文章编号:1001 - 9081(2007)01 - 0095 - 03

LDC-mine——基于局部偏差系数的孤立点挖掘算法

张 长,邱保志
(郑州大学 信息工程学院,河南 郑州 450052)
(zhangchang412@163.com)

摘 要:孤立点检测一直是知识发现(KDD)中一个活跃的领域,如信用卡欺诈,入侵检测等。在这些应用领域中研究孤立点的异常行为能够发现隐藏在数据集中更有价值的知识。提出了一个新的度量 LDC(局部偏差系数)因子和基于 LDC 的孤立点挖掘的算法 LDC-mine。实验证明:该算法能够有效地检测出孤立点。

关键词:孤立点检测;局部偏差系数;局部偏差率
中图分类号: TP311 **文献标识码**:A

LDC-mine: algorithm for mining outlier based on local deviation coefficient

ZHANG Chang, QIU Bao-zhi
(*School of Information Engineering, Zhengzhou University, Zhengzhou Henan* 450052, *China*)

Abstract: Outlier detection has always been a hot research field in Knowledge Discovery in Databases (KDD). Finding the rare abnormal behaviors or the outliers can be more interesting than finding the common patterns like credit card fraud, intrusion detection, etc. This paper provided a new Local deviation coefficient (LDC) factor and an algorithm for mining outlier based LDC-mine. The experiment shows that LDC-mine has higher efficiency of detecting outliers.

Key words: outlier detection; local deviation coefficient; local deviation ratio

0 引言

孤立点检测一直是知识发现(KDD)中一个活跃的领域。孤立点检测的研究对象是数据集中偏离绝大多数对象的很小一部分数据。在许多 KDD 应用中,研究孤立点比研究聚类更有用、更重要。因为,在这些应用领域中研究孤立点的异常行为能发现隐藏在数据集中更有价值的知识。诸如,在欺诈探测中,孤立点可能预示着欺诈行为;在市场分析中,可用于确定极低或极高的收入的消费行为;在医疗分析中,用于发现对多种治疗方式的不寻常的反映。对孤立点的挖掘还可应用于金融欺诈、网络监控、数据清洗、电子商务、职业运动员的成绩分析、故障检测、天气预报、医药研究、信贷信用等众多领域。因此,孤立点检测是一个重要的数据挖掘任务,称为孤立点挖掘或异常挖掘。在数据挖掘中,孤立点检测算法大体上可分为以下几类:统计学方法,基于距离的方法,基于偏离的方法和基于密度的方法[1]。

孤立点挖掘可以描述如下:给定一个 n 个数据点或对象的集合,及预期的孤立点的数目 k,发现与剩余的数据相比是显著相异的、异常的或不一致的头 k 个对象。文献[2]提出了基于距离的孤立点挖掘算法,该算法根据某个距离函数计算对象之间的距离。孤立点是那些与剩余的对象相比有更高距离的数据对象。但是上面的定义是从“全局”的角度看待孤立点的,而真实世界中的数据集往往是复杂且分布不均匀的。基于距离的孤立点挖掘算法就很难正确而有效地处理此类数据集。基于密度的方法能够挖掘出基于距离异常算法所不能识别的一类异常数据—局部异常。局部异常观点摈弃了以前所有的异常定义中非此即彼的绝对异常观念,这更加符合现实生活中的应用。近年来,一些研究人员提出局部孤立点探测[3,4,6]的方法,即每个对象赋予某个度,这个度决定了这个对象成为孤立点的程度。每个点的孤立程度只与它和周围点的距离有关,而与数据集中其他的点没有任何关系,这就体现了“局部”的特性。LOF[3] 和 LSC[4] 就是其中两个主要的算法。其中 LSC 是 LOF 的改进。文献[4]提出的基于局部稀疏系数(LSC)孤立点挖掘算法的主要思想是对数据集中每个对象,计算出离它最近 k 对象的距离,并从中选出最大的距离作为该点的 K- 距离,对数据集中每个对象计算出与它的距离不大于该对象 k- 距离的邻近对象形成一个集合,然后计算每个对象与其对应集合中的所有对象之间平均距离的反比,即局部稀疏率,最后计算集合内所有对象的局部稀疏率之和与该点的局部稀疏率比值的平均比率,即局部稀疏系数(LSC);根据每个对象的 LSC 值从大到小的顺序排列整个数据集,并把前 n 个对象作为孤立点。在局部稀疏系数(LSC)算法中,需要计算数据集中每个对象的局部稀疏率和局部稀疏系数,当数据集规模很大时,计算每个对象的局部稀疏率和局部稀疏系数耗费很大的计算量。同时算法中 K 的值取得也很大,从而使得 LSC 算法在处理大规模数据集上的效率很低。

