文章编号:1001 - 9081(2010)05 - 1284 - 03

# 动态数据环境下基于信息熵的相对离群点检测算法

孙　浩[1]，何晓红[2]
(1. 重庆邮电大学 移通学院,重庆 400065；　2. 重庆邮电大学 生物信息学院,重庆 400065)
(sunhao2001@163.com)

**摘　要**:在基于信息熵的离群点检测算法的基础上,提出一种适用于动态数据环境的检测算法。该算法在有数据对象插入或删除的时候,不必计算所有数据对象的相对离群点因子(ROF)值,而只需重新计算受影响的点的 ROF 值。实验结果表明,该算法在动态数据环境下的运行时间小于原来的算法。

**关键词**:动态数据环境;信息熵;离群点检测;局部离群因子
**中图分类号**: TP311　　**文献标志码**:A

# Entropy-based algorithm to detect relative outliers in dynamic environment

SUN Hao[1], HE Xiao-hong[2]
(1. *College of Mobile Telecommunications, Chongqing University of Posts and Telecommunications, Chongqing* 400065, *China*;
2. *College of Bio-Information, Chongqing University of Posts and Telecommunications, Chongqing* 400065, *China*)

**Abstract:** An algorithm for detecting relative outliers in dynamic environment based on information entropy was proposed. When an object was inserted into or deleted from the dataset, the algorithm made it unnecessary to compute the values of Relative Outlier Factor (ROF) for all objects in dataset, only need to compute for affected objects. The experimental results indicate that the running time of this algorithm is less than that of the original algorithm in dynamic environment.

**Key words:** dynamic environment; entropy; outlier detection; Local Outlier Factor (LOF)

## 0　引言

离群点(Outlier)检测是数据挖掘领域研究的热点问题之一。Hawkins 的定义[1]揭示了离群点的本质:“一个离群点是一个观察点,它偏离其他观察点如此之大,以至引起怀疑是由不同机制产生的”。离群点检测与关联规则发现、分类与聚类等技术所关注的对象不同,其任务是从大量的、复杂的数据集中发现小部分异常数据所隐含的、与常规数据模式显著不同的数据模式,在欺诈甄别、贷款审批、气象预报、客户分类等方面有着广泛的应用。

近年来,国内外的学者针对离群点检测技术开展了专门的研究工作。较早的离群点检测算法大致可以分为基于统计的算法、基于距离的算法和基于偏差的算法等。基于密度的离群点定义算法是在基于距离的基础上建立起来的,将点之间的距离和给定范围内点的个数这两个参数结合起来得到密度的概念[2]。Breunig 等人提出一种基于密度的局部离群点检测算法[3],该算法用局部离群因子(Local Outlier Factor, LOF)来表征数据集中每个数据对象的离群度数。LOF 计算的时间复杂度较高,很难将其应用于维数较多的空间数据库。在 SLOM[4]算法中,将空间数据分为了空间属性和非空间属性,利用空间属性及空间邻接关系确定对象的邻域,以邻域距离 $d$ 和波动因子 $\beta$ 的乘积为空间局部离群度,即 $SLOM = d \times \beta$。但该算法中,由于波动因子仅由对称分布状况来决定,在空间邻居较少或波动幅度较小的情况下难以准确表征波动情况,会导致漏检和误检的现象。

空间数据库中,由于离散属性值之间并没有连续属性数据之间那样固有的度量关系,用于连续属性数据的离群点检测算法在离散属性值的数据里面不能工作。Barbara 等人把信息熵运用于离散属性数据集聚类[5],结果表明将信息熵运用于离散属性数据集是有效的。于绍越等人提出了基于信息熵的相对离群点的检测算法 ENBROD[6],该算法引入了去一划分信息熵增量,在此基础上给出了每个对象所对应的相对离群点因子(Relative Outlier Factor, ROF)的定义。但该算法只适用静态环境,如果数据集中的数据发生变化,则需要重新计算所有数据对象的 ROF,所以该算法在动态数据库中的应用受到了一定的限制。本文在此基础上,提出了一种在动态环境下的相对离群因子检测算法,能够较好地解决 ENBROD 在动态数据环境下的缺陷。

## 1　ROF 的定义

熵是热力学中微观状态多样性或均匀性的一种度量,它反映了系统微观状态的分布几率。将热力学几率扩展到系统各个信息源信号出现的几率就形成了信息熵,信息熵标志着所含信息量的多少,是对信息和随机变量的不确定性的一种度量[7]。

