

“极少数”数据往往比主流数据更具分析价值.目前，离群检测已被广泛应用于网络入侵检测[3-4]、公共卫生[5]、医疗监控[6]等诸多领域中.

一般而言，离群检测算法可以分为五类——基于统计的离群检测算法、基于距离的离群检测算法、基于密度的离群检测算法、基于深度的离群检测算法及基于偏离的离群检测算法[7].如图 1，完全基于距离的离群检测算法具有良好的通用性，结合使用基于距离的离群点定义，可称为度量空间离群检测，在应对大数据多样性挑战方面具有独特的优势.

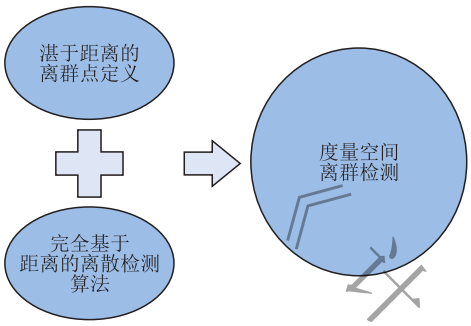


图 1　度量空间离群检测

所谓度量空间，也即距离空间.设 S 是有限非空的数据集合，$dist$ 是定义在 S 上的距离函数，且具有如下 3 个性质：

(1) 正定性.对于任意 $x,y\in S$，$dist(x,y)\geq 0$，并且 $dist(x,y)=0\leftrightarrow x=y$.

(2) 对称性.对于任意 $x,y\in S$，$dist(x,y)=dist(y,x)$.

(3) 三角不等性.对于任意 $x,y,z\in S$，$dist(x,y)+dist(y,z)\geq dist(x,z)$.

那么度量空间可以定义为一个二元组 $(S,dist)$.

由此可知，度量空间对数据类型的要求非常低，只要在数据集上能定义出符合正定性、对称性、三角不等性的距离函数，即可成为度量空间.较低的要求，使度量空间成为一种覆盖范围非常广泛的数据类型抽象.很多难以抽象至多维空间的数据类型，只要定义出相应的距离函数，即可很好地抽象至度量空间.例如，多维空间和欧几里德距离就构成了一个度量空间.多维数据空间可以看作是一个具有坐标信息的特殊的度量空间，它们之间的区别如表 1 所示.

表 1　多维数据空间与度量空间的主要区别

空间类型	坐标信息	所用距离	维度概念	适用范围
多维空间	有	L 族距离等	有	较小
度量空间	无	只需满足距离的三个特性	无	非常广泛

下面以图像、文本和蛋白质为例说明度量空间与多维数据空间的不同之处.

(1) 图像.将每张图片用结构向量(3 维)、纹理向量(15 维)和颜色向量(48 维)3 个特征向量表示.对于距离函数，可以先用欧几里德距离分别求两张图片结构向量、纹理向量之间的距离，再用曼哈顿距离求颜色向量间的距离.最后将这 3 个距离取平均值即可得两张图片之间的最终距离.可以证明，该距离函数满足度量空间距离三特性[8]，因而构成了一个度量空间.

(2) 文本.距离函数常用编辑距离[9]，即由一个字符串转成另一个字符串所需的最少编辑操作次数.而编辑操作可以是把一个字符替换成另一个字符，插入一个字符，删除一个字符等 3 种操作.显然，编辑距离可用来衡量两个字符串的相似度，其值越小，就表示相似度越大.进而，将插入和删除的代价设为 1，替换的代价设为 2，由此得到的距离函数满足度量空间距离三特性，故而构成一个度量空间.长度分别为 i、j 的字符串 a、b 的编辑距离表达式如下：

$$lev_{(a,b)}(i,j)=\begin{cases}\max(i,j), & \text{若 } \min(i,j)=0\\ \min\begin{cases}lev_{a,b}(i-1,j)+1\\ lev_{a,b}(i,j-1)+1\\ lev_{a,b}(i-1,j-1)+1_{(a_i\neq b_j)}\end{cases}, & \text{否则}\end{cases}\tag{1}$$

(3) 蛋白质序列.距离函数常用比对(alignment)距离[10-11].比对距离相当于在编辑距离的基础上加上权重，形成权重矩阵.与文本数据一样，蛋白质序列难以映射到多维数据空间，但其比对距离满足度量空间距离三特性[11]，从而可抽象至度量空间.

上述 3 种数据类型都难以抽象到多维数据空间.而度量空间极低的要求，使得大多数数据类型都可定义出相应的距离函数，从而符合度量空间的定义.退一步来说，即使对于无法直接抽象至度量空间的数据类型，也有相应的一些解决方案.一是使用一些数学方法将非度量距离函数转换为度量距离函数.二是使用距离函数仅满足半度量、伪度量等部分度量性质的通用方法.三是设法从距离函数推导出一些有用结论，再应用于设计通用方法.

基于度量空间的算法只须用户根据数据类型提供满足度量空间距离三特性的距离函数，并且数据类型的表达以及相应的距离函数具体实现都独立于度量空间算法，故同一度量空间算法可以应用于不

同的数据类型,因而具备了更广泛的适用范围.例如度量空间索引算法,或本文提出的度量空间离群检测算法等,既可应用于各类多维数据,也可直接应用于包括上述图像、文本和蛋白质序列等在内的复杂数据类型,且算法本身无须修改.

相对于度量空间来说,多维空间未能对数据类型实现很好的抽象,据此设计的相关算法,需要使用距离以外的信息(维度信息),而对于其它难以抽象至多维空间的数据类型则不兼容.例如面向多维空间研发的离群检测算法,只能用于检测多维数据集的离群点,不能直接用于处理诸如上述图像、文本或蛋白质序列等其它复杂数据类型.

目前,度量空间因其在抽象数据类型上的巨大优势,已被重点应用于对数据类型较为敏感的数据管理分析领域,其中度量空间索引技术已较为成熟.为应对被检测的数据类型多样性的挑战,本文提出度量空间离群检测的概念.实际上,已有的部分离群检测算法符合或在一定程度上符合度量空间离群检测的定义.其中,基于索引的检测算法在速度方面具有很大的优势.然而现有的基于索引的检测算法部分应用了度量空间(距离)之外的信息,或者在只用距离信息的情况下,却未提供支撑点选取算法,未能更好地应用距离三角不等性等,使其检测速度未能进一步提升.

针对这些问题,本文将探讨基于多种支撑点的离群检测算法,同时应用多种支撑点选取算法,进一步拓展距离三角不等性的应用范围,提高检测速度.本文的主要贡献总结如下:

(1)指出基于距离的离群点定义应结合使用完全基于距离的离群检测算法,以确保算法通用性,据此提出度量空间离群检测的概念.

(2)提出支撑点选取的两个目标,分别是密集区域、边缘区域.

(3)指出两个支撑点选取目标难以兼顾,应区别对待,针对其选取目标,使用不同算法选取两类支撑点.

(4)对不同类别的支撑点结合使用不同的剪枝规则,提高剪枝效率,据此提出基于多种支撑点的度量空间离群检测算法.

本文第 2 节将简述基于索引的度量空间离群检测及支撑点选取相关知识;第 3 节分析支撑点选取目标、范围以及给出支撑点选取算法和相应的剪枝规则;第 4 节介绍基于多种支撑点选取的离群检测算法;第 5 节给出实验结果,并加以分析;最后第 6 节对本文工作进行总结,并指出下一步研究方向.

2 相关工作

本节首先阐述基于索引的离群检测算法常用的离群点定义,然后介绍基于索引的离群检测算法,最后对度量空间常用的支撑点选取方法进行简要介绍并作分析.

基于索引的度量空间离群检测应用的离群点定义,即基于距离的离群点定义,主要有 3 种:DB(p,d)离群点[12-13]、k 距离离群点[14]以及 k 最近邻距离和离群点[15].以下简单介绍这 3 种离群点定义,且为简便起见,不区分"点"与"对象"的叫法.

DB(p,d)离群点,就是这样的点,在数据集 T 中,至少有比例为 p 的点与其距离大于 d 的对象.此外有另一个定义与之等价,即与其距离小于或等于 R 的点数量不超过 k 个的点,就是离群点,可称 k-R 离群点.显然,按照该定义,一个点非此即彼,不是正常点,就是离群点,而不可能介于二者之间.

k 距离离群点,即将数据集中每个对象的第 k 最近邻的距离值作为离群度,依离群度排序得到最大的 n 个对象,就是 TOP n 离群点.

k 最近邻和离群点,就是将数据集中每个对象的 k 最近邻的距离值之和作为离群度,依离群度排序得到最大的 n 个对象,就是 TOP n 离群点.与之等价的另一定义是 k 最近邻平均距离离群点,以 k 最近邻的平均距离为离群度,排序得到 TOP n 离群点.