1 LDC-mine 算法

首先,我们定义一些相关概念。

定义 1 对象 p 的 k 距离[3]

对于任何一个正数 k, 对象 p 的 k 距离,即 $k\text{-}distance(p)$,被定义为 $d(p,o)$ 在对象 p 与对象 $o \in D$ 之间必须满足下面的条件,本文的 $d(p,o)$ 指的是 p 与 o 之间的欧式距离。

$$\begin{cases} \text{至少有 } k \text{ 个对象 } o' \in D\backslash\{p\}, & d(p,o') \leqslant d(p,o) \\ \text{至多有 } k-1 \text{ 个对象 } o' \in D\backslash\{p\}, & d(p,o') < d(p,o) \end{cases} \quad (1)$$

收稿日期:2006 - 07 - 06;**修订日期**:2006 - 10 - 08 **基金项目**:河南省科技攻关资助项目(324220066)

作者简介:张长(1980 -),男,河南焦作人,硕士,主要研究方向:数据库、数据挖掘; 邱保志(1964 -),男,河南驻马店人,副教授,主要研究方向:数据库、数据挖掘、人工智能.

定义 2 对象 p 的 k 距离邻居[3]

给定一个对象 p 的 k 距离，p 的 k 距离邻居包括所有的和 p 的距离小于 k 距离的对象。即

$$N_{k\text{-}distance}(p) = \{q \in D \backslash \{p\} \mid d(p,q) \leqslant k\text{-}distance(p)\} \tag{2}$$

这些对象叫做对象 p 的 k 距离邻居。

定义 3 对象 p 的局部偏差率

给出对象 p 的 k 距离邻居，以 p 为圆心，以 k 距离为半径得到一个包含所有 k 距离邻居的圆，计算出 k 距离邻居的质心 p'，然后对象 p 的局部偏差率 $LDR_k(p)$ 为：

$$LDR_k(p) = \frac{dis(p,p')}{\mid N_{k\text{-}distance}(p) \mid} \tag{3}$$

公式(3) 中分子是对象 p 与质心 p' 的欧式距离，而分母代表的是对象 p 的 k 距离邻居的总数。对象 p 的局部偏差率反映了在以 p 为圆心，k 距离为半径的圆内对象集对对象 p 的影响。LDD 的值很大，表明在对象 p 局部范围内的数据点对于 p 的分布是不相关的，则 p 成为孤立点的概率就很大；如果 LDR 为零，说明在对象 p 的周围的数据点是均匀分布，那么 p 成为孤立点的概率就非常小。最终的孤立点判断度量是依靠局部偏差系数，而不是 LDR，那么我们在计算对象的局部偏差系数时，就可以把那些 LDR 为零的对象剪枝。这样就可以节省算法的计算时间，当然这是在牺牲一定精度的条件下，甚至我们在考虑时间效果重于精度效果的条件下，还可以把那些 LDR 为零的对象的 k 距离邻居也剪枝掉，因为从概率上说，那些 LDR 为零的对象很大程度上都属于聚类中的核心对象，那么其的 k 邻居在很大程度上也都应该是聚类中的点。

定义 4 对象 p 的局部偏差系数

给定对象 p 的 k 距离邻居和局部偏差率，p 的局部偏差系数 $LDC_k(p)$ 为：

$$LDC_k(p) = \frac{\sum_{o \in N_K(p)} LDR_k(o)}{\mid N_{k\text{-}distance}(p) \mid} \tag{4}$$

