一种基于加权KNN的大数据集下离群检测算法

王 茜 杨正宽
(重庆大学计算机学院 重庆 400044)

摘 要 传统KNN算法是在基于距离的离群检测算法的基础上提出的一种在大数据集下进行离群点挖掘的算法，然而KNN算法只以最近的第k个邻居的距离作为判断是否是离群点的标准有时也失准确性。给出了一种在大数据集下基于KNN的离群点检测算法，即在传统KNN方法的基础上为每个数据点增加了权重，权重值为与最近的k个邻居的平均距离，离群点为那些与第k个邻居的距离最大且相同条件下权重最大的点。算法能提高离群点检测的准确性，通过实验验证了算法的可行性，并与传统KNN算法的性能进行了对比。

关键词 离群点，数据挖掘，权重，划分
中图法分类号 TP391 **文献标识码** A

Algorithm for Outlier Detection in Large Dataset Based on Weighted KNN

WANG Qian YANG Zheng-kuan
(College of Computer Science, Chongqing University, Chongqing 400044, China)

Abstract Traditional KNN is an advanced algorithm based on the distance of outlier detection algorithm on large dataset. However this algorithm only uses the k^{th} nearest neighbor as the criterion for outlier which is inaccurate under certain condition. This paper presented a weighted KNN outlier detection algorithm for large datasets. In this algorithm, a weight factor is presented. It represents the average distance of its k nearest neighbors. The outliers are those having the largest distance with it's k^{th} neighbor and having the biggest weight under the same condition. The algorithm improves the accuracy of the outlier detection algorithm. Experiment result shows that the algorithm is feasible compared with the traditional KNN.

Keywords Outlier, Data mining, Weight, Partition

1 引言

离群点检测作为知识发现的重要部分，被广泛地应用于欺诈识别、入侵检测、故障诊断及恶劣天气预报等领域。近年来，随着人们对离群数据挖掘重要性认识的不断加深，以及其越来越广泛的应用，离群点挖掘成为了数据挖掘领域的热点之一。离群点检测算法大致可分为：基于分布的方法、基于深度的方法、基于距离的方法、基于密度的方法和基于聚类的方法。基于分布的方法采用标准统计分布模型，那些偏离模型的点被认为是离群点[1]；基于深度的方法主要采用几何学的方法，把数据对象组织到数据空间的不同层面中，那些在较浅层面的数据更有可能是离群点[2]；基于密度的方法为数据集中的点定义局部离群因素(LOF)，且用LOF来计算数据对象的离群程度，离群点被认为是离群程度比较大及与周围的邻居点关系比较疏离的点[3]；基于聚类的方法认为离群点是那些数据集聚类后的副产品[4]。

基于距离的离群方法最早由Knorr与Ng提出[5]，其主要思想是将数据看作空间中的点，而离群点被定义为与数据集中大多数点之间的距离都大于某个阈值的那些点，它使用算法nested-loop和cell-based，前者对数据集中的点N具有$O(N^2)$时间复杂度，后者对数据的维数具有指数的时间复杂度，尤其在超过4维以后效率明显降低，所以它们在高维大数据集下都不具效率。针对其不足，Ramaswamy与kyuseok等人提出了一种在大数据集下挖掘离群点的方法[6]。他们对离群点的定义为前n个与其最近第k个邻居的距离最大的点被认为是离群点，避免了基于距离的离群点检测算法需要用户设定距离参数值d的局限，它使用基于划分的方法并对数据集中的点N有线性的时间复杂度，且数据的维数对算法的执行时间影响不大，但它只以与最近第k个邻居的距离作为判断离群点的标准有时也不够准确，即无法判断在与最近的第k个邻居的距离相同时哪个点更可能是离群点。Augiulli与Pizauti对基于距离的离群点给出了一个新的定义[7]，他们定义前n个离群点是权重最大的前n个点，权重是数据点与其最近k个邻居的距离之和，但其在权重相同的情况下无法判断哪些点更可能是离群点。近年来基于KNN的离群点检测的应用层出不穷，如Fuzzy KNN[8]在处理时流数据上有较高的效率；AGA/KNN[9]能有效处理生物学上的基因数据；OC/KNN[10]针对数据集产生最优聚类结果。其它高效的基于距离的离群点检测算法有Ren等人的By-Neighbors方法[11]，其使用了一种新颖的垂直数据结构来检测离群点。

