海量信息异常检测问题的异常概率排序算法

陈刚[1]，蔡远利[1]，穆静[1]，杨卫丽[2]
（1. 西安交通大学电子与信息工程学院，710049，西安；2. 中国国防科技信息中心，100036，北京）

摘要：针对异常检测算法速度慢、精度低、稳定性差等问题，提出了一种通过异常概率排序提取异常点的算法(OAP). 由于异常点相对正常点更容易通过对数据空间的均匀分割而孤立出来，所以OAP通过数据点在均匀 N 叉分割树中的孤立深度估算异常概率的大小，从而得到异常概率的排序，最终构造由 k 个异常概率最大的点组成的列表，列表中的数据就是所求的异常点. OAP不需要距离或密度的计算，复杂度被降到 $O(n)$ 级. 实验结果表明，对于规模线性增加的海量实验数据集，OAP消耗的CPU时间也线性增加；相对iForest算法，其速度提高了30倍，精度提高了20%～30%，且同一数据集上的多次实验结果一致，稳定性高.

关键词：数据挖掘；异常检测；均匀分割；异常概率排序
中图分类号：TP393 **文献标志码**：A **文章编号**：0253-987X(2011)04-0036-05

Ordinal Anomaly Probability Algorithm for Anomaly Detection Problems of Massive Data Sets

CHEN Gang[1], CAI Yuanli[1], MU Jing[1], YANG Weili[2]
(1. School of Electronics and Information Engineering, Xi′an Jiaotong University, Xi′an 710049, China;
2. National Defense Science & Technology Information Center, Beijing 100036, China)

Abstract: An ordinal anomaly probability method (OAP) is proposed to improve the efficiency, effectiveness and stability of existing anomaly detection algorithms. Since anomalies are easier to be isolated by uniformly partitioning the data space, the order of anomaly probabilities can be evaluated in terms of isolation depths in uniform N-ary partition trees. Then, the k largest probable anomalies are extracted. OAP can ignore the evaluation of distance and density, and hence reduces the complexity to $O(n)$. Experimental results show that the CPU time of OAP increases linearly with a linearly growing data set. Furthermore, comparisons show that OAP is 30 times faster than iForest and is much more stable, while its accuracy is improved by 20%-30%.

Keywords: data mining; anomaly detection; uniform partition; ordinal anomaly probability

在工程实际中，人们往往对异常数据更感兴趣. 例如：在网络安全领域，人们关心的是隐藏在正常网络流量中的入侵数据包[1]；在经济金融领域，银行试图通过对帐户交易记录的分析找出那些违规记录[2]；在交通控制中，人们关心的是如何发现和预防交通违章的发生[3]；同样，如何从医学图像中识别出危险病灶，也逐渐成为疾病诊断学研究的重点方向[4]. 纵观人类历史，有许多重要的发现是通过对异常数据的分析得出的. 例如，海王星的发现就是由天王星轨道观测数据异常引出的. 所以，异常检测技术有着重要的价值，是数据挖掘研究的热点方向. 近年来，随着实际应用中数据规模和维度的迅速提高，如何从海量的高维数据中将异常点提取出来，便成了一个重要而且迫切的问题[5].

收稿日期：2010-06-29. 作者简介：陈刚(1984—)，男，博士生；蔡远利(联系人)，男，教授，博士生导师. 基金项目：国家自然科学基金资助项目(60972146，60602025).
网络出版时间：2011-01-17 网络出版地址：http://www.cnki.net/kcms/detail/61.1069.T.20110117.1044.000.html

Breunig 等人[6]提出的本地异常因数(LOF)算法是一种经典的异常检测算法，考虑了相对密度因素，精度较高，但计算过程复杂，运算量随问题规模增加极快，对于海量的高维数据复杂度高达 $O(n^2)$，实用价值不大. 为提高运算速度，Liu 等人[7]提出的 Isolation Forest(iForest)算法采用构造随机森林的方法估算异常度，复杂度被降到 $O(n\log(n))$级，但该算法人为引入了随机因素，带来精度低和稳定性差等问题. Yu 等人[8]提出的二步法是 iForest 和 LOF 的混合体，该算法精度和稳定性有所提高，但是速度有相当程度的下降，并没有很好地解决速度与精度的矛盾.