文献[6]中给出了信息熵的计算方法以及去一划分信息增量的定义。

**定义 1**　信息熵。如果 $X$ 是一个随机变量,$S(X)$ 是 $X$ 的取值集合,$p(X)$ 是 $X$ 的概率函数,则信息熵 $E(X)$ 表示如下:

$$E(X) = -\sum_{x \in S(X)} p(X) \log(p(x)) \tag{1}$$

**定义 2**　多维随机变量 $\hat{x} = \{x_1, x_2, \cdots, x_n\}$ 的信息熵为:

$$E(\hat{x}) = -\sum_{x_1 \in S(X_1)} \cdots \sum_{x_n \in S(X_n)} p(x_1, x_2, \cdots, x_n) \log p(x_1, x_2, \cdots, x_n) \tag{2}$$

假设数据集中的数据属性间彼此独立,则式(2)可以转化为:

**收稿日期**:2009 - 12 - 08;**修回日期**:2010 - 01 - 26。　**基金项目**:重庆邮电大学自然科学基金资助项目(A2008-04)。

**作者简介**:孙浩(1977 - ),男,四川巴中人,讲师,硕士,主要研究方向:离群知识发现、聚类分析;　何晓红(1977 - ),女,四川巴中人,副教授,博士,主要研究方向:生物信息学。

$$E(\hat{x}) = -\sum_{x_1 \in S(X_1)} \cdots \sum_{x_n \in S(X_n)} p(x_1,x_2,\cdots,x_n)\log(p(x_1)\cdots p(x_n)) = E(X_1) + E(X_2) + \cdots + E(X_n) \quad (3)$$

**定义 3**　去一划分信息熵增量。设 $D$ 是对象集合,$S$ 是 $D$ 的一个子集且 $|S|>1$,$p$ 是 $S$ 中的一个对象,即 $p \in S$。把 $S$ 划分成 $S\backslash\{p\}$ 和 $\{p\}$ 两个簇,记作 $\hat{C} = \{C_1, C_2\}$,得 $E(\hat{C}) = \sum_{k=1,2}\left(\frac{|C_k|}{|D|}E(C_k)\right)$,则子集 $S$ 的去一划分信息熵的增量为:

$$\Delta p = \Delta(p,S) = E(S) - E(\hat{C})$$

**定义 4**　设数据集为 $D$,对任意正整数 $k(k \geq 2)$,对象 $o$ 的 $k$- 两对象信息熵,记作 $k\text{-}ce(o)$,满足:

1)至少有 $k$ 个对象 $o' \in D$,使得 $E(o,o') \leqslant k\text{-}ce(o)$;

2)至多有 $k-1$ 个对象 $o' \in D$,使得 $E(o,o') < k\text{-}ce(o)$。

**定义 5**　对象 $o$ 的 $k$ 邻域,记作 $N_{k\text{-}ce}(o) = \{o' \mid o' \in D \wedge E\{o,o'\} \leqslant k\text{-}ce(o)\}$。

**定义 6**　相对离群因子 ROF 的值 $S_{\mathrm{ROF}}$。

$$S_{\mathrm{ROF}}(o,MinPts) = \frac{\sum_{p \in N_{k\text{-}ce}(o)}\left(\frac{\Delta p}{\Delta o}\right)}{k} = \frac{\sum_{p \in N_{k\text{-}ce}(o)}\Delta p}{\Delta o \times k}$$

ENBROD 算法计算各个对象的 $S_{\mathrm{ROF}}$ 值,来表示各个对象的离群度。

## 2　动态数据环境下的离群点检测算法

在 ENBROD 算法中,每个对象的 $S_{\mathrm{ROF}}$ 都与该对象所处的环境相关,当在数据集 $D$ 中增加、删除或修改数据时,都会影响相关对象的 $S_{\mathrm{ROF}}$,必须对所有的对象重新计算 $S_{\mathrm{ROF}}$。事实上,当更新数据的时候,只会影响个别数据而不是所有对象的 $S_{\mathrm{ROF}}$。杨风召等人在文献[3]提出的 LOF 的基础上,提出了一种在动态环境下局部异常挖掘的增量算法 IncLOF[8],当数据库的数据更新时,只对受影响的点进行重新计算。受此启发,本文采用一种只对受影响的对象重新计算 $S_{\mathrm{ROF}}$。

