文章编号:1001 - 9081(2005)06 - 1353 - 04

一种面向高维混合属性数据的异常挖掘算法

李庆华[1],李 新[1],蒋盛益[1,2]
(1. 华中科技大学 计算机科学与技术学院,湖北 武汉 430074;
2. 衡阳师范学院 计算机系,湖南 衡阳 421008)
(Sandylee712@126.com)

摘 要:异常检测是数据挖掘领域研究的最基本的问题之一,它在欺诈甄别、气象预报、客户分类和入侵检测等方面有广泛的应用。针对网络入侵检测的需求提出了一种新的基于混合属性聚类的异常挖掘算法,并且依据异常点(outliers)是数据集中的稀有点这一本质,给出了一种新的数据相似性和异常度的定义。本文所提出算法具有线性时间复杂度,在 KDDCUP99 和 Wisconsin Prognosis Breast Cancer 数据集上的实验表明,算本法在提供了近似线性时间复杂度和很好的可扩展性的同时,能够较好的发现数据集中的异常点。

关键词:异常检测;聚类;数据挖掘
中图分类号: TP311.13;TP18 **文献标识码**:A

New approach for outlier detection in high dimensional dataset with mixed attributes

LI Qing-hua[1], LI Xin[1], JIANG Sheng-yi[1,2]
(1. *Department of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan Hubei* 430074, *China*;
2. *Department of Computer Science, Hengyang Normal University, Hengyang Hunan* 421008, *China*)

Abstract: The outlier detection problem has important applications in the fields of fraud detection, weather prediction, customer segmentation1 and intrusion detection. Many recent algorithms use concepts of proximity in order to find outliers based on their relationship to the rest of the data. In this paper we proposed a new algorithm to detect outlier in high dimensional domains with mixed attributes based on clustering, and proposed a new method to measure similarity and outlyingness of objects. The algorithm we proposed can give near linear performance. The experimental results on KDDCUP99 and Wisconsin Breast Cancer dataset show that our algorithm is not only effective and scalable but also leads to reasonable good accuracy.

Key words: outlier detection; clustering; data ming

0 引言

异常(Outlier)指的是在数据集中与大部分数据有明显差异的数据。Hawkins 将异常定义为"异常是在数据集中偏离大部分数据的数据,使人怀疑这些数据的偏离并非由随机因素产生,而是产生于完全不同的机制"[1]。异常检测是数据挖掘领域研究的最基本的问题之一,它用来发现数据集中与其他数据显著不同的对象. 异常检测目前已成为数据挖掘的一个重要方面。

异常挖掘可以描述为:对于给定 N 个数据对象和数据异常性的定义,发现数据异常性最大的前 n 个对象或者数据异常性大于给定阈值的所有数据对象。异常挖掘问题由两个子问题构成:

1) 定义在一个数据集中什么样的数据是不一致或异常的数据(即定义异常性);

2) 给出挖掘所定义的异常数据的有效方法。

由对异常数据定义的不同,异常检测的算法主要可以分为三大类:

1) 基于统计的方法[2,3]。根据统计方法对数据对象创建一个统计描述,通过统计描述假设一个分布概率模型,然后根据这些模型采用不一致检验的方法来确定异常数据。它的应用需要事先知道数据集参数,分布参数和离群数据的个数。这种方法对数值性数据有效,但对高维数据、周期性数据、分类数据等比较困难。

2) 基于偏差的方法[4~6]。通过对一组对象的主要特征进行检查来识别异常数据的,偏离特征描述的对象被认为是异常,这种方法可以对各种形式的数据进行离群检测,但需要事先知道数据的特性,以确定相异函数;

3) 基于距离的方法。通常是通过计算数据对象间的距离来确定异常数据。基于距离的方法又可以分为基于最近邻的方法[7,8]、基于密度的方法[9,10]和基于聚类的方法[11~13]三类。基于最近邻的方法通常用在某一给定距离内的邻接点的个数来度量对象的异常程度;基于密度的方法通常用局部异常因子[9] LOF(Local Outlier Factor)来度量对象的异常程度;

收稿日期:2004 - 11 - 12;修订日期:2005 - 01 - 20 基金项目:国家自然科学基金资助项目(60273075)
作者简介:李庆华(1940 -),男,湖北武汉人,教授,博士生导师,主要研究方向:并行处理、数据挖掘; 李新(1977 -),男,湖南益阳人,硕士研究生,主要研究方向:数据挖掘、并行处理; 蒋盛益(1963 -),男,湖南隆回人,副教授,博士研究生,主要研究方向:数据挖掘、算法设计.