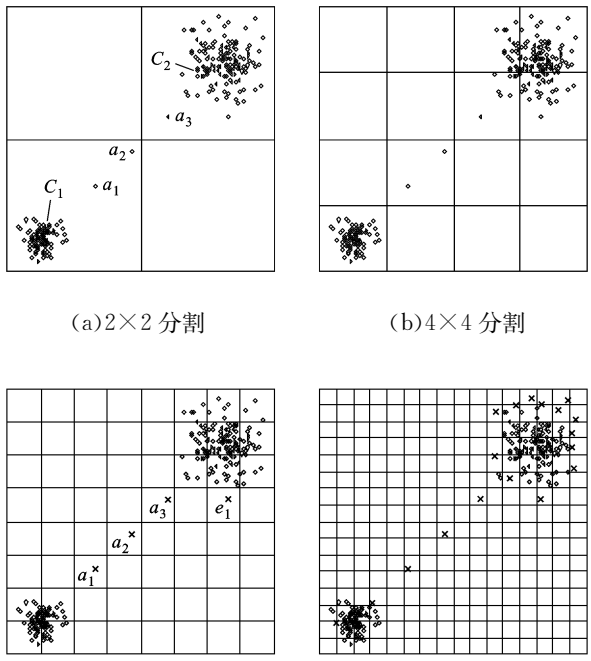
本文通过大量实验，结合异常点少数、独特的本质属性，发现并总结出了一条基本原理：异常点相对正常点来说，更容易通过对数据空间的均匀分割而孤立出来. 由这个简单朴素的基本原理出发，本文发展出一套相对完整的均匀 N 叉分割树理论. 它的贡献在于：剔除了随机因素，使算法获得了极高的稳定性；更合理地利用了异常点的本质属性，精度得到了提高；省去了计算异常度的步骤，运行速度快，复杂度为 $O(n)$级；采用分层的实现方式，内存占用低.

此外，本文创新性引入了何毓琦先生序优化理论的基本思想："顺序"比"数值"更加稳定和鲁棒[9]. 藉此我们突破了传统方法，将关注点从值上转移到了顺序上，即只关注排序，而不关注具体值. 所以，本文所发现的异常概率和孤立深度的反相关关系对于异常概率排序来说是足够的. 对于精确的数值关系是什么样的，本文并不关心. 也正是这个特点，造就了本文算法的高效性、稳定性与鲁棒性，并通过对比实验得到了很好的证明.

1　基本原理

异常点是少数和独特的[5-7]，所以更容易通过对数据空间的均匀分割而独立出来. 下面的例子可以说明这个问题.

在图 1 中，左下角和右上角是 2 个正态分布聚集，记为 C_1、C_2. 中间的 3 个点是异常点，分别记为 a_1、a_2、a_3. 在图 1a～图 1d 中，逐步地对数据空间进行均匀分割. 在图 1c 中可以看到，有 4 个点被分割成了孤立点，它们分别是异常点 a_1、a_2、a_3 以及 C_2 的边缘点 e_1. 在图 1d 中，共有 18 个孤立的点，相对图 1c 中已经发现的 4 个，有 14 个是新发现的，它们都是正态聚集的边缘点. 如果再这样分割下去，新得到的孤立点就只能是 C_1、C_2 的内部点. 由此得到下面的基本原理.



(a)2×2 分割　　(b)4×4 分割

(c)8×8 分割　　(d)16×16 分割

图 1　对数据集的逐步均匀分割

基本原理　越容易通过均匀分割而孤立出来的点，其异常概率越大.

1.1　分割树、N 叉分割树、均匀 N 叉分割树及孤立深度

下面将基本原理中的均匀分割过程数学化，首先定义一般意义下的分割树概念.

定义 1　有数据集 S，其幂集[10]为 2^S. 如果存在一棵树 T，它是结点的有限集，设 T 的任一结点 n 存在 k 个孩子结点[11]，记为 n_i，$1\leqslant i\leqslant k$，且存在映射 M：$T\to 2^S$，如果下面 3 个条件同时成立，则称树 T 为数据集 S 的分割树.

(1) $\forall l,m\in[1,k]$，$l\neq m$，有 $M(n_l)\cap M(n_m)=\varnothing$.

(2) $\bigcup_{i=1}^{k}M(n_i)=M(n)$.

(3) $M(n_0)=S$，其中 n_0 表示 T 的根结点.

定义 1 过于一般，不符合基本原理中均匀分割的模型. 下面的主要工作是对定义 1 进行逐步的特殊化，最终得到均匀 N 叉分割树.

在定义 1 中，如果 k 是一个常数 N，则得到一棵 N 叉分割树.

定义 2　若 T 是数据集 S 的分割树，且 T 的任意结点都有 N 个子树，N 为常数，则称 T 为 S 的 N 叉分割树.

下面定义均匀 N 叉分割树的概念.

