第10章 异常检测

《人工智能算法》

清华大学出版社
2022年7月

提纲

- 引例
- 异常检测概述
- 异常检测算法分类
- 局部异常因子算法（LOF）
- 基于聚类的局部异常因子算法（CBLOF）
- 总结

引例

- 异常数据是指不符合预期行为的数据

ID	0	1	2	3	4	5	6	7	8	9	10
score	80	85	87	82	69	71	0	69	73	77	72

可以直观地从这组一维数据中找出id=6的分数为异常数据，因为其值与其他数据相差很大。

但是，现实生活中的数据可能很多并且不止一维，此时我们就需要更有效算法来找出异常数据。

提纲

异常检测概述 (1)

- **异常检测（Anomaly Detection）**——检测数据中不符合预期行为的数据，其基本思想通过数据挖掘方法找出显著不同于其他数据的异常点，并发现潜在的、有意义的知识
- **异常检测的应用**

欺诈识别

数据清理

网络入侵检测

故障检测

异常检测概述 (2)

异常点类型

- 单点异常——某个点与全局大多数点都不一样，该点构成了单点异常
- 上下文异常——一个点只有在特定的上下文下才叫做异常，如果没有这个上下文，该点就是正常的

> 冬天这里的气温是35℃，不看冬天这个上下文，35℃是正常的，但是加上冬天这个条件，35℃就是异常的

- 集体异常——由多个对象组合构成，即单独看某个个体可能并不存在异常，但这些个体同时出现，则构成了一种异常

提纲

- 引例
- 异常检测概述
- 异常检测算法分类
- 局部异常因子算法（LOF）
- 基于聚类的局部异常因子算法（CBLOF）
- 总结

异常检测算法分类 (1)

异常检测算法
- 基于统计学的算法
- 基于距离的算法
- 基于聚类的算法
- 基于密度的算法
- ……

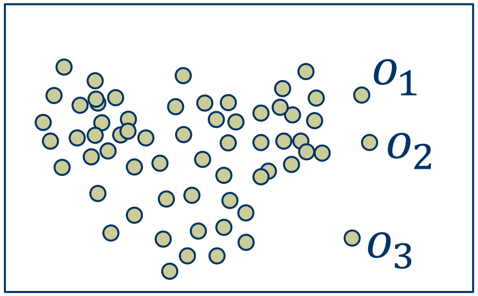
- **基于统计（Statistics-based）的方法**

✓ 正常的数据是遵循特定分布形式的，并且占了很大比例，而异常点的位置和正常点相比存在比较大的偏移。

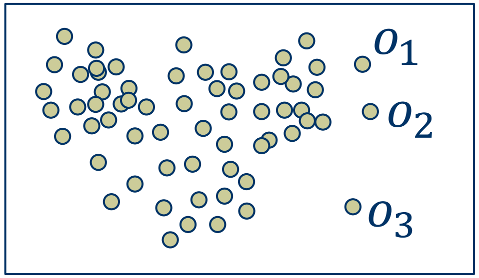
✓ 该方法需假定大部分数据服从一定的分布，而这样的分布在现实中往往很难获取，从而限制了该类算法的发展和应用。

异常检测算法分类 (2)

- **基于距离（Distance-based）的方法**
  - ✓ 将每个数据当作一个点，通过计算每个点与周围点的距离来判断一个点是否为异常点
  - ✓ o_3与其周围点的距离均较远，相比其它点较为异常



- **基于密度（Density-based）的方法**
  - ✓ 将每个数据当作一个点，计算它的周围密度和其临近点的周围密度，基于这两个密度值计算出相对密度，相对密度越大，异常程度越高。
  - ✓ o_1, o_2, o_3的相对密度相比于其相邻对象明显较小，较大可能成为异常点。

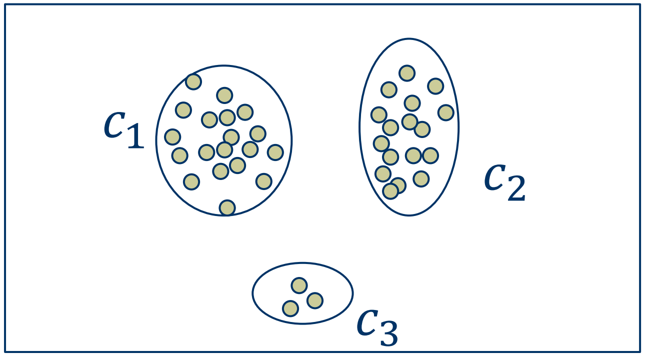


异常检测算法分类 (3)

- **基于聚类（Cluster-based）的方法**

  将数据样本划分为不同的簇，选择小簇中的样本作为候选异常点，

  以非候选点构成的簇和候选点之间的距离作为是否存在异常的依据



c_3为小簇，其内的数据可纳入候选异常点， c_1, c_2为大簇，其内的数据可作为候选正常点

提纲

- 引例
- 异常检测概述
- 异常检测算法分类
- 局部异常因子算法（LOF）
- 基于聚类的局部异常因子算法（CBLOF）
- 总结

局部异常因子算法 (1)