显然,与 k-R 离群点不同的是,k 距离离群点与 k 最近邻和离群点都有离群度的概念,可以排序,也就有了"谁更离群,谁更正常"的意义.

基于索引的离群检测算法可追溯至 1998 年,Knorr 等人首先提出基于距离的离群点定义[12],随后总结提出了 3 类基于距离的检测算法——基于索引的算法(Index-based)、嵌套循环算法(Nested-loop)和基于单元的算法(Cell-based)[13].

值得一提的是,Knorr 等人并未提出基于索引的具体检测算法,认为利用多维索引结构,例如 R 树[16]、KD 树[17]和 X 树[18]等索引算法的变种,能有效检测基于距离的离群点.由于使用了多维索引结构,这些基于索引的离群检测算法应用范围也就局限于多维数据.此外,Knorr 等人认为数据集的索引

结构已经提前建立，因此可直接应用基于这些索引的检测算法.在不考虑索引结构建立的计算复杂度前提下，基于索引的算法具有较高效率和良好的扩展性.

约翰霍普金斯大学(The Johns Hopkins University)的 Chaudhar 等人为了加快大规模数据集的离群检测速度，提出了基于 KD 树索引的离群点检测算法[19].考虑到构建 KD 树索引的开销，算法具有 $O(nk)$的时间复杂度(其中 n 为数据集规模，k 为数据对象维数).然而，该算法虽然具有较低的时间复杂度，但在构建及应用 KD 树索引使用了度量空间以外的信息(维度信息)，同样仅适用于多维数据集，通用性较低.

此后，北达科他州立大学(North Dakota State University)的 Ren 等人基于 P 树索引对数据集进行本地裁剪，然后在剩下的子集中进行离群检测，提出一种垂直的基于 P 树索引的离群点检测算法[20].然而，与基于 KD 树的算法一样，基于 P 树的算法用到了维度信息，并不是完全基于距离的离群检测算法，没有度量空间离群检测在通用性方面的优势.

Wang 等人和 Pillutla 等人分别就不同的离群点定义，提出了基于局部敏感哈希(LSH)的离群检测算法[21-22].他们利用 LSH 技术的特点——在对相似点在哈希之后，仍能在一定程度上相似，并具有一定的概率保证.正因如此，与本文所提精确检测算法不同的是，这些算法实际上只是近似检测算法.

王习特等人针对分布式环境下的离群检测问题，设计了一种基于空间的数据划分算法，平衡节点工作负载，并在每个计算节点本地使用 R 树索引进行批量过滤，快速获得离群点候选集，最后通过节点间通信计算得到最终结果[23].R 树索引在该算法中起到非常重要的加速作用，但只适用于多维空间，对于度量空间，由于没有维度概念，只有距离信息，无法使用该算法.

Bay 等人于 2003 年提出的 ORCA 算法[24](因其实现程序 orca 得名)，分块检测数据集，并应用简单的剪枝规则——一旦计算得对象的当前离群度低于 TOP n 离群点阈值，则该对象不再可能成为离群点(因为随着 k 最近邻搜索的进行，离群度只会不变或者变小，而不可能增大)，获得近似线性的检测速度，成为离群检测领域的 state-of-art 算法.在此基础上，Bhaduri 于 2011 年提出了 iORCA(indexing ORCA)算法[25]，在数据集中随机选取支撑点，然后计算所有对象与支撑点的距离并按降序排序而建立索引，在检测离群点时，按与被检对象在索引中的距离从近到远的顺序“螺旋式”搜索 k 最近邻，并应用终止规则提前结束检测过程且获得正确的结果.iORCA 算法具有索引时间开销小，检测效率高的优点，成为基于支撑点索引技术的度量空间离群检测算法的典范，是与本文所提算法最接近且最新的算法.

在度量空间支撑点选取方面，面向度量空间索引的支撑点选取算法研究已经比较成熟.目前较为常用的算法有 FFT 算法[26]、HF 算法[27]、基于划分边界邻域分析的方法[28]等.这些算法非常适用于度量空间索引，但对于离群检测来说，因这些算法仅在边缘等区域选取支撑点，所以并不完全满足离群检测的支撑点选取目标.

而针对离群检测的支撑点选取算法目前刚刚起步.本领域代表性算法 iORCA 根据离群点数量在数据集中比例较低的特点，随机选取支撑点，因而具有较大的概率选取到数量比例较高的正常点作为支撑点.但如果不幸选取到离群点作为支撑点，其算法性能将急剧下降，起不到算法所期望的“尽量先检测离群点，以尽快提高离群点阈值”的作用.此外，iORCA 算法的随机选取方法是针对单支撑点设计的，尽管它可直接扩展到多支撑点，但性能同样存在着较大的不确定性.

3　支撑点选取算法

本节先介绍算法所用的相异支撑点选取算法，再讲述使用这些支撑点建立索引的过程，以及与此对应的剪枝规则.

3.1　支撑点选取算法

本节首先分析支撑点选取的目标和范围，再阐述两类(相异)支撑点选取算法，最后分析密集支撑点选取算法采用的数据集划分方法.

3.1.1　离群检测的支撑点选取目标

包括离群检测在内的很多数据挖掘算法高度依赖搜索过程，而索引对加快搜索速度来说具有重要意义.对于度量空间离群检测来说，支撑点的选取对于索引的构建和离群检测本身都尤为重要.良好的支撑点有利于区分数据集中的对象，减少支撑点空间新坐标重叠现象，也能更好地应用距离三角不等性排除非离群点.现分析支撑点选取的目标.

(1)边缘支撑点

如图 2 所示单位正方形区域内均匀分布的二维数据集,含 1000 个点.显然,如图 3(a)以(1/3,1/3)与(2/3,2/3)为支撑点,即支撑点选取于该区域内部,将导致很多点映射至支撑点空间的新坐标重叠,而选取于边缘位置则能在一定程度上避免这个问题.但即使在边缘区域,不同的选取位置,结果也不尽相同.如图 3(b)与图 3(c),分别选取该正方形区域的相邻的两个顶点(0,0)与(0,1)和相对的两个顶点(0,0)与(1,1)作为支撑点,把数据集的每个对象按照其到两个支撑点的距离映射到支撑点空间,其中 x 坐标是数据到第一个支撑点的欧几里德距离,y 坐标是数据到第二个支撑点的欧几里德距离.从图 3(c)中可以看出,选取相对顶点作为支撑点的时候,仍然出现较多的新坐标重叠,导致数据投影后分布范围较小,而且稀疏不一.如图 3(b),选取相邻的顶点作为支撑点的时候,数据投影后分布范围较广,且比较均匀,因此更容易区分,有望获得较好的搜索性能.由此可知支撑点选取于某些边缘位置有利于区分数据,即支撑点选取目标之一为"边缘支撑点".

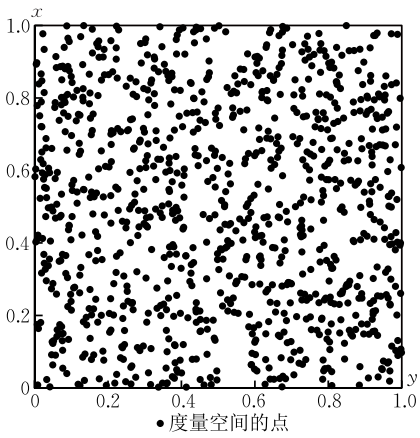
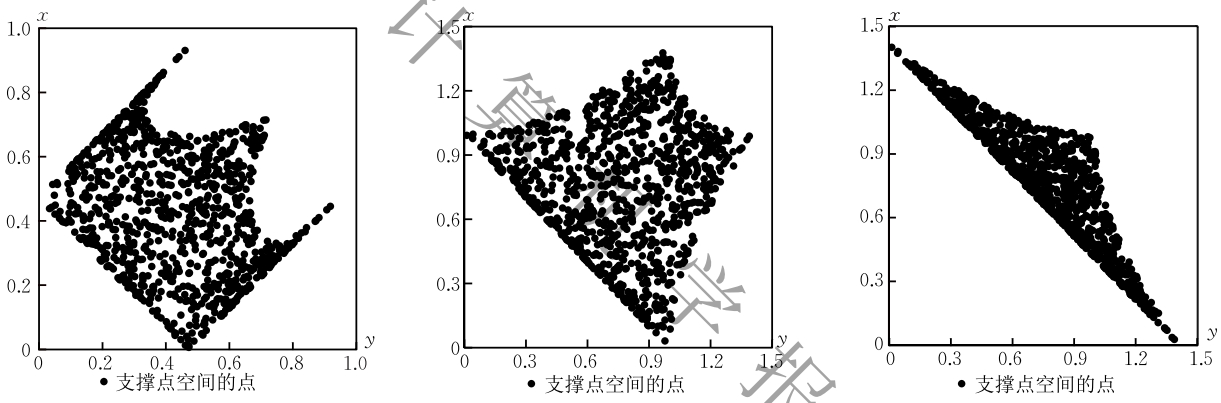


图 2 某二维均匀分布数据集



(a) 以(1/3,1/3)与(2/3,2/3)为支撑点 (b) 以(0,0)与(0,1)为支撑点 (c) 以(0,0)与(1,1)为支撑点

图 3 使用不同支撑点映射得到的支撑点空间

(2)密集支撑点

对于离群检测来说,支撑点既要能把数据区分开来,又要能在检测过程中,尽量排除非离群点,以进一步节省检测时间,甚至可提前结束离群检测过程而获得正确结果.