定义 3 n 是 N 叉分割树 T 的任意结点,它的孩子结点记为 n_i,对任意 $i\in[1,N]$以及维度 d,如果所有 $s\in M_d(n_i)$有 $s\in[\min(M_d(n)),(\min(M_d(n))+\max(M_d(n)))/2)$ 或 $s\in[(\min(M_d(n))+\max(M_d(n)))/2,\max(M_d(n))]$,其中 M_d 的象是 M 象的第 d 维分量所组成的集合,则称 T 为均匀 N 叉分割树.

定理 1 如果 T 为 S 的均匀 N 叉分割树,且 S 的维度为 d,则有 $N=2^d$.

证明 根据排列组合中的乘法原理,由定义 3 知,对每一维度匀有 2 种可能性,则总的可能的数目为 $N=\underbrace{2\times\cdots\times2}_{\text{总共}d\text{个}}=2^d$.

以图 1 中的均匀分割过程为例,由于是 2 维数据,所以生成的是均匀 4 叉分割树. 图 1a 将数据集分为 4 份,依此类推,最终生成了图 1b~图 1d 中的分割. 下面定义孤立深度的概念.

定义 4 有数据集 S 及其分割树 T,如果 T 中某一结点 n 被 M 映射为 S 中的单一数据点 p,此时称 p 被孤立,且称 n 在 T 中的深度 D 为 p 数据点的孤立深度.

1.2 k-probable 异常列表

定义 5 数据集中异常概率最大的 k 个点按异常概率从大到小排列而成的列表被称为 k-probable 异常列表,简称 k 表.

下面依据定义 5 得到基本原理的重要推论.

推论 1 孤立深度最小的 k 个点,依孤立深度从小到大排列所组成的列表是一个 k 表.

图 2 展示了 k 表的生成过程. 在实现时,可将 k 表的生成过程嵌入到均匀分割的过程中,而且在整个算法运行过程中,仅需存贮最深一层的状态,极大地提高了算法速度而且减小了资源占用.

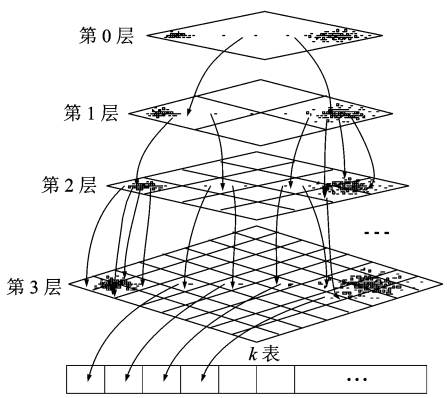


图 2 k 表生成过程

2 算法描述

2.1 均匀 N 叉分割树及 k 表生成算法

总体的均匀分割树算法采用递归的方式表述,如图 3 所示. 其中,数据集 X、当前深度 l、最大深度 $l_{\max}$是输入变量. 在第一次调用时 l 取 0. d_X 表示数据集 X 的维度,N 表示分割树的结点,$M(N)$表示结点的映射集合(参看定义 1),N.child[i]表示结点的第 i 个孩子;addToAnomalyList(x,l)的作用是将 x 点按照深度 l 的顺序插入到 k 表中;split 函数用以产生均匀分割,结果存放在变量 X_{sub}中,它是均匀分割子集的集合,以引用方式调用,$X_{\text{sub}}[i]$表示第 i 个均匀分割子集.

从图 3 可知,算法的层间平均复杂度为 $O(\log(n))$,层内平均复杂度为 $O(n)$,粗略分析,算法的平均复杂度为 $O(n\log(n))$,但是由于定理 1 以及异常点的少数性与独特性,层数 l 达不到 $\log(n)$ 就会结束. 理论及实验证明,层数 l 的实际值很小. 所以,算法的实际平均复杂度为 $O(n)$.

```
算法: UNST_tree
输入: X, l, l_max
输出: UNST_tree, k表
UNST_tree(X, l, l_max) {
        M(N)=X;
        if (l>=l_max) return N;
        if (|X|=1) {
            addToAnomalyList(x, l);
            return N;
        }
        split(X, 0, &X_sub);
        for(i=0;i<2d_X, i++)
            N.child[i] = UNST_tree(X_sub[i], l+1, l_max);
        return N;
}
```

图 3 均匀 N 叉分割树及 k 表生成算法伪代码

2.2 均匀分割算法

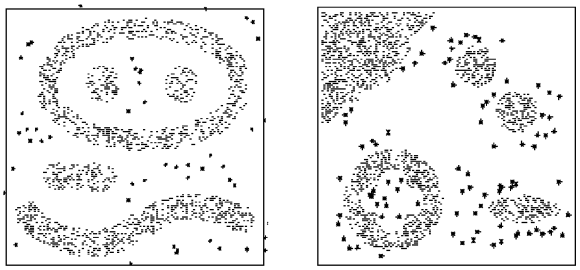
本节实现 split 算法,具体代码见图 4,仍然采用递归的方法表述. 其中数据集 X、当前维度 d、结果集的引用 X_{sub}是输入变量,$X_{\max}[d]$、$X_{\min}[d]$分别表示 X 数据集第 d 维的最大值与最小值,$x[d]$表示点 x 的第 d 维数据;add 操作的作用是将实参加入集合.