其中，分子是对象 p 的 k 距离邻居集中的对象的 LDR 和。对象 p 的局部偏差系数反映了 p 的 k 距离邻居内邻近对象分散程度。$LDC_k(p)$ 的值越高，表明对象 p 周围的邻居不是群集的，所以 p 成为孤立点的概率非常大；反之，一个低 $LDC_k(p)$ 的值表示 p 周围的邻居是密集的，成为孤立点的概率很低。

算法：LDC-mine( )

Inputs: Data objects, int k

Outputs: Ranked list of k objects with highest LDC

1. Compute the k-distance of each object.
2. Find k-distance neighborhood of each object
3. Compute local deviation ratio of each object by definition 3.
4. Obtain the candidate set.
5. Compute LDC using the candidate set by definition 4.
6. Rank outliers as those with the highest local deviation coefficients end.

// LDC-mine

上面是 LDC-mine 算法的形式化描述。对于每个对象 p，LDC-mine 算法包含六个步骤：

(1) 利用定义 1 计算 p 的 k 距离。

(2) 利用定义 2 计算 p 的 k- 距离邻居。

(3) 根据定义 3 计算 p 的局部偏差率。

(4) 通过剪枝过程，得到孤立点的候选集。在这里，我们可以根据不同的需要采用两种剪枝技术：a) 剪枝那些 LDR 为零的数据对象；b) 剪枝那些 LDR 为零的数据对象和它的 k 邻居对象。

(5) 对于候选集，根据定义 4 计算每个对象的局部偏差系数。

(6) 通过局部偏差系数，排序数据集，前 k 个数据对象就是我们要挖掘的孤立点。

由于孤立点的个数远小于数据集的规模，所以可以在第 (6) 步中，遍历孤立点的候选集数据，每次得到最大的 LDC 值，然后输出。由于 $k \ll n$，所以在时间上更高效。

2 实验与分析

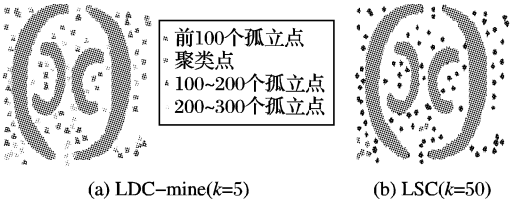
本文实验环境是 P4 2.93GHz，256MB 内存，Windows XP 专业版操作系统。算法在 VC6.0 环境下用C++语言实现。

2.1 算法复杂度分析

该算法第一步的时间复杂度为 $O(kN^2)$，第二步为 $O(N^2)$，第三步为 $O(kN)$，第四步为 $O(N)$，第五步为 $O(kM)$，第六步为 $O(kM)$，这里，k 是每个数据对象的最小邻居数，N 是数据集的规模，M 是候选集的规模，通常在进行完剪枝过程后，M 要远远小于 N。所以说，LDC-mine 算法的总的时间复杂度为 $O(kN^2)$。算法的时间复杂度与每个对象的最近邻居个数成线性关系，与数据集的规模呈平方关系。LSC 算法的时间复杂度也是 $O(kN^2)$。

2.2 算法正确性分析

综合数据集：为了验证该算法的正确性，我们首先用文献[5]中的 2 维的 Data Set 4 来做实验。该数据集有 8486 个记录，4 个大簇和一些分布不均匀的孤立点数据。图 1 为运行 LDC-mine 算法和 LSC 算法后的效果比较图。在 LDC-mine 中，参数 k 设置为 5，而在 LSC 中，参数 k 必须至少等于 50 才能正确的辨别出大部分的孤立点。而在我们的实验中，参数 k 的设置不能过大，因为如果 k 过大的话，那么在局部的范围内对于孤立点的影响将可能相互抵消而导致无法正确地辨别出来。从图 1 中，我们可以看到凡是 LSC 算法检测出的孤立点，LDC-mine 算法也都可以被检测出来，甚至 LSC 算法检测不出来的孤立点，LDC-mine 算法也能够检测出来。所以实验表明 LDC-mine 孤立点挖掘算法是行之有效的，同时我们可以发现数据的 LDC 值从高到低的过程中，孤立点的分布逐渐地向大簇靠拢，这也体现了“局部”的概念。