到稿日期：2010-11-05 返修日期：2011-05-02 本文受国家自然科学基金项目(61073058)资助。
王 茜(1964－)，女，博士，副教授，硕士生导师，主要研究方向为信息安全、电子商务、远程教育课件工具；杨正宽(1986－)，男，硕士生，主要研究方向为数据挖掘。

本文提出了一种基于加权 KNN 的高效离群检测算法，离群点定义为那些具有最大的与第 k 个邻居的距离和权重的点，其权重为与最近的 k 个邻居的平均距离。算法由两部分组成，其首先将数据集划分成互不相交的子集，使用基于划分的算法找到候选子集，然后从候选子集中利用加权的思想寻找离群点，通过一些剪枝方法减少对一些非离群点的计算。

2 相关定义

2.1 基于距离的离群点(Distance-Based Outliers)定义

定义 1 对数据集上任意点 p，$D^k(p)$表示点 p 与其最近的第 k 个邻居的距离，n 表示需要找到的离群点个数，k 表示最近邻居数。当且仅当数据集中最多只有 $n-1$ 个点 p'满足 $D^k(p')>D^k(p)$，则点 p 是 D_n^k 离群点[6]。

性质 1 若点 p 在其 $D/2$ 邻域内的点数大于 k，即 $N(Nbr(p,D/2))>k$，则点 p 与其 $D/2$ 邻域内的所有点均不是离群点。其中 D 表示前 n 个离群点 D^k 距离的最小值，$Nbr(p,D)$表示点 p 的 D 领域内的邻居，$N(Nbr(p,D/2))$表示点 p 在 $D/2$ 领域内的邻居数。因为 $N(Nbr(p,D))\geqslant N(Nbr(p,D/2))>k$，即点 p 与其 D 领域内的任意点 q 之间的距离 $dist(p,q)\leqslant D$，本文数据集中各点之间的距离采用欧氏距离的平方。

2.2 数据点与最小边界矩形之间距离的定义

最小边界矩形(minimum bounding rectangle, MBR)[6]表示的是数据集中距离相近点集合的边界值表示的这样的一个集合。δ 维下的数据点 p 表示为 $p=[p_1,p_2,\cdots,p_\delta]$，$\delta$ 维下的边界矩形 R 用其主对角线上的两个端点来表示，两个端点为 $r=[r_1,r_2,\cdots,r_\delta]$，$r'=[r_1',r_2',\cdots,r_\delta']$，且 $r_i\leqslant r_i'$，$1\leqslant i\leqslant n$。

定义 2 数据点 p 与矩形 R 之间的最小距离

$$\mathrm{MINDIST}(p,R)=\sum_{i=1}^{\delta}x_i^2,\ x_i=\begin{cases}r_i-p_i, & \text{当 } p_i<r_i\\ p_i-r_i', & \text{当 } r_i'<p_i\\ 0, & \text{其它}\end{cases}。$$

定义 3 数据点与矩形 R 之间的最大距离

$$\mathrm{MAXDIST}(p,R)=\sum_{i=1}^{\delta}x_i^2,\ x_i=\begin{cases}r_i'-p_i, & \text{当 } p_i<\dfrac{r_i+r_i'}{2}\\ p_i-r_i, & \text{其它}\end{cases}。$$