- **LOF的基本思想**

通过比较每个样本和其邻域样本的密度来判断该样本是否为异常。样本的密度越低，越有可能是异常点。

LOF算法中样本的密度通过样本的k距离邻域计算得到，而不是通过全局计算得到，这里的“k距离邻域”即为该算法中“局部”的概念。

局部异常因子算法 (2)

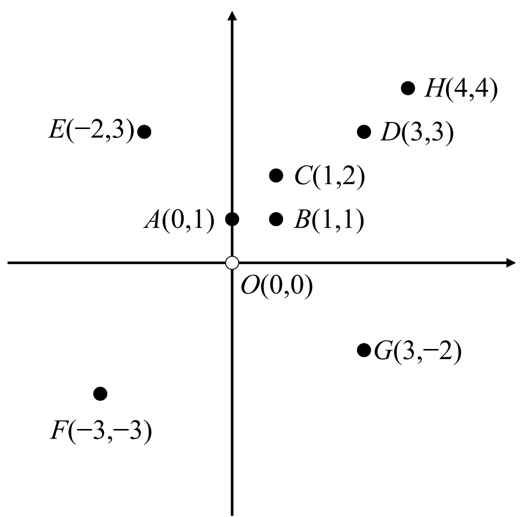
◆ **LOF的相关定义**

（1）k距离（k-distance）

样本O和距离O的第k近的样本之间的距离，k为事先设定的阈值：

$$D_k\,(x_i) = \| x_i - x_{i,k} \| = \sqrt{(x_i^1 - x_{i,k}^1)^2 + (x_i^2 - x_{i,k}^2)^2 + \cdots + (x_i^t - x_{i,k}^t)^2}$$

其中，$x_{i,k}$表示距x_i第k近的样本，x_i表示第i个样本，$\|x_i - x_{i,k}\|$表示两样本间的距离，t表示样本维度。



k=1时，$D_1\,(O) = 1$

k=2时，$D_2\,(O) = \sqrt{2}$

k=3时，$D_3\,(O) = \sqrt{5}$

k=4时，$D_4\,(O) = \sqrt{13}$

k=5时，$D_5\,(O) = 3\sqrt{2}$

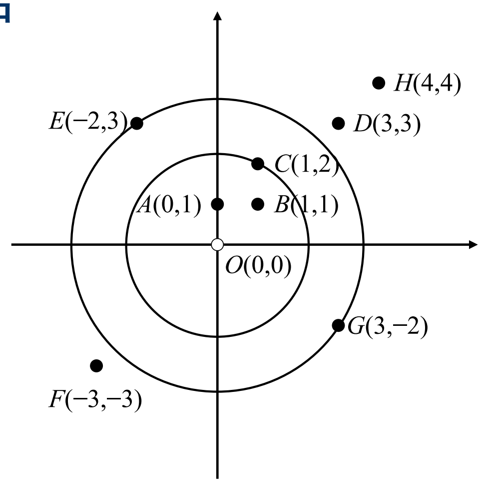
k=6时，$D_6\,(O) = 4\sqrt{2}$

局部异常因子算法 (3)

（2）k距离领域

（k-distance Neighborhood）

到样本O的距离小于O的k距离的所有样本构成的集合



k=3时，O的k距离邻域为$\{A, B, C\}$

k=4时，O的k距离邻域为$\{A, B, C, E, G\}$

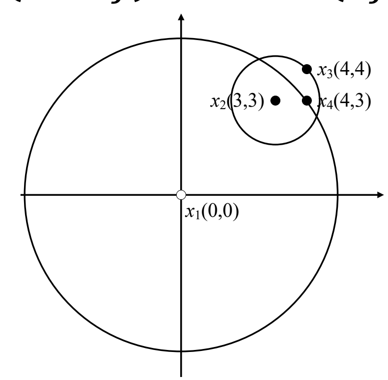
（3）可达距离

（Reachability Distance）

$$RD_k(x_i, x_j) = \max\{D_k(x_i), \| x_i - x_j \|\}$$

当x_i到x_j的距离比x_i的k距离大时，x_i到x_j的可达距离为$\|x_i-x_j\|$，否则为x_i的k距离。

$$RD_k(x_i, x_j) \neq RD_k(x_j, x_i)$$



局部异常因子算法 (4)

（4）局部可达密度（Local Reachability Density）

样本O的k距离邻域内的样本到O的平均可达距离的倒数：

$$LRD_k(x_i) = \left(\frac{1}{N}\sum_{j=1}^{N} RD_k\left(x_j^N, x_i\right)\right)^{-1}$$

用x^N表示k距离邻域中有N个样本，x_j^N表示x^N中的第j个样本。

（5）局部异常因子（ Local Reachability Density）

O的k距离邻域中所有样本的局部可达密度的均值与O的局部可达密度之比：

$$LOF_k(x_i) = \frac{\frac{1}{N}\sum_{j=1}^{N} LRD_k\left(x_j^N\right)}{LRD_k(x_i)}$$