离群检测的过程中,随着离群度阈值 c 不断增大,支撑点周围将会有越来越多的对象不必检测而可直接作为非离群点排除,这就要求支撑点要选取在密集区域.如图 4 所示,位于密集区域的支撑点在离群检测过程中,能排除的非离群点较多,图 5 所示的支撑点则位于相对稀疏区域,其所能排除的非离群点很少,从而不适合作为支撑点.

综上所述可知,应用于离群检测的支撑点选取的两个目标是边缘支撑点和密集支撑点.然而,这两个目标实际上难以同时达到,因为边缘支撑点往往处于稀疏区域,而密集支撑点往往处于内部区域.

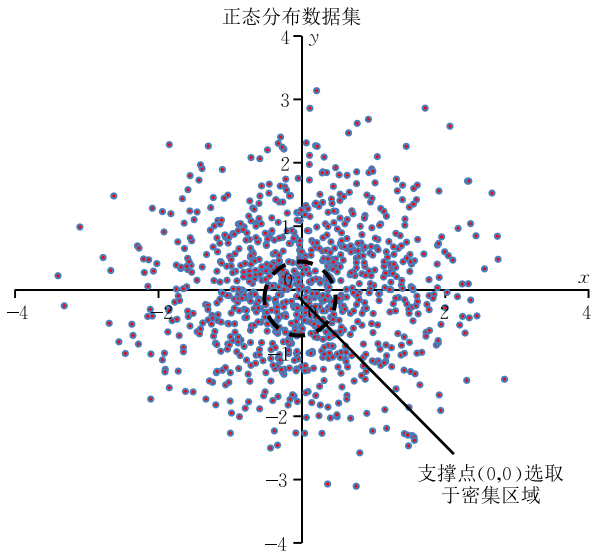


图 4 支撑点选取于密集区域

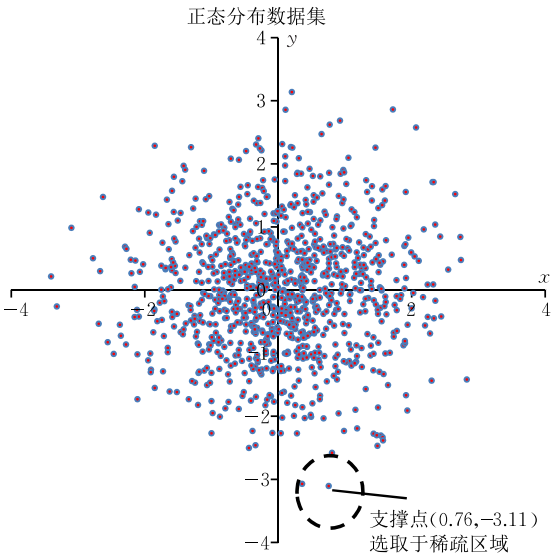


图 5　支撑点选取于稀疏区域

3.1.2　支撑点选取的范围

支撑点选取目标对选取结果非常重要. 一般来说,如果支撑点选取目标正确,那么在全局数据集选取的效果将优于在其子集上的选取结果,至少不比子集选取结果差,因为全局数据集包含子集. 但在数据集规模较大的情况下,基于全局数据集选取支撑点时间开销过大. 更为严重的是,如果数据集无法一次性调入内存,其还将引起较多的 IO 开销.

离群检测的支撑点选取目标和离群点数量较少的特点决定了其比度量空间索引更适合在数据集子集(以下简称子集)上选取支撑点. 由 3.1.1 节可知,选取目标之一是选取密集区域的对象,避免选取离群点作为支撑点. 而按照 Hawkins 的离群点定义可知,离群点在数据集中仅占一小部分,假设离群点占比为 1%. 那么子集的离群点分布情况大致可分为如下 3 种:

(1) 子集的离群点比例偏少,甚至完全没有离群点,显然这种情况对支撑点选取非常有利. 以下简称情况(1).

(2) 在数据集均匀分布的情况下,子集的离群点比例同样大约占 1%,这种情况相当于在全局数据集上均匀抽样,显然不影响支撑点选取操作. 而且结果将近似于在全局数据集上的选取. 以下简称情况(2).

(3) 极端情况下,数据集中的离群点都位于该子集中,那么只要子集规模大于 1%一定比例即可. 也就是说,一般离群点在空间分布上较为分散,彼此距离较远,因而只要在这些离群点的基础上再加上一定比例的正常点,再选取密集区域的对象作为支撑点,即可避免选取到离群点作为支撑点. 以下简称情况(3).

对于上述 3 种情况来说,情况(1)、(2)显然很容易选取支撑点,情况(3)只要让子集规模大于离群点规模一定比例即可. 值得庆幸的是,情况(1)、(2)非常常见,而情况(3)作为极端情况比较少见. 因此,基于子集选取支撑点显然能够满足目标一(边缘支撑点)的. 而对于目标二(密集支撑点),同样由于情况(1)、(2)较为常见而基本上能够满足.

3.1.3　相异支撑点选取算法

由 3.1.1 节的分析可知,支持点选取的两个目标难以同时满足,因此更好的办法是分别选取. 所谓相异支撑点选取算法,就是使用不同算法选取多种支撑点,依据离群检测不同阶段的不同需求分别予以应用,以克服同一种支撑点无法兼顾所有需求的缺点. 具体而言,就是在密集区域选取支撑点,配合使用终止规则,以提前结束离群检测过程;在边缘区域选取支撑点,用于构建支撑点空间,在检测过程中通过比较支撑点空间距离而减少距离计算次数. 考虑到精确选取密集支撑点的时间开销较大,本文实际上对于两个支撑点的选取算法分别是近似的密集支撑点选取算法、边缘支撑点选取算法.

众所周知,度量空间离群检测与度量空间相似性搜索一样,都可以通过选取支撑点,建立索引,从而加快检测/搜索速度. 然而,针对相似性搜索建立的索引,通常是多次、重复使用,以搜索不同的对象,因而值得花费较多时间选取支撑点及建立索引[29-30];而离群检测则不然,其建立的索引往往只使用一次或少量几次,因而索引建立开销不能太大. 而精确确定密集区域的时间开销非常大,甚至可能超过离群检测本身. 因此,本文提出近似的密集支撑点选取算法,以期可接受的时间开销之内确定密集支撑点.

如图 6 所示,近似的密集支撑点选取算法首先随机选取基准点,然后计算用于选取支撑点的数据集所有对象与基准点的距离,并按距离排序. 此时数据集相当于被降到一维,再进行等量划分(例如划分为 10 段),近似认为距离增量最小的分段为最密集

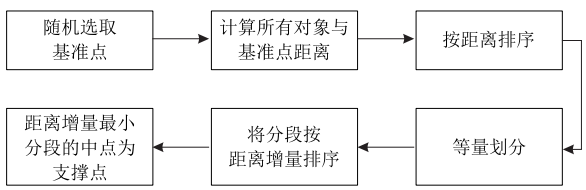


图 6　近似的密集支撑点选取算法

区域，取其中点作为支撑点.

FFT(Farthest-First Traversal)算法[26]为经典的支撑点选取算法，其核心思想是让支撑点之间的距离尽可能远，从而使数据映射到支撑点空间之后的各个坐标尽可能不同，以充分利用多个支撑点的优势.

图 7 为 FFT 算法流程. FFT 首先从数据集中随机选取一个对象作为支撑点，并将其加入支撑点集 P，然后计算数据集中除支撑点外所有对象 x 到 P 的距离，

$$dist(x,P)=\min\{d\mid d=dist(x,p_i),p_i\in P\} \quad (2)$$

取该距离值最大的对象作为下一个支撑点. 依此类推，直到选取足够数量的支撑点.

pick any $z\in S$ and set $P=\{z\}$
while $|P|<k$:
$z=\arg\max_{x\in S}\rho(x,P)$
$P=P\cup\{z\}$

图 7 FFT 算法

然而，细看 FFT 算法可发现，其选取的第一个支撑点为随机选取所得，性能存在着较大的不确定性. 因此本文实际采用的支撑点为 FFT 算法选取结果去掉首个支撑点余下的其它支撑点.