定义 4 矩形 R 与 S 之间的最小距离

$$\mathrm{MINDIST}(R,S)=\sum_{i=1}^{\delta}x_i^2,\ x_i=\begin{cases}r_i-s_i', & \text{当 } s_i'<r_i\\ s_i-r_i', & \text{当 } r_i'<s_i\\ 0, & \text{其它}\end{cases}。$$

定义 5 矩形 R 与 S 之间的最大距离

$$\mathrm{MAXDIST}(R,S)=\sum_{i=1}^{\delta}x_i^2,\ x_i=\max\{|s_i'-r_i|,|r_i'-s_i|\}。$$

2.3 数据点权重的定义

定义 6 对数据集上的点 p，其权重定义为 $W_k(p)=(\sum_{i=1}^{k}dist(p,n_i(p)))/k$，其中 $n_i(p)$表示离 p 最近的第 i 个邻居，$dist(p,n_i(p))$表示点 p 到它的最近第 i 个邻居的距离，即点 p 的权重为其到最近的 k 个邻居的平均距离。

3 基于加权的离群点检测算法

3.1 算法思想

算法是在大数据集下进行离群点挖掘，其分为两个阶段。第一个阶段是寻找候选划分，由于数据量比较大，为了提高算法的时间效率，开始阶段采用了 BIRCH[12]算法进行预聚类，因为 BIRCH 在聚类数据时对数据量比较大、维数比较高的数据具有较好的时间性能。算法将数据集中的点聚集成一些小的聚类划分，并用聚类中点的 MBR 来代表这些划分，聚类后的划分生成一棵索引树 CF-Tree(此树结构是一棵平衡树，除叶子节点外，中间节点都包含指向其孩子节点和父节点的索引，树叶节点是一个聚类，我们这里称为划分)并通过剪枝方法从这些划分中找到有可能包含离群点的候选划分。第二阶段是寻找离群点，使用加权的方式在候选划分包含的点中找到前 n 个离群点。

3.2 算法描述

3.2.1 候选划分计算阶段

为了找出候选划分，需要计算每个划分 P 距离的上界值($P.upper$)和下界值($P.lower$)，且对划分中包含的任意点 p，有 $P.lower\leqslant D^k(p)\leqslant P.upper$。因此首先介绍 $P.upper$ 和 $P.lower$ 的计算方法：以 MAXDIST 和 MINDIST 为标准寻找 t 个最靠近 P 的划分，t 个划分包含的数据点至少达到 k 个，$P.upper$ 为 t 个划分中离 P 最远的那个划分到 P 的最近距离，而 $P.lower$ 为 t 个划分中离 P 最远的那个划分到 P 的最远距离。

具体计算过程为：输入参数为 Root(包含所有划分的索引)，minDkDist(离群点的最小 D^k 值)。首先用两个集合 $lowerHeap$ 和 $upperHeap$ 来保存满足如下条件的划分，其为与当前划分具有最近的下界值和上界值，且在 $lowerHeap$ 与 $upperHeap$ 中降序排列。然后通过 $lowerHeap$ 和 $upperHeap$ 来更新 $P.lower$ 与 $P.upper$，更新方式为：如果新扫描到的一个划分 Q 与当前划分 P 有 $\mathrm{MINDIST}(Q,P)<P.lower$，则把划分 Q 加入 $lowerHeap$ 中，如果 $lowerHeap$ 中包含的点多于 k 个，则删除其中离 P 最远的划分，此时如果 $lowerHeap$ 中的数据点多于 k 个，则 $P.lower$ 更新为 $lowerHeap$ 中离 P 最远的划分与 P 间的最近距离。$P.upper$ 的更新也是一样。如果 $P.upper$ 值小于 minDkDist，则划分 Q 中不可能包含有离群点，算法停止。算法首先从根节点开始寻找，如果其不是叶子节点，则把当前节点的孩子节点插入以该节点与划分 P 间的距离升序排列的链表中。对于链表中的任意节点，如果其与 P 的最近距离大于 P 的下界值且与 P 的最远距离大于 P 的上界值，则把此节点从链表中删除，这样就可以减少很多不必要的计算。算法具体步骤如下：