### 2.1　受影响对象

由于数据的修改可以看成先删除数据,然后再插入数据,因此本文只讨论数据的插入和删除这两种情况。

设 $\forall x \in D$,数据 $p$ 的插入或删除会影响 $N_{k\text{-}ce}(x)$ 的值,从而引起 $S_{\mathrm{ROF}}(x,MinPts)$ 的变化。$\exists y \in D$,如果 $x \in N_{k\text{-}ce}(y)$,则还会引起 $S_{\mathrm{ROF}}(y,MinPts)$ 的变化。

**定义 7**　受影响对象的集合。设数据集为 $D$,$p$ 为任意对象,当向 $D$ 中插入或删除 $p$ 而使 ROF 产生变化的集合:

$$Affect(p) = \{q \in D\backslash\{p\} \mid p \in N_{k\text{-}ce}(q)\} \cup \{u \in D\backslash\{p\} \mid v \in N_{k\text{-}ce}(u) \wedge p \in N_{k\text{-}ce}(v) \wedge v \in D\backslash\{p,u\}\} \cup \{u \in D\backslash\{p\} \mid v \in N_{k\text{-}ce}(u) \wedge w \in N_{k\text{-}ce}(v) \wedge p \in N_{k\text{-}ce}(w) \wedge v \in D\backslash\{p,u\} \wedge w \in D\backslash\{p,u,v\}\}$$

### 2.2　数据插入时的离群点检测算法

当一个数据对象插入到数据集 $D$ 中,需要重新计算 $S_{\mathrm{ROF}}$ 的数据点有 $p$ 和 $Affect(p)$。对于 $\forall o \in Affect(p)$,$N_{k\text{-}ce}(o)$ 都需要更新,如图 1 所示。

数据对象 $p$ 插入时,当 $d(p,q) = k\text{-}ce(q)$ 时,如图 1(a)所示,则直接将数据点 $p$ 插入到 $q$ 的邻域里面去;当 $d(p,q) < k\text{-}ce(q)$ 时,如果至多 $\exists k-2$ 个对象 $o' \in D\backslash\{q\}$,满足 $d(o',q) < k\text{-}ce(q)$,如图 1(b)所示,则直接将数据点 $p$ 插入到 $q$ 的邻域里面去;如果 $\exists k-1$ 个对象 $o' \in D\backslash\{q\}$ 满足 $d(o',q) < k\text{-}ce(q)$,如图 1(c)所示,则将数据点 $p$ 插入到 $q$ 的邻域里面去,同时删除 $N_{k\text{-}ce}(q)$ 中离 $q$ 最远的数据点。

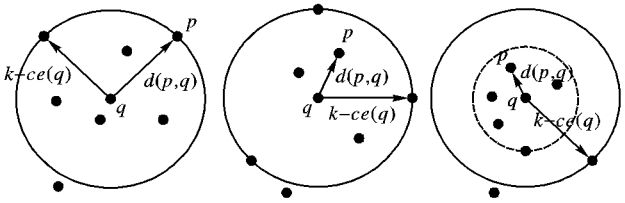

图 1　插入对象时,受影响点邻域变化情况($k=5$)

算法　UpdateENBROD。

Input:数据集 $D$,插入对象 $p$,邻域大小 $k$;

Output:更新后数据集的 $S_{\mathrm{ROF}}$。

```
Procedure:
/* 步骤 1:初始化受影响对象的集合 affectList */
affectList = ∅;
/步骤 2: * updateKNN( ) 找到 k 邻域发生变化的数据点,并更新其邻域表 */
KNNChangeList = updateKNN (D, p);
For each o in KNNChangeList
{
  if (q in N_{k-ce}(o))
  {addAffectList(q);}
}
/* 步骤 3:更新受影响对象的去一划分信息增量 */
for each o in affectList
{
  计算对象 o 的邻域所对应的 E(D) 和 E(Ĉ) 的值;
  计算 Δ(o, N_{k-ce}(o));
}
/* 步骤 4:更新 S_ROF */
for (i = 0; i < affectList. Count; i++)
{
  sum = 0;
  for each o' in N_{k-ce}(o)
  { sum += Δ(o');}
  S_ROF[i] = sum / (k * Δ(o));
}
```