3 实验结果及分析

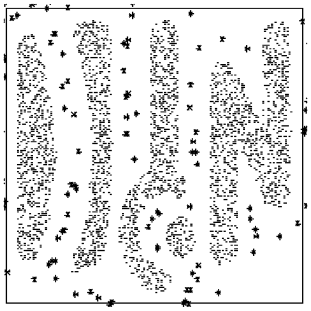
下面通过实验来比较本文算法和 iForest 算法的性能差异. 实验使用 6 个数据集,图 5 给出了其中 3 个的图像.

```
算法：split
输入：X, d, X_sub
输出：分割集 X_sub
split(X, d, & X_sub) {
    if (d > d_X) {
        X_sub.add(X);
        return;
    }
    foreach(x in X) {
        if (X_min[d] < x[d] <= (X_min[d]+X_max[d])/2)
            X_1add(x);
        else
            X_2add(x);
    }
    split(X_1, d+1);
    split(X_2, d+1);
}
```

图 4　均匀分割算法伪代码



(a)SDS1　　(b)SDS3



(c)SDS5

圆点为正常点；星号为异常点

图 5　实验数据集举例

表 1 给出了完整的数据集属性. 实验平台为配有 1.66 GHz CPU 和 1 GB 内存的 Linux 主机. 实验程序采用 C 语言编写，编译器为 GCC-4.4.3.

表 1　数据集属性

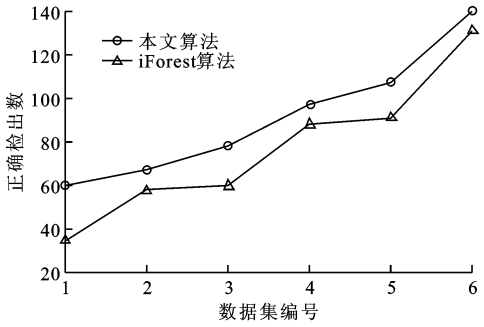
数据集名称	元素数	异常点数	正常聚集数
SDS1	4 060	60	3
SDS2	5 075	75	3
SDS3	6 090	90	5
SDS4	7 105	105	4
SDS5	8 120	120	6
SDS6	10 150	150	4

本文比较 3 个性能参数：①正确检出异常点数(true positive)；②错误点数(mistakes)，其中包括错误检出正常点(false positive)和未检出异常点(false negative)；③运行时间. 实验结果如表 2 所示.

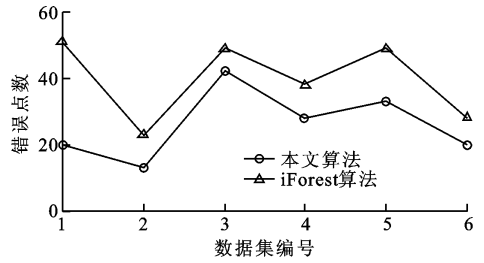
表 2　算法性能比较

数据集	正确检出数		错误点数		运行时间/ms	
	iForest	OAP	iForest	OAP	iForest	OAP
SDS1	35	60	51	20	68	3
SDS2	58	67	23	13	93	3
SDS3	60	78	49	42	113	4
SDS4	88	91	38	28	132	4
SDS5	91	107	49	33	147	5
SDS6	131	140	28	20	191	6

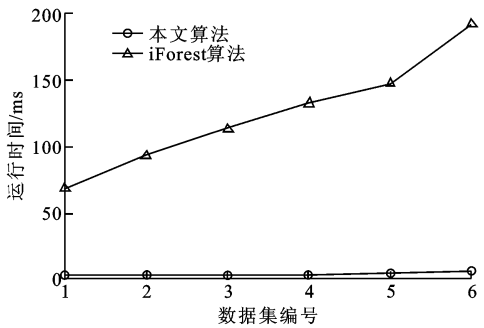
图 6 给出了 2 种算法的性能对比曲线.