输入：CF-Tree 的根节点 Root；离群点的最小 D^k 值 minDKDist；邻居数 k；当前划分 P。

步骤 1 初始化 $P.lower=P.upper=\infty$，$lowerHeap=upperHeap=\Phi$。

步骤 2 从树的根节点开始生成一个节点链表，如果当前节点是叶子节点，则用叶子节点划分 Q 计算 $\mathrm{MINDIST}(P,Q)$和 $\mathrm{MAXDIST}(P,Q)$，否则转到步骤 6。

步骤 3 如果 $\mathrm{MINDIST}(P,Q)\leqslant P.lower$，则把划分 Q 插入以 $P.lower$ 降序排列的 $lowerHeap$ 中；如果 $\mathrm{MAXDIST}(P,Q)\leqslant P.upper$，则把当前划分 Q 中插入以 $P.upper$ 降序排列的 $upperHeap$ 中。

步骤 4 如果 $|lowerHeap|\geqslant k$，则删除 $lowerHeap$ 的队首元素，此时又如果有 $|lowerHeap|\geqslant k$，则 $P.lower=\mathrm{MINDIST}(P,Q)$，其中 Q 为 $lowerHeap$ 中的当前队首划分；如果 $|upperHeap|\geqslant k$，则删除 $upperHeap$ 的队首元素，此时又如果有 $|upperHeap|\geqslant k$，则 $P.upper=\mathrm{MAXDIST}(P,Q)$，其中

Q 为 $upperHeap$ 中的队首划分。

步骤 5　如果 $P.upper \leqslant$ minDKDist，则算法停止，输出 $P.upper$ 与 $P.lower$，否则转到步骤 3。

步骤 6　把当前节点的孩子节点插入节点链表，并以降序排列，转到步骤 2。

在候选划分计算阶段，首先对数据集 DataSet 中的点采用 BIRCH 算法，将数据集中的点聚类成一些划分，并建立一棵索引树 CF-Tree。通过计算每个划分的上界值($P.upper$)和下界值($P.lower$)来寻找候选划分。算法用一个集合 S 来临时保存当前找到的这样一些划分：这些划分有最大的下界值并以该值降序排列，且 S 中至少包含 n 个点，当 S 中的点大于 n 个时更新 minDkDist，其为 S 中划分的最小下界值。当每个划分都计算完后，以最后得到的 minDKDist 值找候选划分并用集合 candSet 存放。candSet 中得到的划分为 $P.upper$ 且比 minDKDist 大的划分。

然后再为 candSet 中的每个划分寻找邻居划分 NPSet，这些邻居划分为与当前划分的最近距离比当前划分的上界值小的划分。之后根据性质 1，这些划分中如果其 minDKDist/2 邻域内的点超过 n 个，则从 candSet 和 NPSet 中删除这些划分，计算候选划分的具体步骤如下：

输入：数据集 DataSet；邻居数 K；离群点数 n。

步骤 1　初始化临时划分集合 S 和 $P.lower$ 的最小值 $\min(D^k)$，使其 $S=\Phi$，$\min(D^k)=0$ 。

步骤 2　对数据集 DataSet 调用 BIRCH 算法，并初始化一棵 CF-Tree。对每个划分 P 计算其 $P.upper$ 和 $P.lower$。

步骤 3　如果 $P.lower \geqslant \min(D^k)$，则 P 并入集合 S 中。

步骤 4　如果 $|S| \geqslant n$，则把 S 中具有最小 $P.lower$ 的划分删除。

步骤 5　如果 $|S| \geqslant n$，则更新 $\min(D^k)$，转到步骤 3，否则转到步骤 6。

步骤 6　对于数据集 DataSet 上的每个划分，如果其 $P.upper \geqslant \min(D^k)$，则 candSet=candSet$\cup${P}。