基于聚类的方法通常用数据所属类的大小来度量对象的异常程度。本文所介绍的异常挖掘算法是一种基于聚类分析的方法。

1 算法描述和分析

数据挖掘也叫知识发现(Knowledge Discovery from Database, KDD)。聚类分析是数据挖掘的方法之一,它将数据集中的数据进行分组,使得同一类内的数据相似性尽量大,不同类之间的数据相似性尽量小。相似度是用来判断二个数据之间的差异程度,本文中使用"距离"来描述数据之间的相似程度。

在实际问题中正常记录和异常记录的数量与本质存在着很大的不同,因此对应的记录分布也存在着很大的不同。正常记录数量较大,出现的频率较高;异常记录数量较少,出现的频率很低。本文算法抓住异常(Outlier)是数据集中与大部分数据有明显差异的数据这一特点,对分类属性以频度为基础对"距离"给出了新的定义,并且依据这一"距离"的定义给出了一个对类的异常度的新定义。

本文算法用于异常检测时分为模型建立和数据评估两个阶段,首先通过训练集得到用于异常检测的模型,然后再用这一模型进行异常检测。即首先通过聚类得到一系列的类,然后再对这些类进行标记,保存这些类的"中心"(对于数值属性,其中心即其均值;对于分类属性,其中心保存是其所有可能的取值的统计频度),数值属性规范化时得到的最大、最小值、标志,以及聚类半径的阈值;检测时计算待检测数据与每个非噪音类间的距离,由这个数据点与其最邻近类的距离以及这一最邻近类的标识来判断这一数据的异常与否。

1.1 连续属性的规范化

由于每一个数值属性的取值范围不同,为了防止具有较大初始值域的数值属性与具有较小初始值域的数值属性相比权重过大的情况发生,我们要对数值属性进行规范化将其转换到一个统一的范围[0,1]。对数值属性作变换:$X' = \frac{X - X_{\min}}{X_{\max} - X_{\min}}$,其中 $X_{\min}$ 为这一属性的最小值,$X_{\max}$ 为这一属性的最大值。并保存每一属性的最大、最小值作为模型的一部分。

1.2 距离的定义

定义1 给定对象 $p = \{p[i] \mid i \in [1,m]\}$,$q = \{p[i] \mid i \in [1,m]\}$,$p,q$ 在属性 i 上的距离 $dif(p[i],q[i])$ 为:

1) 对于分类属性:

$$dif(p[i],q[i]) = \begin{cases} 1 & p[i] \neq q[i] \\ 0 & p[i] = q[i] \end{cases} = 1 - \begin{cases} 0 & p[i] \neq q[i] \\ 1 & p[i] = q[i] \end{cases}$$

2) 对于数值属性:$dif(p[i],q[i]) = | p[i] - q[i] |$;

定义2 给定对象 $p = \{p[i] \mid i \in [1,m]\}$,$q = \{q[i] \mid i \in [1,m]\}$,两个对象 p,q 间的距离 $d(p,q)$ 为:

$$d(p,q) = \left(\sum_{i=1}^{m} dif(p[i],q[i])^x \right)^{1/x} \quad (x > 0)$$

定义3 给定对象 $p = \{p[i] \mid i \in [1,m]\}$,类 C,对象 p 与类 C 间的距离 $d(p,C)$ 为:

$$d(p,C) = \left(\sum_{i=1}^{m} dif(p[i],C[i])^x \right)^{1/x} \quad (x > 0)$$

这里 $dif(p[i],C[i])$ 为 p 与 C 在属性 D_i 上的距离,对于分类属性 D_i 其值为 $dif(p[i],C[i]) = 1 - \frac{Sup_{C|D_i}(p[i])}{| C[i] |}$,$p[i]$ 为 p 在属性 D_i 上的取值,$Sup_{C|D_i}(p[i])$ 表示 C 在属性 D_i 上对 p_i 的支持度。对于数值属性 D_i 其值定义为 $dif(p[i],C[i]) = | p[i] - c[i] |$。