3.1.4 等距划分 VS 等量划分

在将数据集降至一维之后，先分段，再取最密集段的中点为支撑点. 至于如何分段，是一个值得探讨的问题. 常见的方法有等距划分与等量划分. 下面分析这两个划分方法.

图 8 展示的是等距划分. 它先在数据集中选定基准点，然后计算基准点到与其距离最远对象，按相同的距离增量划分. 假设最远距离为 d_f，拟划分为 n 段，那么可分别在与基准点距离为 d_f/n，$2d_f/n$，…，$(n-1)d_f/n$ 等处划分，从而将数据集划分为相对于基准点距离增量相等但每段包含的对象数量不一定相等的 n 段. 其确定密集区域的方法是，先统计各段所含对象数量，再对此数量进行排序，数量大者为支撑点选取所需的密集区域.

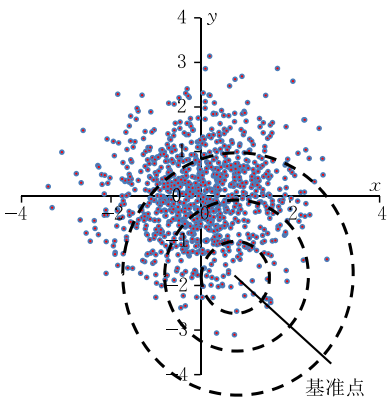


图 8 正态分布数据集等距划分示例

如图 9 所示，等量划分先从数据集选取基准点，然后自基准点开始，按相等的对象数量增量进行划分. 假设数据集规模为 N，拟划分为 n 段，那么可自基准点开始，逐渐增加距离，在对象数量为 N/n，$2N/n$，…，$(n-1)N/n$ 等处划分，从而将数据集划分为对象数量相等但距离增量不一定相等的 n 段. 在确定密集区域时，可先统计每段的距离增量，再按此增量大小排序，取增量小者作为密集区域.

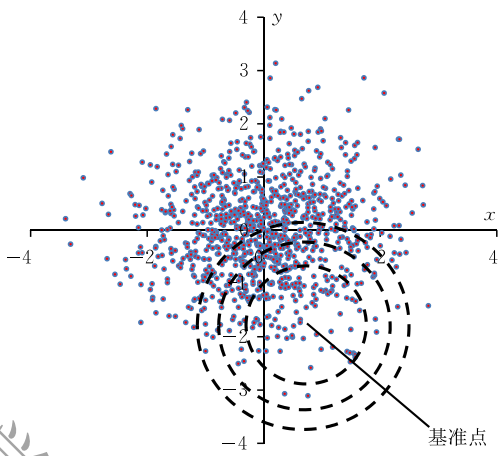


图 9 正态分布数据集等量划分示例

显然，通过等距划分和等量划分都可以得到相对密集的区域. 但实际上，这两种划分方法的细节情况大不相同. 等距划分更能直观形象地反映数据空间分布情况，但在数据集稀疏与密集区域过渡处，有可能造成密集与稀疏数据被划分到同一段(以下简称“疏密同段”). 而等量划分则能在很大程度上规避这一问题，因为等量划分时，密集区域往往就足够形成单独一段，甚至多段了. 然而，这也造成了等量划分的一大缺陷——密集区域被连续划分为多段，但实际上很可能在空间距离上非常邻近，不利于支撑点选取.

现进一步考虑等距划分和等量划分对在单一密集划分区域选取支撑点的影响. 通常可以选取中点作为支撑点，具体包括距离上的中位点(以下简称距离中点)和对象数量上的中位点(以下简称数量中点)：

(1) 对于等距划分，若按距离中点选取，若存在上述疏密同段的情况时，由于一般来说密集区域范围较小，稀疏区域范围较大，距离中点很可能属于稀疏区域，从而造成支撑点选取效果不理想的情况. 然

而，如果按数量中点选取，即使出现疏密同段，由于密集区域的对象数量往往占较大比例，而稀疏区域占比很小，数量中点几乎就只会在密集区域被选取了. 对于疏密同段的一种极端情况——划分区域数据呈现“密-疏-密”分布，且两个密集区域对象数量几乎相等时，数量中点可能就处于稀疏区域了. 但这种划分区域在上述第一步按数量排序获取密集区域时，很难被选中. 综合可知，等距划分宜用数量中点选取方法.

(2) 对于等量划分，由于基本上不存在疏密同段情形，对于候选区域，就更不至于疏密同段了. 因此，等量划分无论按距离中点还是数量中点选取支撑点，都会在候选区域的密集处选取得到.

最后从支撑点选取数量分析. 当只选取一个支撑点时，“等距划分＋数量中点”、“等量划分＋距离中点或数量中点”都能选取到较为理想的支撑点(符合密集支撑点选取目标). 然而在选取第二个或以后的支撑点时，情况就不一样了. 对于等距划分，因其相对来说能较好地反映数据集的空间分布情况，想选取与第一个支撑点距离较远的对象，甚至边缘对象，或者在靠近边缘较为密集的区域选取，都很容易实现. 而对于等量划分方法来说，如前所述，可能会导致单个密集区域被划分到连续多个片段，进而使选取出的多个支撑点相距较近，不利于使用三角不等性排除非离群点.

综上所述可知，当只需选取一个支撑点时，“等距划分＋数量中点”与“等量划分＋距离中点/数量中点”的选取方案都可行；但当所需支撑点数量在 2 个或 2 个以上时，应优先考虑“等距划分＋数量中点”的选取方案.

3.2　索引的建立

索引的建立过程如图 10 所示. 与相异支撑点选取算法对应的是，支撑点也分成两部分——密集支撑点和边缘支撑点. 数据集的所有对象与密集支撑点计算距离之后，按距离值降序排序，得到一维索引. 而数据集边缘支撑点计算距离之后，将这些距离值作为坐标，从而形成支撑点空间. 显然，支撑点空间的维数等于边缘支撑点数量.

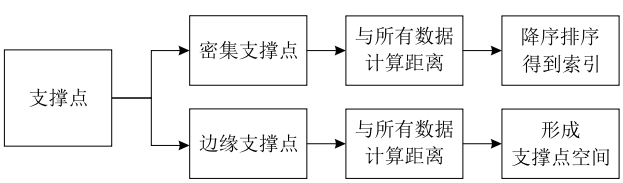


图 10　索引建立过程

3.3　剪枝规则

本节使用基于距离三角不等性的两个定理作为剪枝规则，减少距离计算次数. 给定 D 为数据集，c 为离群度阈值，$P=\{p_1,p_{2i}\}$ 为支撑点集($i=0,1,2,\cdots$)，其中 p_1 为密集支撑点，p_{2i} 为边缘支撑点，x 为离群检测算法拟检测的任意对象，$dist()$ 为距离函数，$nn_k(p,D)$ 表示对象 p 在数据集 D 中的第 k 最近邻.

定理 1(剪枝规则 1).　终止规则.

如果

$$dist(x,p_1)+dist(p_1,nn_k(p_1,D))<c \tag{3}$$

那么离群检测过程可以终止并得到正确的结果.

定理 2(剪枝规则 2).　排除非 k 最近邻的对象.

如果

$$||dist(x_t,p_i)-dist(x_j,p_i)||>dist(x_t,nn_k(x_t,D)) \tag{4}$$

那么 x_j 不可能为 x_t 的 k 最近邻.

上述定理 1 与由密集支撑点建立的索引配套使用，如果所用支撑点选取于高度密集区域，那么式中 $dist(p_1,nn_k(p_1,D))$ 将趋近于 0，此时将近似地有

$$dist(x,p_1)<c \tag{5}$$

换言之，在距离密集支撑点 c 范围内的对象几乎都可以作为非离群点排除，而不需要检测. 由此可见密集支撑点的重要性.

而定理 2 与由边缘支撑点建立的支撑点空间结合使用. 实际上，定理 1 和定理 2 都并不适用于对方的用途. 由密集支撑点建立的索引，能尽快使定理 1 的式子成立，从而提前终止离群检测过程. 而边缘支撑点往往处于稀疏区域，不利于定理 1 式子的成立. 对于定理 2 来说，使用边缘支撑点建立支撑点空间的效果较好.

4　基于多种支撑点的度量空间离群检测算法及分析

本节先阐述基于多种支撑点的度量空间离群检测算法，然后对算法复杂度进行分析，并对密集支撑点能够排除的不包含离群点的数据块数量进行详细讨论.