(a) LDC-mine(k=5) (b) LSC(k=50)

图 1 LDC-mine 算法和 LSC 算法效果比较

(a) LDC-mine(k=10) (b) LSC(k=10)

图 2 LDC-mine 算法和 LSC 算法效果比较

然后，再使用一个相对复杂的综合数据集 DB2，该数据集中一共有 23724 个数据点，其中有 6 个大簇和很多的孤立点和噪音数据。其中还有一条很强的干扰噪音数据线。对于这样分布十分复杂的数据集，传统的 LSC 算法就很难发现其中

的孤立点和噪音数据。而 LDC-mine 算法就可以有效地发现它们。图 2 就是 LDC-mine 算法和 LSC 算法对于数据集 DB2 的效果比较。很明显,LSC 算法对于该数据集的孤立点发现可以说是失败的,所以 LDC-mine 算法在发现复杂分布的孤立点上比 LSC 算法更加健壮,更加有效。

真实数据集:数据集是来自美国国家曲棍球联盟(NHL)1998 年球员统计数据集[7],该数据集一共有 816 条记录,16 个属性。其中包括一些未知数目的孤立点。我们从整个统计数据集中选出 4 个属性(name, total score, plus/minus, penalty minutes)来组成实验数据集。除此之外,还给每一条记录赋予独立的 ID 号以便评价结果。我们感兴趣的是 top_n 条记录为孤立点。实验中,参数的设置为 $k = 5$,$top\_n = 10$,表 1 为前十个记录的 LDC 值。

表 1　数据集执行 LDC-mine 后的前 10 条记录的 LDC 值

ID	539	697	174	414	172	440	173	190	603	80
LDC	3.171	3.007	2.966	2.531	2.466	2.368	2.344	2.302	2.289	2.202

从表 1 可以看出,ID 号为 539 的记录为最强的孤立点,原因是该名球员在被罚出场相对长的情况下,仍然进了相当多的球。ID 为 697 的球员因为同样的理由而被LDC-mine算法检测出来。其次,往下看数据记录可知,前 10 条记录的 LDC 的值相对较大,对应的 10 名球员都是相比于正常机制下的异常点。所以,通过对真实数据集的测试,LDC-mine 算法在发现孤立点方面是有效的。

2.3　算法的时间分析比较

数据规模与时间的关系对比:在比较数据规模与时间的关系实验中,还是使用图 2 中的 2 维数据集,数据集的规模即 N 从 1 000 变化到 8 000,每次递增 1 000,参数 k 的设置为 10,我们采用的剪枝方法是仅仅剪去那些 LDR 为零的数据对象。图 3 是实验结果的比较。

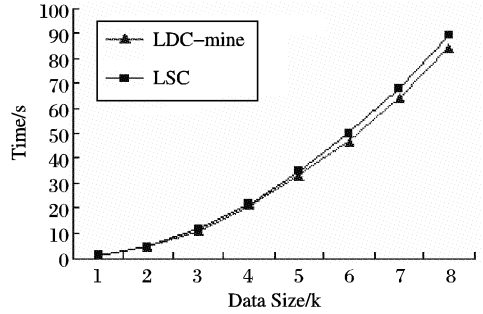


图 3　LDC-mine 和 LSC 对于不同数据规模执行时间对比

从图 3 可以看出,LDC-mine 算法运行时间要低于 LSC 算法的运行时间,这也符合我们的时间复杂度的分析。当在剪枝过程中也剪去那些 LDR 为零的数据对象的 k 邻居的时候,运行时间一定会比 LDC-mine 算法更低。同时需要说明的是,在数据集规模大于 6 000 之后,对于 LSC 算法,要想检测出大多数孤立点来说,$k = 10$ 是不合适的,k 需要一个相对大一些的值,如最小为 50,但是对于 LDC-mine 算法来说,k 的值不需要相对大的变化就能检测出绝大多数的孤立点。这就说明,虽然 LDC-mine 和 LSC 算法的时间复杂度相同,但是 LDC-mine 比 LSC 在最近邻个数上的选择要更加合理。所以随着数据集规模的增大,在发现孤立点的正确率和执行时间上,LDC-mine 算法比 LSC 算法要更加高效。