定义4 对给定的类 C_1 与 C_2,C_1 与 C_2 间的距离 $d(C_1,C_2)$ 为:

$$d(C_1,C_2) = \left(\sum_{i=1}^{m} dif(C_1[i],C_2[i])^x \right)^{1/x} \quad (x > 0)$$

这里 $dif(C_1[i],C_2[i])$ 为 C_1 与 C_2 在属性 D_i 上的距离。对于分类属性 D_i,其值为:

$$dif(C_1[i],C_2[i]) = 1 - \frac{1}{| C_1 | \cdot | C_2 |} \sum_{p \in C_1} Sup_{C_1|D_i}(p[i]) \cdot Sup_{C_2|D_i}(p[i]) = 1 - \frac{1}{| C_1 | \cdot | C_2 |} \sum_{q \in C_1} Sup_{C_1|D_i}(q[i]) \cdot Sup_{C_2|D_i}(q[i])$$

对于数值属性 D_i,其值定义为 $dif(C_1[i],C_2[i]) = | c_1[i] - c_2[i] |$。

1.3 聚类算法

算法1 无初始类集的情况下,对混合属性数据进行聚类。

输入:数据对象集 P

输出:类集 $C = \{C_1,C_2,\cdots,C_k\}$

算法描述:

1) 初始时,聚类集合 C 为空;读入一个新的对象 p_1;

2) 以这个对象构造一个新的类 C_1,初始化类个数计数器 $K = 1$;

3) 若已到数据集末尾,则转6),否则读入新对象 p_i,利用给定的距离定义,计算它与类集 C 中每个已有类 C_i($i \in (1,k)$)间的距离 $d(p,C_i)$,并选择最小的距离 $d_{\min}(p,C)$ 以及所对应的类 C_j;

4) 若最小距离 $d_{\min}(p,C)$ 超过给定的阈值 d,则以此对象构造一个新的类 C_{k+1} 将其添加到类集 C 中,将类个数计数器 $K = K + 1$,转3);

5) 否则将该对象并入具有最小距离的类 C_j 中,更新 C_j 的各分类属性值的统计频度及数值属性的质心,转3);

6) 结束。

对算法1做简单修改就可以实现对混合属性数据的增量聚类。

算法2 有初始类集的情况下,对混合属性数据进行增量聚类。

输入:数据对象集 P,类集 $C = \{C_1,C_2,\cdots,C_m\}$

输出:类集 $C' = \{C'_1,C'_2,\cdots,C'_n\}$

算法描述:

1) 初始时,类集为已有的 $C = \{C_1,C_2,\cdots,C_m\}$,初始化类个数计数器 $K = m$;

2) 打开新数据集 P;

3) 若已到数据集末尾,则转6),否则读入新对象 p_i,利用给定的距离定义,计算它与类集 C 中每个已有类 C_i($i \in (1,k)$)间的距离 $d(p,C_i)$,并选择最小的距离 $d_{\min}(p,C)$ 以及所对应的类 C_j;

4) 若最小距离超过给定的阈值 d,则以此对象构造一个

新的类 C_{k+1} 将其添加到类集 C 中,将类个数计数器 $K = K + 1$,转 3);

5) 否则将该对象并入具有最小距离的类 C_j 中,更新 C_j 的各分类属性值的统计频度及数值属性的质心,转 3);