(a)正确检出数



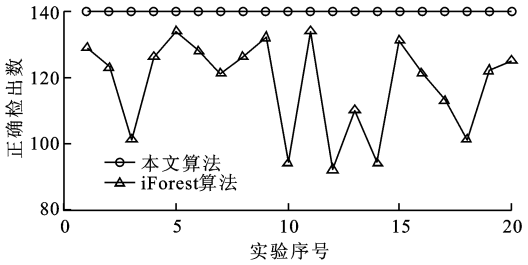
(b)错误点数



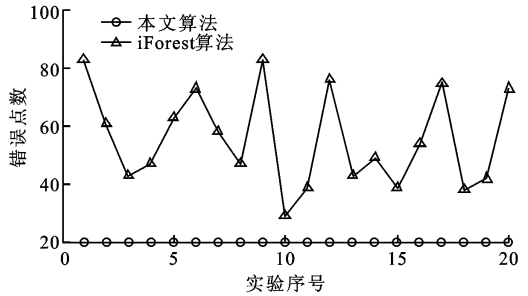
(c)运行时间

图 6　2 种算法的性能对比

为了对比本文算法与 iForest 算法在稳定性上的差异，下面的实验在表 1 中最大的数据集 SDS6 上进行，且连续运行 20 次，得到图 7. 在其他数据集上的运行结果与此类似.



(a)正确检出数稳定性



(b)错误点数稳定性

图 7　2 种算法的稳定性对比

由图 6c 可见，对于规模线性增加的海量实验数据集，OAP 的运行时间也线性增加，而且相对 iForest 算法，速度提高 30 倍. 可见，本文算法不但精度高，而且速度快，且时间复杂度的线性性更好，可扩展性更强. 由图 6a、图 6b 可见，本文算法的精度提高了 20%～30%. 由图 7 可见，本文算法在同一数据集上的多次实验结果是一致的，因此稳定性高. 综上，本文算法在准确性、稳定性和速度上全面超过了 iForest 算法.

4　结束语

本文发现了异常概率和孤立深度的反相关关系. 在它的基础上建立了由均匀 N 叉分割树中的孤立深度估算异常概率排序，从而建立 k-probable 异常列表的方法. 本文方法不依赖于异常度值的估算，而关注于异常概率的排序，因此速度快、稳定性高、鲁棒性好，而且由于更合理地利用了异常点少数、独特的本质属性，所以精度也得到了提高. 此外，本文算法结构简单，非常适应于高维海量数据，也有很好的并行化潜力，具有广阔的应用前景.

参考文献：

[1] ESKIN E, ARNOLD A, PRERAU M, et al. A geometric framework for unsupervised anomaly detection [J]. Advances in Information Security, 2002, 56(4): 21-31.

[2] RICHARD B, DAVID H. Statistical fraud detection: a review [J]. Statistical Science, 2002, 17(3): 235-255.

[3] LI Xiaolie, LI Zhenhui, HAN Jiawei. Temporal outlier detection in vehicle traffic data [C]// Proc 2009 Int Conf on Data Engineering (ICDE'09). Piscataway, NJ, USA: IEEE, 2009: 1319-1322.

[4] MARCEL P, ELIZABETH B, SEAN H, et al. A brain tumor segmentation framework based on outlier detection [J]. Medical Image Analysis, 2004, 8(3): 275-283.

[5] HAN Jiawei, MICHELINE K. Data mining: concepts and techniques [M]. Singapore: Elsevier, 2006: 5-9.

[6] MARKUS M, HANS P, RAYMOND T, et al. LOF: identifying density-based local outliers [C] // Proc ACM SIGMOD'00. New York, USA: ACM, 2000: 93-104.

[7] LIU Feitong, TING Kaiming. Isolation forest [C]// Proceedings of IEEE International Conference on Data Mining ICDM'08. Piscataway, NJ, USA: IEEE, 2008: 413-422.

[8] YÜ Xiao, TANG Lu'an, HAN Jiawei. Filtering and refinement: a two-stage approach for efficient and effective anomaly detection [C]// Proc Int Conf on Data Mining ICDM'09. Piscataway, NJ, USA: IEEE, 2009: 90-104.

[9] HO Y C, SREENIVAS R. Ordinal optimization of DEDS [J]. Discrete Event Dynamic Systems, 1992, 45(9): 61-88.

[10] 祝颂和，陆诗娣. 离散数学 [M]. 西安：西安交通大学出版社，1991：6.

[11] KNUTH D, Art of computer programming [M]. New York, USA: Addison-Wesley, 1998: 51-53.

(编辑　刘杨)