4.1　基于多种支撑点的度量空间离群检测算法

算法 1 描述了 VPOD 算法的具体过程. 算法在初始化离群度阈值、TOP n 离群点 D_n-$outlier$ 和数据块 B(第 1 行)之后，调用近似的密集支撑点选取算法和 FFT 算法，分别从子集选取密集支撑点和边

缘支撑点,其中 p_2 为 FFT 算法去掉第一个支撑点的选取结果(第 2～3 行).随后依据密集支撑点建立索引,并通过边缘支撑点映射到支撑点空间(第 4 行).如同 iORCA 算法,按照从远到近的顺序逐数据块检测离群点(第 5 行).

算法 1. Various Pivots based Outlier Detection algorithm(VPOD 算法).

输入:k, n, $pivotNum$, D

输出:D_n-$outlier$

1. $c \leftarrow 0$; D_n-$outlier \leftarrow \varnothing$; $B \leftarrow \varnothing$;
2. $p_1 \leftarrow densityPivotSelection$(subset of D);
3. $p_2 \leftarrow FFTPivotSelection$(subset of D, $pivotNum$);
4. $L \leftarrow$buildIndex(D, p_1, p_2);
5. WHILE $B \leftarrow get\text{-}next\text{-}block(L(D))$ {
6. IF (Rule 1 holds for $B(0)$) THEN break;
7. ELSE{
8. $startID \leftarrow$median of B;
9. $order \leftarrow$spiralOrder($L.id$, $startID$);
10. FOR each d in D with order{
11. FOR each b in B {
12. IF(Rule 2 doesn't hold for d and b){
13. $dis \leftarrow dist(b,d)$; update $b.NN$
14. IF $w_k(b,D) < c$ remove b from B;}}}
15. FOR each b in B {
16. D_n-$outlier \leftarrow$TOP($B \cup D_n$-$outlier$, n)
17. $c \leftarrow w_k(D_{n,k}, D)$ //update c;}}}
18. return D_n-$outlier$;

在检测过程中,首先判断每个数据块第一个对象是否使终止规则成立,若成立,则终止整个离群检测过程,输出 TOP n 离群点检测结果(第 6 行).若不成立,则从索引序列上该数据块的中点对象开始,按照螺旋顺序搜索该数据块所有对象的 k 最近邻(第 8～9 行).在搜索 k 最近邻时,先计算剪枝规则二是否成立,若成立,则该对象不可能成为被检对象的 k 最近邻,不需要计算其真实距离(第 12～13 行).实时计算被检对象的当前离群度,如果小于离群度阈值 c,则直接从数据块移除(第 14 行).每检测完一个数据块,更新 TOP n 离群点和离群度阈值(第 16～17 行).当所有数据块都检测完或者终止规则成立时,结束检测过程,输出 TOP n 离群点.

4.2 算法分析

相对于 iORCA 算法来说,VPOD 算法增加了两类支撑点的选取过程,该过程较 iORCA 算法的随机选取耗时略多,但相对于整个离群检测过程来说,这个时间开销是忽略不计的.在时间复杂度方面,VPOD 算法采用了相异支撑点选取算法,包括两种支撑点的选取——密集支撑点与边缘支撑点.密集支撑点在选取时需要与数据集(子集)所有对象计算距离,时间复杂度为 $O(N)$,然后按距离值排序,时间复杂度为 $O(N\log N)$.边缘支撑点采用 FFT 算法选取,视所选支撑点的数量,进行若干次时间复杂度为时间复杂度为 $O(N)$ 的距离计算操作,并且计算距离的同时,仅需保留最大距离的对象作为支撑点,而不需要进行排序.此外,由于支撑点一般仅需从数据集子集选取,通常在一个数据块选取即可,节省了选取时间.

建立索引时,VPOD 算法也较 iORCA 算法多了映射数据集至支撑点空间的过程,多出的距离计算次数为 $pivotNum \times |D|$,与整个离群检测过程比起来,同样忽略不计.

在离群检测过程中,VPOD 算法较 iORCA 增加了剪枝规则二,大幅减少距离计算次数,使算法运行时间明显缩短,甚至不到 iORCA 算法所用时间的一半.检测过程中,假设 N_B 为数据块数量,在最坏情况下,所有 N_B 个数据块都需要被检测,且每个数据块在被检测时都需要调取 N_B 个数据块来搜索 k 最近邻,总共需要调取数据块的次数为 $O(N_B^2)$.然而由于 VPOD 算法常常触发终止规则而提前结束检测,实际需要检测的数据块远低于 N_B.以下分析 VPOD 算法实际需要调取的数据块数量.

为简化分析,我们假设给定数据集,大小为 s,服从二维正态分布

$$f(x,y)=\frac{1}{2\pi}\mathrm{e}^{-\frac{x^2+y^2}{2}} \tag{6}$$

同时,近似认为密集支撑点选取算法在一个大小为 t 的数据块选取到的支撑点是该数据块最密集的对象 p_b.为便于分析,此处引入位置序号及相对位置的概念.

定义 1. 位置序号.

位置序号(Location Order),标记为 Lo,就是对象在数据集中按密集程序排序的序号.

定义 2. 相对位置

相对位置(Relative Location),标记为 Lr,就是位置序号与数据集大小之比,即

$$Lr=\frac{Lo}{s} \tag{7}$$

显然,数据集最密集对象 p_w(即二维正态分布的中心位置)的位置序号为 $Lo=0$,相对位置 $Lr=0$,而最稀疏处的对象位置序号为 $Lo=s$,相对位置 $Lr=1$.

现求 p_b的相对位置期望值，即随机抽取的 t 个对象(一个数据块大小)之中，最小相对位置的期望值 $E(Lo)$.

设 p_b的相对位置为 x_t，那么该数据块其余 $t-1$ 个对象的相对位置范围为 $x_t\sim1$，故这样的抽取概率 $P_t(x_t)$为

$$P_t(x_t)\approx\frac{(1-x_t)^{t-1}}{\int_0^1(1-x_t)^{t-1}\mathrm{d}x_t},\ t\geq1 \tag{8}$$

将 x_t的值从 0～1 的范围内积分，可得其期望值 $E(x_t)$为

$$\begin{aligned}E(x_t)&=\int_0^1P_t(x_t)x_t\,\mathrm{d}x\\&=\int_0^1\frac{(1-x_t)^{t-1}}{\int_0^1(1-x_t)^{t-1}\,\mathrm{d}x_t}x_t\,\mathrm{d}x_t\\&=\int_0^1\frac{x_t(1-x_t)^{t-1}}{\int_0^1(1-x_t)^{t-1}\mathrm{d}x_t}\mathrm{d}x_t\end{aligned} \tag{9}$$

注意到 $P_t(x_t)$的分母为常数，可先提出积分之外，故上式可化简为

$$E(x_t)=\frac{\int_0^1x_t(1-x_t)^{t-1}\mathrm{d}x_t}{\int_0^1(1-x_t)^{t-1}\mathrm{d}x_t} \tag{10}$$

式(10)的分子和分母都是 B 函数，即

分子为：$B(2,t)=\int_0^1x_t(1-x_t)^{t-1}\mathrm{d}x_t$；

分母为：$B(1,t)=\int_0^1x_t^0(1-x_t)^{t-1}\mathrm{d}x_t$.

而 Γ 函数对于正整数 i，有

$$\Gamma(i)=(i-1)!$$

由 B 函数与 Γ 函数的关系，有

$$B(i,j)=\frac{\Gamma(i)\Gamma(j)}{\Gamma(i+j)}.$$

结合上式，有

$$B(i,j)=\frac{(i-1)!(j-1)!}{(i+j-1)!}.$$

综上，可得

$$E(x_t)=\frac{B(2,t)}{B(1,t)}=\frac{\frac{(2-1)!(t-1)!}{(2+t-1)!}}{\frac{(1-1)!(t-1)!}{(1+t-1)!}}=\frac{1}{t+1} \tag{11}$$

也就是说，在大小为 t 的数据块里选取到最密集对象 p_b时，其相对位置期望值为

$$E(x_t)=\frac{1}{t+1} \tag{12}$$

位置序号为

$$Lo=Lr\times s=E(x_t)s=\frac{s}{t+1} \tag{13}$$

由于数据块规模一般都较大，例如常用 $t=1000$，因此无论从相对位置还是位置序号来看，p_b都已非常接近数据集最密集对象 p_w.