6) 结束。

1.4 聚类半径阈值 d 的选择

聚类半径阈值 d 会影响聚类的质量和算法的执行时间。当 d 减小时,由聚类算法所得到的类的个数将会增加,同时算法所需的执行时间也会随之增加;当 d 增加时,由聚类算法所得到的类的个数将会减少,同时算法所需的执行时间也会随之减少。另一方面如果聚类半径过大则每个类中数据的个数将会增加,使得正常数据与异常数据混杂在一个类中的可能性增加,不利于异常数据的检测。为了使聚类算法能较快地得到适用于异常检测的类集,我们要选择合适的聚类半径。聚类的基本思想是要使得属于同一个类的数据点之间的距离尽可能的小,而属于不同类的数据点之间的距离尽可能的大。于是我们考虑取一个小于所有数据点之间距离的平均值作为聚类半径的阈值,另一方面,如果聚类半径过小则会导致,产生类的个数过多,影响算法的效率。通过在不同数据集上的试验,我们发现当聚类半径在 $[EX - 0.5DX, EX]$ 这一区间时,算法有很好的聚类精度。由于对于一个较大的数据集而言计算所有数据点两两之间距离的平均值的代价太大,我们采用了采样的策略:首先在数据集中随机选择 N_0 个数据点对并根据距离的定义计算它们之间的距离 $d_i, i \in (1, N_0)$ 然后求出这些距离的平均值 EX 以及方差 DX,在 N_0 足够大的情况下 EX 和 DX 将稳定在一个很小的范围内。最后我们根据不同的数据集聚类精度和执行时间的要求在 $[EX - 0.5DX, EX]$ 中选择一个合适的值作为聚类半径的阈值。

1.5 噪音处理及异常类标记

为了得到用于异常检测的模型,我们要对聚类的结果进行噪音处理及异常标记。

定义 5　对类 C,若其中元素个数小于 NoiseCount(预先给定)则 C 为噪音类。

定义 6　对类集 $C = \{C_1, C_2, \cdots, C_k\}$,计算每个类的类异常因子定义为:

$$COF(C_i) = \left(\frac{\sum_{C_j \neq C_i} d(C_i, C_j)^\gamma}{K} \right)^{1/\gamma};$$

首先对所有的非噪音类,计算每个类的 $COF(C_i)$,按 $COF(C_i)$ 降序排列各类,求满足:$\frac{\sum_{i=1}^{b_1} |C_i|}{\sum_{j=1}^{k} |C_j|} \leq \beta$ 的最大 b_1,将类 $C_1, C_2, \cdots, C_{b_1}$ 标识为 outlier 类,而将 $C_{b_1+1}, \cdots, C_k$ 标识为 normal 类。这里 β 为预先给定的参数,为数据集的异常数据比例。

1.6 数据评估

完成对聚类结果的噪声处理及异常标识后我们就得到了可以用来检测的模型,检测算法如下:

算法 3　按照已有模型 M(所有非噪音类的中心及其标识)对数据对象集 P 进行评估。

输入:数据对象集 P,已有检测模型 M 输出:带检测标识数据对象集 P'

算法描述:

1) 读入已有模型 M,打开数据对象集 P;

2) 若已到数据集末尾,则转 5),否则读入新对象 p_i,首先用模型中保存的最大值、最小值对其中的连续属性进行标准化,然后利用给定的距离定义,计算它与已有模型 M 中每个已有类 $C_i (i \in (1,k))$ 间的距离 $d(p, C_i)$,并选择最小的距离 $d_{\min}(p, C)$ 以及所对应的类 C_j;

3) 若最小距离超过给定的聚类半径阈值 d,标记此数据为异常,转 2);

4) 否则以类 C_j 的标识标记此数据对象,转 3);

5) 结束。

1.7 算法时间复杂性分析

假设待处理数据集的大小为 N,由聚类算法所产生的类的个数为 k。在建立模型阶段,聚类算法的时间复杂度为 $O(N \cdot k)$,而标记异常类的算法的时间复杂读为 $O(k^2)$;在检测阶段,对于大小为 N' 的待检测数据集,时间复杂度为 $O(N' \cdot k)$。由于聚类算法所产生的类的个数 k 要远远小于数据集的大小,建立模型和检测数据两个阶段的时间复杂度都是与待处理数据集的大小成近似线性关系的。同时,由于这一算法是增量算法,使得我们可以对已经得到的模型进行更新,这一更新过程时间复杂度也是与待处理数据集的大小成近似线性关系的。由以上分析可以看出本算法具有较好的可扩展性。