步骤 7　对 candSet 上的所有划分，如果 $MAX(P,Q) \leqslant P.upper$ 且 $P \in$ canSet，$Q \in$ Pset，则 NPSet=NPSet$\cup\{Q\}$。

步骤 8　如果 $|P \cup \{Q: Q \in candSet \cup NPSet$ 且 $MAXDIST(P,Q) \leqslant$ minDkDist/2$\}| > n$，则 candSet = candSet $-\{P\}$。

步骤 9　输出 canSet 和 NPSet。

3.2.2　离群点检测阶段

为了找到离群点，需要计算每个点的 D^k。在计算 D^k 阶段，先以 candSet 和 NPSet 中的点建立一个 R-tree 结构，此树的结构也是一棵平衡树且与 CF-Tree 类似，然后再以这个结构计算每个点的 D^k 和权重 w_k，其计算方法与上阶段计算划分的上界值和下界值类似。该过程用 $nearHeap$ 来保存当前点 p 的最近 k 个邻居，用 $D^k(p)$ 来跟踪点 p 与其最近的第 k 个邻居间的距离，以 R-tree 结构的根节点 Root 为入口来寻找当前点 p 的最近的 k 个邻居。首先把 Root 节点插入一个节点链表 nodeList 中，如果当前寻找到的节点不是叶子节点，则把当前节点插入以此节点与点 p 间的最小距离 MINDIST 升序排列的节点链表 nodeList 中。算法循环寻找点 p 的最近的 k 个邻居，如果当前点 q 与点 p 间的距离 $dist(p,q) < D^k(p)$，则将点 q 插入以其与点 p 间的距离降序排列的集合 $nearHeap$ 中的相应位置。如果 $nearHeap$ 中的对象数超过 k 个，则删除其离 p 最远的对象，并更新 $p.Dkdist$ 值。此步中有两个减少计算量的优化措施：当寻找到点 p 的最近的第 k 个邻居的距离小于 minDkDist 时，算法则停止计算，此时点 p 不可能是离群点；当从节点链表 nodeList 中循环寻找点 p 的最近的 k 个邻居时，如果当前节点与点 p 间的距离的最小值比当前寻找到的 $D^k(p)$ 还大时，则从节点链表中删除此节点，即不必再计算该节点和其子节点中所包含的点与点 p 的距离，因为它们不可能是点 p 的最近的 k 个邻居点。

在计算前 n 个离群点时先用一个集合 $OutHeap$ 保存当前找到的前 n 个具有最大 D^k 的点，然后继续计算候选点的 D^k 值，如果比当前集合中的点的 minDK($OutHeap$ 中所有点的最小的 D^k 值)值还大，则更新集合 $OutHeap$，并删除集合 $OutPHeap$ 中 D^k 最小的点，如果有几个这样的点，则要删除的点为这几个点中 w_k(权重)最小的点。计算每个点的 D^k 值的具体步骤如下：

输入：R-tree 的根节点 Root，邻居数 k，离群点的最小 D^k 值 minDkDist，当前点 p。

步骤 1　初始化 $D^k(p)=\infty$，$w_k(p)=0$，$nearHeap=\Phi$。

步骤 2　从树的根节点开始生成一个节点链表。如果当前节点是叶子节点，则计算 $dist(p,q)$，q 为叶子节点中包含的数据点。

步骤 3　如果 $dist(p,q) \leqslant D^k(p)$，则把点 q 加入以 $D^k(p)$ 降序排列的集合 $nearHeap$ 中。

步骤 4　如果 $|nearHeap| > k$，则删除集合中离 p 最远的点。此时又如果有 $|nearHeap| = k$，则 $D^k(p)=dist(p,r)$ 且 r 为 nearHeap 中离 p 最远的点。