3　结语

本文提出了基于局部偏差系数的孤立点挖掘算法 LDC-mine,该算法能够很好地检测出数据集中的异常点。实验表明,该算法在识别孤立点上,比传统的 LSC 算法更加有效。未来的工作是进一步研究在高维空间中孤立点的识别方法以及如何提高在真实数据集上的精度问题。

参考文献:

[1] HAN JW, KAMBER M. Data Ming: Concepts and Techniques[M]. Morgan Kaufmann Publishers Press. 2001.

[2] KNORR EM, NG RT. Algorithms for Mining Distance-Based Outliers in Large Datasets[J]. Proceedings of the 24th VLDB Conference[C]. 1998. 392 - 403.

[3] BREUNIG MM, KRIEGEL HP, NG RT, *et al.* LOF: Identifying density-based local outliers[A]. Proceedings of ACM SIGMOD International Conference on Management of Data[C]. 2000. 93 - 104.

[4] MALIK A, EZEIFE CI. LSC - mine: Algorithm for Mining Local Outliers[A]. Proceedings of the 15th Information Resource Management Association (IRMA) International Conference[C]. 2004. 5 - 8.

[5] HSU CM, CHEN MS. Subspace Clustering of high dimensional spatial data with noises[A]. PAKDD 2004, LNAI 3056[C]. 2004. 31 - 40.

[6] CHIU AL, FU AW. Enhancements on Local Outliers Detection[A]. Proceedings of the 7th International Database Engineering and Application Symposium[C]. 2003. 298 - 307.

[7] The Internet Hockey Database[EB/OL]. http://www.hockeydb.com/. Player Statistics for 1998 - 1999, 2006 - 06 - 08.

(上接第 57 页)

3　结语

针对分布式多传感器环境,研究了目标识别应用的实现技术,提出了一种基于加权 D-S 证据理论组合规则的决策融合方法。通过分析多传感器目标识别系统的信息模型,指出传感器的决策可信度由其被支持度及与目标间的距离确定。并且对于活动目标,传感器的决策可信度是变化的。该可信度可体现为加权 D-S 证据理论组合规则中的证据权值,根据传感器支持度及其与目标的距离,给出了一种权值确定方法。最后的仿真实验表明,与传统 D-S 证据理论组合规则相比,该方法提高了融合效率,能较快速地完成识别任务。

参考文献:

[1] WINDEATT T, GHADERI R. Binary Labeling and Decision-level Fusion[J]. Information Fusion, 2001, 2(2): 103 - 112.

[2] WU H, SIEGEL M, ABLAY S. Sensor Fusion Using Dempster-Shafer Theory II: Static Weighting and Kalman Filter-like Dynamic Weighting[A]. Proceeding Instrumentation and Measurement Technology Conference[C]. 2003. 907 - 912.

[3] BUEDE DM, GIRARDI P. A Target Identification Comparison of Bayesian and Dempster-Shafer Multisensor Fusion[J]. IEEE Transaction of System, 1997, 27(5): 569 - 577.

[4] 李秋华, 李吉成, 沈振康. 一种基于多传感器时间—空间信息融合的红外小目标识别方法[J]. 红外与毫米波学报, 2002, 21(3): 209 - 212.

[5] 段新生. 证据理论与决策、人工智能[M]. 北京, 中国人民大学出版社, 1993.

[6] DUARTE M, HU YH. Vehicle Classification in Distributed Sensor Networks[J]. Journal of Parallel and Distributed Computing, 2004, 64(7): 826 - 838.