考虑到二维正态分布的特点，单纯从相对位置或者位置序号来判断 p_b与 p_w的距离尚不够准确. 现推导 p_b与 p_w的距离. 设该距离值为 Z，问题转化成求 Z 的分布函数，且有

$$Z^2=X^2+Y^2,\ Z\geq0,$$

$$\begin{aligned}F_Z(z)&=P\{X^2+Y^2<z^2\}=\iint\limits_{x^2+y^2<z^2}\frac{1}{2\pi}\mathrm{e}^{-\frac{x^2+y^2}{2}}\mathrm{d}x\mathrm{d}y\\&=\frac{1}{2\pi}\int_0^{2\pi}\mathrm{d}\theta\int_0^z\mathrm{e}^{-\frac{r^2}{2}}r\,\mathrm{d}r=1-\mathrm{e}^{-\frac{z^2}{2}}\end{aligned} \tag{14}$$

令 $E(x_t)=F_Z(z)$，即有

$$\frac{1}{t+1}=1-\mathrm{e}^{-\frac{z^2}{2}} \tag{15}$$

由式(15)可解得

$$z=\sqrt{-2\ln\frac{t}{t+1}} \tag{16}$$

从式(16)可看出，当 t 的值较大时，p_b与 p_w的距离是很小的正数，例如当 $t=1000$ 时，z 的值仅为 0.029.

鉴于 p_b与 p_w的距离非常小，相对于离群度阈值 c 来说可以忽略不计. 为便于分析，此处进一步假设密集支撑点选取算法选取到的支撑点就是整个数据集最密集的支撑点，即 $p_b=p_w$，并设算法需要检测的数据块数量占数据集比例为 q，那么算法不需检测的数据块比例为 $1-q$，根据剪枝规则一对高度密集支撑点的分析结果，可令 $F_Z(z)=1-q$，得

$$1-\mathrm{e}^{-\frac{z^2}{2}}=1-q \tag{17}$$

解之，得

$$q=\mathrm{e}^{-\frac{z^2}{2}} \tag{18}$$

举例说，当 $z=1$ 时，$q\approx0.607$；而当 $z=2$ 时，q 仅 0.37，即只需要检测 37% 的数据块. 算法实际应用时，数据集不一定符合正态分布，也不一定是二维的，因此实际需要检测的数据块比例各不相同，而算法的运行时间也深受该比例影响，本文第 5 节将讲述实验结果.

5　实验结果及分析

本节首先介绍实验所用数据集，然后实验平台和设置方法，最后给出实验结果及详细分析.

5.1 实验数据集

本文实验一共使用 4 个真实数据集，全部来源于 UCI Machine Learning Repository①，具体如下：

KDD Cup 1999 数据集：最初来源于美国国防部高级规划署(DARPA)，后经哥伦比亚大学的 Stolfo 教授等处理而成. 本文采用其 10%版本的 TCP 数据集，包含 190065 个数据，每个数据包括 42 个属性.

Shuttle 数据集：来源于美国国家航空航天局(NASA)的航天飞机数据. 该数据集包含 58000 个数据，每个数据包括 9 个属性.

Molecular Biology 数据集：为 DNA 数据集，来源于 Genbank 64.1，包含 3190 个数据，每个数据包括 60 个属性. 需要注意的是，该数据集不适合欧氏距离或海明距离，而是采用蛋白质的比对距离(Alignment).

Landsat Satellite 数据集：即陆地卫星数据集，最初来源于美国航空航天局的遥感数据，包含 6435 个数据，每个数据包括 36 个属性.

5.2 实验平台及设置

实验环境为 Intel Core i7-2600 CPU@3.40GHz 及 8GB 内存的 Windows 7 SP1 操作系统，实验程序使用 Visual Studio 2012 开发，并以 Release 模式编译.

除非特别说明，为便于与对比算法 iORCA 比较，本文实验设置参数与其一致，即最近邻数量 $k=5$，拟检测离群点数量 $n=30$，数据块大小 $m=1000$. 值得一提的是，由于 VPOD 算法使用与 iORCA 所用的离群点定义保持一致，离群点检测结果也与其保持一致，因此本文未进行准确率方面的实验.

实验对各个数据集都作为归一化处理(min-max 标准化方法)，将数据映射至 0～1 区间，并采用以下距离函数：

$$dist(x_i,x_j)=\sqrt{\sum_{k=1}^{d}\delta(x_{ik},x_{jk})} \tag{19}$$

其中，

$$\delta(x_{ik},x_{jk})=\begin{cases}(x_{ik}-x_{jk})^2, & \text{对于连续属性}\\ x_{ik}-x_{jk}\ ?\ 0:1, & \text{对于离散属性}\end{cases}.$$

显然，在只有连续属性时，式(19)实际上相当于欧几里德距离，而在只有离散属性时，该函数相当于海明距离. 值得注意的是，只要数据集带有符合度量空间距离三特性(对称性、正定性和三角不等性)的距离函数，就可以使用本文离群点定义和算法处理.

5.3 实验结果及分析

为减少实验误差，本文各个实验均运行 10 次并取平均值作为最终结果. 对于 VPOD 需要取某一数据块来选取支撑点的情况，则分别在前 10 个数据块选取. 值得注意的是，Molecular Biology 数据集与 Landsat Satellite 数据集数据块数量小于 10 个，则按实际数据块数量选取，再取平均值.

实验首先测试了 VPOD 算法与各个对比算法的运行时间随数据集规模变化情况，为便于对比分析，iORCA 与 VPOD 建立索引所需时间也置于图中，尽管其数值很小难以看出.

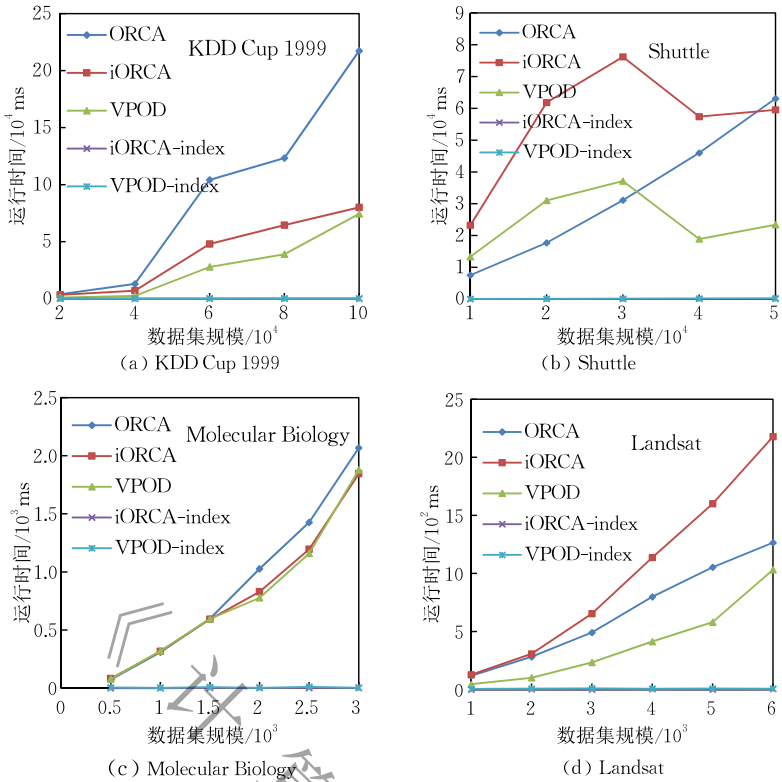
如图 11 所示，VPOD 算法在 4 个数据集上的运行时间基本上都明显低于对比算法 iORCA 和 ORCA. 并且其建立索引的时间开销相对于整个离群检测过程来说，可以忽略不计.

从图 11(a) KDD Cup 1999 数据集上的实验结果可以看出，使用了索引的 iORCA 和 VPOD 算法运行速度都大幅领先于 ORCA 算法，并且 VPOD 算法更胜一筹. 而从图 11(b)可以看出，在数据集规模较小时，VPOD 算法并不占优势，这是因为 Shuttle 数据集本身并不完全符合 Hawkins 的离群点定义. 而且在其规模增大之后，使用索引的 VPOD 和 iORCA 算法运行时间反而减少，从而领先于未使用索引的 ORCA 算法，这是由于 Shuttle 数据集在规模增大之后，其分布呈现不均匀，更能使索引发挥加速作用. 图 11(c)所示 Molecular Biology 数据集上，VPOD 算法仅略优于对比算法 iORCA，以及小幅优于 ORCA 算法，原因在于该数据集分布较均匀，并没明显的离群点，未能发挥本文两类支撑点的剪枝作用. 图 11(d)所示 Landsat 数据集上，iORCA 算法的终止规则并未明显减少对数据块的检测，导致其运行时间甚至高于 ORCA 算法. 而即使在这样的情况下，VPOD 算法依然取得高于 ORCA 的运行速度，这充分体现了算法所用相异支撑点选取算法的优越性能.

值得注意的是，在 Shuttle 数据集上索引发挥作用时，得益于相异支撑点选取算法及相应的剪枝规则，VPOD 算法运行速度大幅领先于 iORCA 算法. 而在 KDD Cup 1999 和 Landsat 数据集上，VPOD 算法也有不同程度的领先优势.