步骤 5　如果 $D^k(p) \leqslant \min(D^k)$，则返回 $D^k(p)$ 算法结算，否则返回步骤 3。

步骤 6　把当前节点的子节点加入节点链表，并以降序排列，转到步骤 2。

当每点的 $D^k(p)$ 计算得到后，则以如下步骤检测出离群点：

输入：candSet 候选集，NPSet 邻居集，k 邻居数，n 离群点数，p 当前点。

步骤 1　初始化离群点集合 $OutPHeap$ 和 minDkDist，其 minDkDist 表示所有点最小的 D^k，$OutPHeap=\Phi$，minDkDist=0。

步骤 2　用候选划分计算阶段得到的候选划分集和邻居集中的点创建一棵 R-tree。

步骤 3　计算每个数据点的 $D^k(p)$ 和 $w_k(p)$。

步骤 4　如果 $D^k(p) >$ minDkDist，则点 p 有可能是离群点，插入集合 $OutPHeap$ 中。

步骤 5　如果 $|OutPHeap| > n$，则删除集合 $OutPHeap$ 中 D^k 最小的点，如果有几个这样的点，则删除的点为这几个点中 W_k 最小的点。

步骤 6　如果 $|OutPHeap| = n$，则更新 minDkDist=min (D^k)，并返回步骤 3，否则输出 $OutPHeap$。

算法首先计算候选划分，这样可以有效约减数据集中大量的非离群点的数据，在后面计算离群点时就可以有效地避免扫描数据集中大量的数据点。在候选划分中计算离群点时我们引入了数据点权重的概念，对具有相同 D^k 值的数据对象，那些与其最近的 k 个邻居的平均距离(权重)更大的点更可能是离群点，这样就可以更准确地删除输出集合中不可能

是离群点的数据点。

4 实验与结果

在这部分，使用实验来比较我们的算法与传统的 KNN 算法。所有的实验采用平台为 Core 2 Duo 2.00GHz，内存为 2GB 的 PC，操作系统为 Windows XP。

实验 1(算法的有效性) 如图 1 所示，实验数据集为二维模拟数据，包含 1200 条记录，分布于 100 * 100 的区域中。实验中最近邻居参数 $k=10$，离群点参数 $n=12$。与传统的 KNN 算法相比，实验结果如图 1 所示，算法能找到前 12 个离群点。

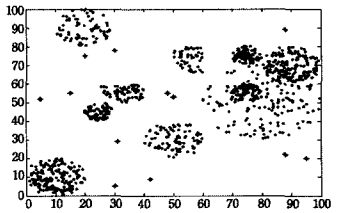

图 1 包含 12 个离群点的二维数据集

实验 2(算法的执行时间) 实验采用一组模拟数据，此数据集上数据产生的概率都相同，且范围都一致。数据量 N 的大小从 100000 到 500000，数据的维数确定为 2，4 和 8，实验设定的邻居参数 $k=100$，需要找到的离群点数 $n=100$，算法执行时间与数据量的关系如图 2 所示。算法对数据量的大小和数据维数的大小具有线性的时间复杂度。通过计算候选划分，使用一些剪枝方法避免了计算数据集中大量的非离群点，从而节省了时间。

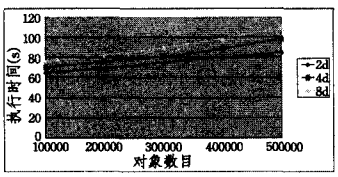

图 2 算法扩展性测试

实验 3(与传统 KNN 算法在执行时间上的比较) 实验用一组模拟数据，数据集上数据产生的概率都相同，且范围都一致。数据量 N 的大小从 100000 到 500000，数据的维数为 2，实验设定的参数 n 和 k 都为 100。实验结果如图 3 所示，虽然我们的基于划分算法 W-KNN 与传统 KNN 的基于划分的算法都具有与数据量的大小 N 线性的时间复杂度，但我们通过性质 1 进一步约减候选划分中不可能包含离群点的划分，缩短了算法的执行时间。