### 2.3　数据删除时的离群点检测算法

数据对象 $p$ 从数据集 $D$ 中删除时,需要重新计算 ROF 值的数据点为 $Affect(p)$,删除数据对象时的算法描述同 UpdateENBROD,只是在函数 updateKNN ($D$, $p$) 的处理方式上不同:如果对象 $o$ 满足 $p \in N_{k\text{-}ce}(o)$,则将其加入到 $k$ 邻域变化的集合 $KNNChangeList$,并更新这些点的 $k$ 邻域。更新时受影响点的邻域变化情况如图 2 所示。

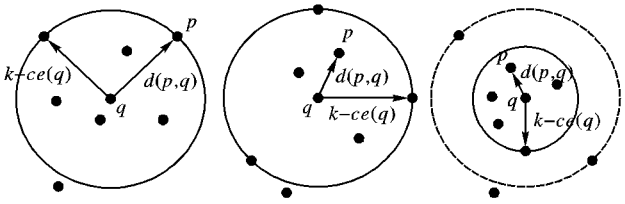

图 2　删除对象时,受影响点邻域变化情况($k=5$)

删除数据对象 $p$ 时,当 $d(p,q) = k\text{-}ce(q)$,且 $|N_{k\text{-}ce}(q)| > k$ 时,如图 2(a)所示,此时直接将数据对象 $p$ 删除;当 $d(p,q) < k\text{-}ce(q)$,且 $|N_{k\text{-}ce}(q)| > k$ 时,如图 2(b)所示,此时直接删除数据对象 $p$;当 $d(p,q) < k\text{-}ce(q)$,且 $|N_{k\text{-}ce}(q)| = k$,如图 2 中(c)所示,此时删除数据对象 $p$ 的同时,重新计算 $N_{k\text{-}ce}(q)$,其他离 $q$ 最近的点(可能不只一个)会补充到 $q$ 的 $k$ 邻域中。

## 3 实验及其结果分析

文献[6]采用了 UCI 数据集[9]中 16 个属性的 Zoo 数据和 9 个属性的 Winsconsin breast cancer 数据，本文为了在更复杂的数据上比较 UpdateENBROD 和 ENBROD 算法，采用 UCI 数据集中的 Thyroid disease records 数据。该数据集中有 3 772 条记录，每个记录有 30 个属性值，被标记为 negative(3 540 个记录，占 93.8%)和 sick(232 个记录，占 6.2%)。实验中，在标记为 negative 的数据中选分别随机选出 100，200，…，600 条数据，求出每个数据对象的 $S_{ROF}$，然后在标记为 sick 的数据中随机选出几条数据插入到前面的数据集中作为异常数据记录，分别采用 UpdateENBROD 和 ENBROD 求出每个数据对象的 $S_{ROF}$；然后随机删除一些数据对象，再次采用 UpdateENBROD 和 ENBROD 分别求出每个数据对象的 $S_{ROF}$。实验在 CPU 为 Intel core 2 T5500 1.66 GHz，内存 2 GB 的 PC 机上进行，算法用C++实现。

表 1 给出了算法 UpdateENBROD 和 ENBROD 在数据记录数为 600，$k$ = 80 发现所有异常记录时排序后的最大记录数。

**表 1 发现所有异常记录时的最大记录数**

| 异常数据的条数 | 发现所有异常数据记录时的最大记录数 | |
|---|---|---|
| | UpdateENBROD | ENBROD |
| 3 | 3 | 3 |
| 5 | 5 | 5 |
| 10 | 11 | 11 |

图 3 给出了当数据集中有数据对象更新时，算法 UpdateENBROD 和 ENBROD 在不同数据记录数下的运行时间曲线图。

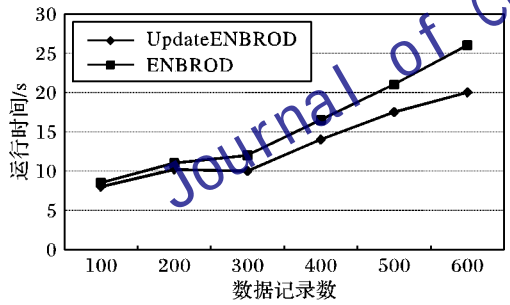


图 3 UpdateENBROD 和 ENBROD 的运行时间曲线图

由表 1 可以看出，当数据集里有 3 条和 5 条离群数据记录时，UpdateENBROD 与 ENBROD 都能准确地发现异常的数据，当数据集中插入了 10 条异常数据，两个算法都在前 11 条里面发现了所有的异常数据，所以 UpdateENBROD 与 ENBROD 都能较高效地发现离群数据的记录，检测离群数据的能力完全一样，从而说明了算法 UpdateENBROD 的正确性。当数据集中的数据发生动态更新的时候，从图 3 可以看出，当数据集的记录数不大的时候，两个算法的运行时间相差不大，随着数据集记录数的增加，UpdateENBROD 的运行时间小于 ENBROD。