鉴于 iORCA 和 VPOD 算法建立索引的时间在图 11 中呈现紧贴 X 轴的现象，为了进一步了解其具

① http://archive.ics.uci.edu/ml/datasets.html



(a) KDD Cup 1999　(b) Shuttle　(c) Molecular Biology　(d) Landsat

图 11　算法在 4 个数据集上的运行时间

体情况，表 2 和表 3 列举了这两个算法在 KDD Cup 1999 数据集建立索引具体耗时及占整个离群检测所需时间比例.

表 2　iORCA 与 VPOD 在 KDD Cup 1999 数据集建立索引时间开销

数据集大小	iORCA 索引耗时/ms	占比/%	VPOD 索引耗时/ms	占比/%
20 000	3.2	0.09	23.4	1.58
40 000	14.0	0.20	40.7	1.18
60 000	21.5	0.04	62.4	0.15
80 000	26.5	0.04	66.9	0.12
100 000	29.8	0.04	93.7	0.07

表 3　iORCA 与 VPOD 在 Shuttle 数据集建立索引时间开销

数据集大小	iORCA 索引耗时/ms	占比/%	VPOD 索引耗时/ms	占比/%
10 000	3.1	0.13	7.7	0.57
20 000	4.8	0.08	10.8	0.35
30 000	7.8	0.10	13.9	0.37
40 000	11.1	0.19	17.1	0.90
50 000	14.1	0.24	21.9	0.93

从表 2 和表 3 可以看出，无论 iORCA 还是 VPOD 算法，其建立索引所需时间都是非常少的. VPOD 算法在建立索引时需要较多时间用于选取支撑点及建立支撑点空间，故耗时较多，但相对于整体的时间开销来说，仍然忽略不计. 而这小比例的耗时，换来了整体检测时间的大幅减少.

VPOD 算法在运行时间上的优势，归根到底是距离计算次数的减少. 图 12 为 3 个算法分别在 4 个数据集上的距离计算次数情况. 可以看出，在距离计算次数上，VPOD 算法显示出比运行时间更大的优势. 这充分说明了相异支撑点选取算法对于减少距离计算次数有着明显作用，而距离计算在搜索领域(包括搜索离群点)通常被认为会带来较为昂贵的时间开销，尤其是对于维度较高的情况. 而对于维数较低的数据集，VPOD 算法的优势相对较小，例如 Shuttle 数据集仅 9 维，尽管距离次数获得大幅减少，但其在运行时间上的优势并不明显，甚至在数据集规模较小时未能领先 ORCA 算法.

为了研究最近邻数量 k 对 VPOD 算法的影响，实验测试了其运行时间随 k 变化的情况. 鉴于 Landsat 数据集规模较小，本实验仅使用 KDD Cup 1999 及 Shuttle 两个数据集. 从图 13 可以看出，得益于相异支撑点选取算法所选得的支撑点较强的剪枝能力，在 k 值增大的情况下，算法运行时间增幅并不明显，甚至某些时候还有可能减少，例如 Shuttle 数据集. 该数据集在 k 自 10 增大到 40 的过程中，其离群度阈值反而更能使 VPOD 算法起到剪枝的作用，从而减少距离计算次数，加快检测速度.

实验最后测试了 VPOD 算法运行时间随边缘支撑点个数变化的情况(数据集规模同上). 从图 14

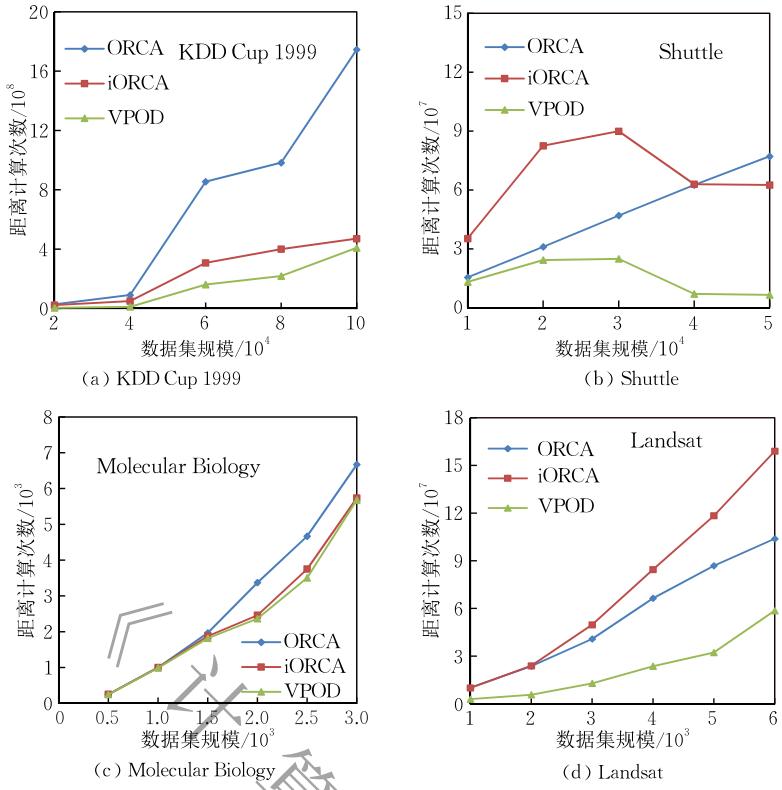


图 12 算法在 4 个数据集上的距离计算次数

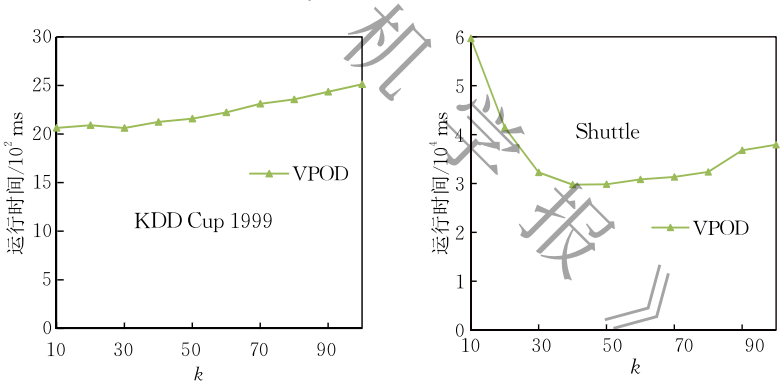


图 13 VPOD 算法在两个数据集上的运行时间随 k 变化情况

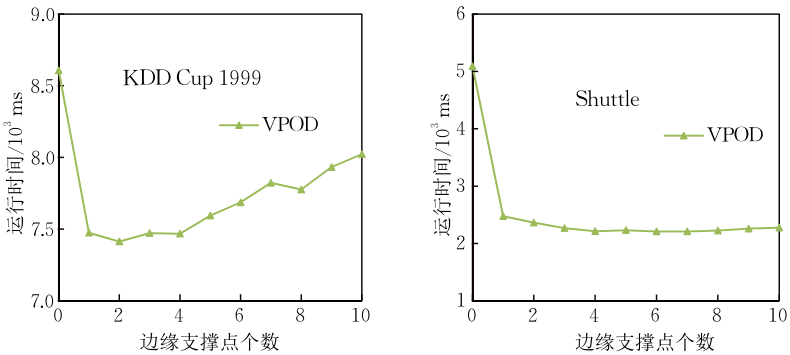


图 14 VPOD 算法在两个数据集上的运行时间随边缘支撑点个数变化情况

可以看出,当边缘支撑点个数为 0,即不使用边缘支撑点,不将数据集映射至支撑点空间的情况下,算法运行时间明显高于边缘支撑点数量不为 0 的情况.而当边缘支撑点数量增至 1、2 或 3 时,运行时间即已减少至较为理想的状态,继续增大后运行时间的改善情况并不明显,反而会增加选取支撑点及建立

索引的时间开销.因此可以认为边缘支撑点个数设置为 2 左右即可获得较好的效果,没必要一味增加其数量.

对于拟检测离群点数量 n 来说,当其增大时,将导致离群度阈值增长速度变慢,但这个影响对于文中 3 个算法是同样存在的,VPOD 算法仍将占优势.而对于数据块大小 m 来说,当其增大时,将使得单次更新离群度阈值幅度变大,但更新次数减少,尤其是检测第一个数据块时,由于离群度阈值的初始值为 0,无法使用离群度阈值来排除非离群点.因此,过大的 m 值将导致这个阶段时间开销较大.

6　结束语

本文针对 iORCA 算法仅提供随机支撑点选取不同算法导致其离群检测结果不稳定的论述,且未充分利用距离三角不等性减少距离计算次数的问题,提出了基于相异支撑点选取的离群检测算法 VPOD.该算法采用两种不同的支撑点选取算法,分别选取密集支撑点和边缘支撑点,然后建立索引及支撑点空间,在离群检测过程中充分使用剪枝规则,减少距离计算次数.实验结果表明,VPOD 算法较已有的 iORCA 及 ORCA 算法在检测速度方面有了明显的提升,平均加速比达 2.05.