2 实验设计及结果分析

为了验证算法的有效性,我们在 KDDCUP99[14] 数据集上对本算法进行了验证。KDDCUP99 数据集包含在 22 种网络环境下的攻击数据,一共有大约 490 000 条记录;每条记录含有 41 种属性和 1 个标志位,其中数值属性 34 种,分类属性 7 种。我们从其中选择出 10% 的子集作为测试集 B,其中包含 38 894 条正常数据和 394 868 条攻击数据。训练集是按照一个给定的正常/攻击比例从 KDDCUP99 数据集中随机选择数据而产生的,我们构造了 2 个训练集 A_1 和 A_2,其中 A_1 包含 38 838 条正常数据和 1 621 攻击数据(攻击数据占 4%),A_2 包含 19 542 条正常数据和 257 条攻击数据(攻击数据占1.3%)。

同时我们还在 Wisconsin Breast Cancer[13] 数据集上对算法进行了验证。Wisconsin Breast Cancer 数据集包含 9 个数值属性和 1 个标志位。C_1 中共有 483 条记录,其中良性 444 条,恶性 39 条(恶性数据占 8.1%);C_2 中共有 261 条记录其中良性 20 条,恶性 241 条(恶性占 92.34%)。

为了检验建模算法对输入数据顺序的相关性,我们对 A_1 中记录的顺序进行了 3 次随机打乱,并在打乱后的数据集上进行训练然后在 B 上进行检测。实验 1、2、3 的结果表明建模算法对输入顺序不敏感。然后在 A_2 上采用不同的聚类半径阈值建立模型然后进行检测,对照实验的结果可以看出随着聚类半径的减小模型的检测率有明显的提高。这是因为 A_2 中异常记录非常少,聚类半径的阈值如果较大就会导致将本来就很少的异常数据混入正常数据中,从而使得检测率很低。同时也可以看到随着聚类半径阈值的减小,聚类个数明显增加,建模时间和检测时间基本同比例增长(线性增长)。接着我们在实验(在 A_2 上聚类)中得到模型的基础上对 A_1 以同样的聚类半径阈值进行增量聚类,对得到新的模型用 B 进行评估。实验

的结果表明通过增量聚类后的新模型比已有的模型的检测能力的到了很大提高。最后用 A_1 建模以 EX 为聚类半径阈值建模在 KDDCUP99 完整数据集上评估得到了很好的结果，可以看出测试时间与测试集大小基本成线性关系，但是由于 KDDCUP99 整个数据集有约 800M，在系统内存不足的情况下磁盘 I/O 的时间消耗较多。另外我们还在 Wisconsin Breast Cancer 数据集上进行了实验，同样取得了很好的结果。

实验结果如表 1 所示。

表 1 实验结果

序号	训练集	异常比例	训练集大小	聚类半径阈值	聚类个数	建模时间(秒)	测试集	测试集大小	检测率	检测时间(秒)
1	$A_1$①	4%	40459	EX = 0.253	15	6	B	433762	98.73%	55
2	$A_1$②	4%	40459	EX = 0.253	16	5	B	433762	98.73%	59
3	$A_1$③	4%	40459	EX = 0.253	16	6	B	433762	98.73%	55
4	A_2	1.3%	19799	EX - 0.25DX = 0.209	24	4	B	433762	98.73%	83
5	A_2	1.3%	19799	EX = 0.242	14	2	B	433762	41.93%	58
6	$A_2 + A_1$	3.22%	60258	EX = 0.243	19	11	B	433762	94.42%	86
7	A_1	4%	40459	EX = 0.238	21	6	ALL	4898430	99.55%	690
8	C_1	8.1%	483	EX - 0.25DX = 0.182	47	0	C_2	261	98.1%	1

说明：聚类半径阈值是通过随机采样计算得到，故对相同的数据集，每次试验也可能有所不同，但总体偏差不大.