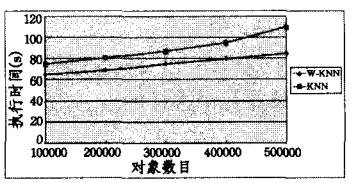

图 3 同原算法执行时间比较

实验 4(与传统 KNN 算法在精确性上的比较) 实验数据集为 Breast Cancer Wisconsin (Original) Data Set[13]，此数据集包含 699 个实例，每个实例包含 10 个属性。数据集中良性数据有 461 条，恶性数据有 238 条，为了使恶性数据成为离群数据，随机去除其中的两百条恶性数据。设置参数 $n=45$，运用本算法，文献[7]和传统 KNN 算法寻找数据集中离群点。实验结果显示与传统的 KNN 方法及只用权重来判断离群点的标准相比较，离群检测的准确度为 98%，而传统的 KNN 方法只有 95%，且只用权重做判断标准时离群检测的精度为 96%。实验证明我们的方法更具精确度。

结束语 本文给出了一种基于加权 KNN 的离群点挖掘算法，通过优化候选划分单元提高算法的效率，并通过实验证明了算法的有效性。由于 KNN 找到的是前 n 个与第 k 个邻居距离最大的点，而一些局部离群点却难以找到，以后的研究方向是使用一种聚类算法把整个数据集聚集成密度分布均匀的不同块，在每块上应用本文离群点挖掘的方法来找到离群点，这样就能有效地找到局部离群点。

参 考 文 献

[1] Baranett V, Lewis T. Outlier in Statistical Data[M]. New York: John Wiley press, 1994

[2] Johnson T, Kwok I, Ng R. Fast Computation of 2-Dimensional Depth Contours[C] // Proc of 4th. Int. Conf. on KDD. New York, 1998: 224-228

[3] Breuing M M, Kriegel H P, Ng R T. LOF: Identifying density based local outliers[C]// Proc of ACM Conference. 1996: 93-104

[4] Birant D, Kut A. Spatio-temporal outlier detection in large databases[C]// Information Technology Interfaces. 2003: 179-184

[5] Knorr E, Ng R. Algorithms for mining distancebased outliers in large datasets[C]// Proc of the 24th Conf on VLDB. New York, 1998: 392-403

[6] Ramaswamy S, Rastogi R, Kyuseok S. Efficient Algorithms for mining outliers from large datasets[C]// Proc of the ACM SIGMOD International Conference on Management of Data. New York: ACM Press, 2000: 93-104

[7] Angiulli F, Pizzuti C. Outlier Mining in Large High-Dimensional Data Sets[C] // Proc of the IEEE Transaction On Knowledge And Data Engineer. VOL. 17, 2005: 1041-4374

[8] Ostermark R. A fuzzy vector valued KNN-algorithm for automatic outlier diction[C] // Proc in the Applied Soft Computing. 2009: 1263-1272

[9] Lee C-P, Lin W-S, Chen Y-M, et al. Gene Selection and sample classification on microarry data based on adaptive genetic algorithm/k-nearest neighbor method[C]// Proc in the Expert Systems with Applicatications. 2010: 07

[10] Maier M, Hein M, von Luxburg U. Optimal construction of k-nearest-neighbor graphs for identifying noisy cluster[C]// Proc in the Theoretical Computer Science. 2009: 1749-1764

[11] Ren Dong-mei, Rahai I, Perrizo W. A vertical distance-based outlier detection method with local pruning[C]// Proc of CIKM'04. Washington, DC, USA: ACM Press, 2004: 279-284

[12] Zhang Tian, Ramakrishnan R, Livny M. An efficient data clustering method for very large databases[C] // Proceedings of the ACM SIGMOD Conference on Management of Data. 1996: 103-114

[13] http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Original%29

[14] 王越，刘亚辉，徐传运. 基于距离和的孤立点用户意义分析算法及应用[J]. 重庆理工大学学报：自然科学版，2010，24(1)：55-59