下一步,我们将对支撑点选取算法进行优化,继续探索距离三角不等性在减少距离计算次数方面的作用,以期进一步提高检测速度.此外,我们对基于支撑点技术的度量空间并行离群检测也非常感兴趣,将基于集群架构开展研究工作.

参　考　文　献

[1] Li Yue. Research of information analysis method based on outlier detection. Information & Communications, 2013, (3): 132-133(in Chinese)
(李越. 基于离群点数据挖掘的情报分析方法探析. 信息通信, 2013, (3): 132-133)

[2] Hawkins D M. Identification of Outliers. Berlin, Germany: Springer, 1980

[3] Othman Z A, Bakar A A, Ibrahim R, et al. Rough outlier method for network intrusion detection. International Journal of Information Processing & Management, 2013, 4(7): 39-50

[4] Kumar M, Mathur R. Unsupervised outlier detection technique for intrusion detection in cloud computing//Proceedings of the International Conference for Convergence of Technology. Pune, India, 2014: 1-4

[5] Srimani P D P K, Koti M S. Outlier mining in medical databases by using statistical methods. International Journal of Engineering Science & Technology, 2012, 4(1): 239-246

[6] Hauskrecht M, Batal I, Valko M, et al. Outlier detection for patient monitoring and alerting. Journal of Biomedical Informatics, 2013, 46(1): 47-55

[7] Pimentel M A, Clifton D A, Clifton L, et al. A review of novelty detection. Signal Processing, 2014, 99: 215-249

[8] Iqbal Q, Aggarwal J K. Image retrieval via isotropic and anisotropic mappings. Pattern Recognition, 2002, 35(12): 2673-2686

[9] Levenshtein V I. Binary codes capable of correcting deletions, insertions and reversals. Problems of Information Transmission, 1965, 10(1): 707-710

[10] Shen S-Y, Wang K, Hu G, et al. On the alignment space//Proceedings of the IEEE Engineering in Medicine and Biology Conference. Shanghai, China, 2006: 244-247

[11] Gusfield B D. Algorithms on strings, trees, and sequences: Computer science and computational biology. ACM Sigact News, 2010, 28(3): 41-60

[12] Knorr E M, Ng R T. Algorithms for mining distancebased outliers in large datasets//Proceedings of the International Conference on Very Large Data Bases. New York, USA, 1998: 392-403

[13] Knorr E M, Ng R T, Tucakov V. Distance-based outliers: Algorithms and applications. The VLDB Journal, 2000, 8(3-4): 237-253

[14] Ramaswamy S, Rastogi R, Shim K. Efficient algorithms for mining outliers from large data sets. ACM SIGMOD Record, 2000, 29(2): 427-438

[15] Angiulli F, Pizzuti C. Outlier mining in large high-dimensional data sets. IEEE Transactions on Knowledge and Data Engineering, 2005, 17(2): 203-215

[16] Guttman A. R-Trees: A dynamic index structure for spacial searching//Proceedings of the 1984 ACM SIGMOD International Conference on Management of Data. 1984: 47-57

[17] Bentley J L. Multidimensional binary search trees used for associative searching. Communications of the ACM, 1975, 18(9): 509-517

[18] Berchtold S, Keim D A, Kriegel H P. The X-tree: An index structure for high-dimensional data//Proceedings of the 22nd International Conference on Very Large Data Bases. San Francisco, USA, 1996: 28-39

[19] Chaudhary A, Szalay A S, Moore A W. Very fast outlier detection in large multidimensional data sets//Proceedings of the Data Mining and Knowledge Discovery (DMKD). Madison, USA, 2002

[20] Ren D, Rahal I, Perrizo W, et al. A vertical distance-based outlier detection method with local pruning//Proceedings of the 13th ACM International Conference on Information and Knowledge Management. Washington D.C., USA. 2004: 279-284

[21] Wang Y, Parthasarathy S, Tatikonda S. Locality sensitive outlier detection: A ranking driven approach//Proceedings of the IEEE 27th International Conference on Data Engineering (ICDE). Hannover, Germany, 2011: 410-421

[22] Pillutla M R, Raval N, Bansal P, et al. LSH based outlier detection and its application in distributed setting//Proceedings of the 20th ACM International Conference on Information and Knowledge Management. Glasgow, UK, 2011: 2289-2292

[23] Wang Xi-Te, Shen De-Rong, Bai Mei, et al. BOD: An efficient algorithm for distributed outlier detection. Chinese Journal of Computers, 2016, 39(1): 36-51(in Chinese)
(王习特, 申德荣, 白梅等. BOD: 一种高效的分布式离群点检测算法. 计算机学报, 2016, 39(1): 36-51)

[24] Bay S D, Schwabacher M. Mining distance-based outliers in near linear time with randomization and a simple pruning rule//Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Washington, USA, 2003: 29-38

[25] Bhaduri K, Matthews B L, Giannella C R. Algorithms for speeding up distance-based outlier detection//Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. San Diego, USA, 2011: 859-867

[26] Hochbaum D S, Shmoys D B. A best possible heuristic for the k-center problem. Mathematics of Operations Research, 1985, 10(2): 180-184

[27] Traina J R C, Santos Filho R F, Traina A J, et al. The Omni-family of all-purpose access methods: A simple and effective way to make similarity search more efficient. The VLDB Journal, 2007, 16(4): 483-505

[28] Mao R, Miranker W L, Miranker D P. Pivot selection: Dimension reduction for distance-based indexing. Journal of Discrete Algorithms, 2012, 13: 32-46

[29] Mao R, Xu H, Wu W, et al. Overcoming the challenge of variety: Big data abstraction, the next evolution of data management for AAL communication systems. IEEE Communications Magazine, 2015, 53(1): 42-47

[30] Mao R, Zhang P, LI X, et al. Pivot selection for metric-space indexing. International Journal of Machine Learning and Cybernetics, 2016, 7(2): 311-323

XU Hong-Long, born in 1986, Ph.D., lecturer. His research interests include data mining and big data.

TANG Song, born in 1993, double bachelor's degree. His research interests include ordinary differential equation and metric space.

MAO Rui, born in 1975, Ph.D., professor. His research interests include metric space indexing, big data management and analysis.

SHEN Jing, born in 1993, M. S. candidate. Her research interest is complex network.

LIU Gang, born in 1977, Ph.D., lecturer. His research interests include parallel algorithm, network on chip.

CHEN Guo-Liang, born in 1938, professor, Ph.D. supervisor, member of the Chinese Academy of Sciences. His research interests focus on high performance computing.

Background

Data mining is an important technique to discover the value from mass data and make it useful. Many data mining methods focus on finding large patterns. However, in some cases, one person's noise is another person's signal, and there may be some important information included in rare events of exceptional cases. Outlier detection is such a technique to find small patterns, and it has been widely applied in many fields, such as network intrusion detection, public health and medical monitoring.

The wide use of outlier detection has aroused enthusiasm in many researchers. Over the past few decades, many outlier definitions and detection algorithms have been proposed in the literatures. Among these, distance based outlier definitions and detection algorithms have got rapidly development because of their advantage in universal property. Index based detection methods have further speeded up the detection procedure. Though excellent methods have emerged in large numbers, such as ORCA and iORCA, two state-of-art algorithms, some problems still keep unsolved in the existed methods. Most of them apply domain information other than distance, making the universal property weakened. What's more, some pivot based methods are short of pivot selection method, and they cannot take full advantage of distance triangle inequality, resulting in low detection speed.

In this paper, we proposed a new perspectivecalled outlier detectionin metric space, which consists of distance based outlier definition and complete distance based outlier detection algorithm, to limit the outlier detection method in

metric space, so that it can achieve more universal property. Among this, we proposed Various Pivots based Outlier Detection Algorithm(VPOD). With the help of two different pivot selection methods and the related pruning rules, VPOD get the average speedup of 2.05 over the latest method in the same field, and the distance calculation times are reduced by 51.14% on average.

This work is supported by the National High Technology Research and Development Program of China under Grant No. 2015AA015305, the National Natural Science of China under Grant Nos. U1301252, U1501254, the Guangdong Key Laboratory Project No. 2017B030314073, the Guangdong Natural Science Foundation under Grant No. 2015A030313636, and the Shenzhen Science and Technology Plan under Grant No. CXZZ20140418182638764. These projects focus on big data management and analysis, such as metric space index and metric space data mining. As a research team, we rely on the support of these projects and achieved a series of scientific research results, including metric space indexing algorithm, pivot selection method for index, metric space data mining method, et al. The achievement of this paper is a part of metric space data mining based on the above projects, namely outlier detection inmetric space.