表 2 在 KDDCUP99 上的检测率对照

方法	检测率
A_1 上建模(聚类半径取 EX)	98.73%
A_2 上建模(聚类半径取 EX - 0.25DX)	98.73%
文献[15][15]	91.8%

在与其他的方法[15,16]的比较中(见表 2)，本文方法的检测率明显好于对照方法。需要说明的是文献[15]、[16]中的方法是一种有指导的学习方法，训练模型时利用标志位，而本文的方法不需要利用标志位，只需要给出训练集中异常数据的比例就可以了，对先验知识要求很少。另外，关于训练集选择要注意训练集中的异常数据不能过少，过少的异常记录如果再加上比较大的聚类半径阈值，可能导致模型产生较大的偏差；但也不能过多，那有可能使得某一类异常数据的数量过多，使得算法不再把它当作异常数据而作为正常数据，从而产生很多的漏报。一般情况下训练集中的攻击数据不要超过 10% 为宜。

3 结语

在本文中我们提出了一种新的基于高维混合属性数据聚类的异常检测的方法。为了对高维混合属性进行聚类分析，首先给出了在高维混合属性条件下数据之间距离、数据与类之间距离、类之间距离的定义；接着依照每个类之间的关系给出了一种度量类的异常程度的方法。这一算法具有近似线性时间复杂度和很好的可扩展性，适合高维、大数据集的处理，并且对训练集的先验知识要求很低，可以比较容易的得到训练集。在 KDDCUP99 和 Wisconsin Breast Cancer 数据集上的实验结果表明，本文的算法在准确性上优于已知的一些算法。

参考文献：

[1] HAWKINS D. Identification of Outliers[M]. Chapman and Hall, London, 1980.

[2] BARNETT V, LEWIS T. Outliers in statistical data[M]. John Wiley, 1994.

[3] BICKEL DR. Robust estimators of the mode and skewness of continuous data[J]. Computational Statistics and Data Analysis, 2002, 39(2): 153 - 163.

[4] ARNING A, AGRAWAL R, RAGHAVAN P. A Linear Method for Deviation Detection in Large Databases[A]. Proc 2nd Int Conf on Knowledge Discovery and Data Mining[C], Portland, OR, AAAI Press, 1996. 164 - 169.

[5] SARAWAGI S, AGRAWAL R, MEGIDDO N. Discovery-Driven exploration of OLAP data cubes[A]. Proc 6th Int Conf on Extending Database Technology[C]. Valencia: Springer - Verlag, 1998. 168 - 182.

[6] HE ZY, XU XF, DENG SC. Discovering cluster-based local outliers [J]. Pattern Recognition Letters, 2003, 24(9 - 10): 1651 - 1660.

[7] KNORR EM, NG RT. A Unified Approach for Mining Outliers[A]. Proceedings of the 7th CASCON[C], 1997. 236 - 248.

[8] KNORR EM. Outliers and data mining: Finding exceptions in data [D]. Ph D thesis, THE UNIVERSITY OF BRITISH COLUMBIA (CANADA), 2002.

[9] BREUNIG MM, KRIEGEL HP, NG RT, *et al*. LOF: Identifying density-based local outliers[A]. Proceedings of SIGMOD_00[C], Dallas Texas, 2000. 427 - 438.

[10] PAPADIMITRIOU S, KITAGAWA H, GIBBONS PB, *et al*. LOCI: Fast Outlier Detection Using the Local Correlation Integral[R]. Technical Report, IRP-TR-02-09, 2002.

[11] JIANG MF, TSENG SS, SU CM. Two-phase clustering process for outliers detection[J]. Computational Statistics and Data Analysis, 2001, 36(3): 351 - 382.

[12] PORTNOY L, ESKIN E, STOLFO S. Intrusion detection with unlabeled data using clustering[A]. In Proc ACM CSS Workshop on Data Mining Applied to Security (DMSA-2001)[C]. Philadelphia, PA: November, 2001. 5 - 8.

[13] 李翠平，李盛恩，王珊，等. 一种基于约束的多维数据异常点挖掘方法[J]. 软件学报，2003，14(9)：1571 - 1576.

[14] MERZ CJ, MERPHY P. UCI repository of machine learning databases[EB/OL]. http://www.ics.uci.edu/ mlearn/ MLRRepository.html, 2004.

[15] LI XY. Clustering and classification algorithm for computer intrusion detection[M]. Ph D, thesis, Arizona state university, 2001.

[16] ELKAN C. Results of the KDD'99 Classifier Learning Contest[EB/OL]. http://www.cs.ucsd.edu/users/ elkan/clresults.html, 2004.