## 4 结语

在研究 ENBROD 算法的基础上，本文提出了一种动态数据环境下的离群点检测算法，当数据集中的数据存在一些更新操作的时候，不必重新计算所有数据点的 ROF 值，而只需要计算受影响点的 ROF 值。实验表明，该算法在动态数据环境下的时间性能优于 ENBROD。

**参考文献：**

[1] HAWKINS D. Identification of outliers [M]. Berlin: Springer-Verlag, 1980.

[2] 薛安荣，鞠时光，何伟华，等. 局部离群点挖掘算法研究[J]. 计算机学报，2007，30(8)：1455－1463.

[3] BREUNIG M M, KRIEGEL H-P, NG T, *et al.* LOF: Identifying density-based local outliers [J]. ACM SIGMOD Record, 2000, 29(2): 93－104.

[4] CHAWLA S, SUN PEI. SLOM: A new measure for local spatial outliers [J]. Knowledge and Information Systems, 2006, 9(4): 412－429.

[5] BARBARÁ D, LI YI, COUTO J. Coolcat: An entropy-based algorithm for categorical clustering [C]// Proceedings of the 11th International Conference on Information and Knowledge Management. New York: ACM Press, 2002: 582－589.

[6] 于绍越，商琳. 基于信息熵的相对离群点的检测方法：ENBROD [J]. 南京大学学报：自然科学版，2008，44(2)：212－218.

[7] SHANNON C E. A mathematic theory of communication [J]. Bell System Technical Journal, 1948, XXVH(3): 379－423, 623－656.

[8] 杨风召，朱扬勇，施伯乐. IncLOF：动态环境下局部异常的增量挖掘算法[J]. 计算机研究与发展，2004，41(3)：477－484.

[9] MURPHY P M, AHA D W. UCI repository of machine learning database[EB/OL]. (1994－06) [2009－10－06]. http://www.ics.uci.edu/mlearn/MLRepository.html.



(上接第 1283 页)

**参考文献：**

[1] LEE D D, SEUNG H S. Learning the parts of objects by nonnegative matrix factorization [J]. Nature, 1999, 401(6755): 788－791.

[2] XU WEI, LIU XIN, GONG YIHONG. Document clustering based on non-negative matrix factorization [C]// Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. New York: ACM Press, 2003: 267－273.

[3] GUILLAMET D, VITRIA J. Non-negative matrix factorization for face recognition [C]// Proceedings of the 5th Catalonian Conference on AI: Topics in Artificial Intelligence, LNCS 2504. Berlin: Springer-Verlag, 2002: 336－344.

[4] VIRTANEN T. Monaural sound source separation by non-negative matrix factorization with temporal continuity and sparseness criteria [J]. IEEE Transactions on Audio, Speech, and Language Processing, 2007, 15(3): 1066－1074.

[5] DONOHO D, STODDEN V. When does nonnegative matrix factorization give a correct decomposition into parts? [C]// Proceedings of the 17th Annual Conference Neural Information Processing Systems. Cambridge, MA, USA: MIT Press, 2003: 1141－1148.

[6] LEE D D, SEUNG H S. Algorithms for non-negative matrix factorization[C]// Advances in Neural Information Processing Systems. Cambridge, MA, USA: MIT Press, 2001, 13: 556－562.

[7] CAO BIN, SHEN DOU, SUN JIAN-TAO, *et al.* Detect and track latent factors with online nonnegative matrix factorization [C]// Proceedings of the 20th International Joint Conference on Artificial Intelligence. San Francisco, CA, USA: Morgan Kaufmann Publishers, 2007: 2689－2694.

[8] KLINGENBERG B, CURRY J, DOUGHERTY A. Non-negative matrix factorization: Ill-posedness and a geometric algorithm[J]. Pattern Recognition, 2009, 42(5): 918－928.