若$LOF_k\ (x_i)$>阈值，则数据异常，否则正常。

局部异常因子算法 (5)

- **算法步骤**

输入：数据样本集 $X=\{x_1, x_2, \cdots, x_n\}$, $x_i=\{x_i^1, x_i^2, \cdots, x_i^t\}(1\le i\le n)$

（1）设定邻域值k和阈值ε

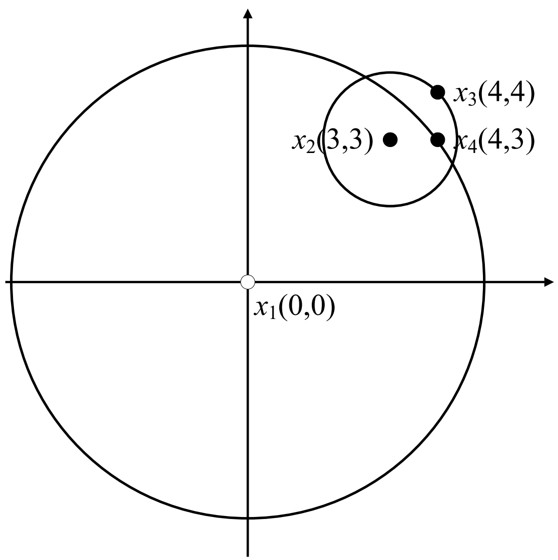
（2）计算k距离和k距离邻域

（3）计算局部可达密度

（4）计算局部异常因子

（5）比较局部异常因子与ε的大小

局部异常因子算法 (6)



设置k=2，
阈值ε=1.5

计算样本点间距离

	x_1	x_2	x_3	x_4
x_1	0	4.24	5.66	5
x_2	4.24	0	1.41	1
x_3	5.66	1.41	0	1
x_4	5	1	1	0

计算样本点间可达距离

	x_1	x_2	x_3	x_4
x_1	0	5	5.66	5
x_2	4.24	0	1.41	1.41
x_3	5.66	1.41	0	1.41
x_4	5	1	1	0

计算k距离

$D_2(x_1)$	$D_2(x_2)$	$D_2(x_3)$	$D_2(x_4)$
5	1.41	1.41	1

局部异常因子算法 (7)

计算局部可达密度

$LRD_2(x_1)$	$LRD_2(x_2)$	$LRD_2(x_3)$	$LRD_2(x_4)$
0.216	0.828	0.828	0.707

计算局部异常因子

$LOF_2(x_1)$	$LOF_2(x_2)$	$LOF_2(x_3)$	$LOF_2(x_4)$
3.549	0.926	0.926	1.172

与阈值（1.5）进行比较

x_1	x_2	x_3	x_4
异常	正常	正常	正常

局部异常因子算法 (8)

- **算法** **LOF** $(X\{x_1, x_2, \cdots, x_n\}, k, \varepsilon)$

构建$n\times n$的样本间矩阵D //$D_{(i,j)}$表示第i行第j列，D_i表示第i行
For i=1 To n Do
 For j=1 To n Do
  $D_{(i,j)} \leftarrow$计算样本x_i与x_j之间的距离
 End For
 $Kd_i \leftarrow$根据D_i计算样本x_i的k距离
 $KN_i \leftarrow$根据KD_i与D_i生成样本x_i的k距离邻域样本的索引
End For
For i=1 To n Do
 For each j in KN_i Do
  $D_{(i,j)} \leftarrow \max\{KD_i, D_{(i,j)}\}$
 End For
End For

局部异常因子算法 (9)

For i=1 To n Do

LRD$_k(i)$← $\left(\sum_{j\in KN_i} D_{(j,i)}/|KN_i|\right)^{-1}$ //计算局部可达密度

End For

For i=1 To n Do

LOF$_k(i)$← $\frac{1}{|KN_i|}\left(\sum_{j\in KN_i} LRD_k(j)\right)/LRD_k(i)$ //计算局部异常因子

End For

For i=1 To n Do

If LOF$_k(i)$ > ε Then

$N \leftarrow N \cup \{i\}$ // x_i是异常样本

Else

$M \leftarrow M \cup \{i\}$ // x_i是正常样本

End If

End For

Return LOF$_k$, M, N

//输出：LOF$_\mathrm{k}$：局部异常因子矩阵；M：正常点索引矩阵；N：异常点索引矩阵

时间复杂度: $O(n^2)$

空间复杂度: $O(n^2)$

局部异常因子算法 (10)

- **优点**

算法简单直观，不需知道数据集分布，并能量化每个样本的异常程度。

- **缺点**

算法时间复杂度为$O(n^2)$，当数据数量和维度很大时，计算量也会变得很大；将样本不同维度属性之间的差别等同看待，有时并不符合实际需求，会带来量纲和计算量的问题；且算法的表现很依赖于k值和阈值的选择。

提纲

- 引例
- 异常检测概述
- 异常检测算法分类
- 局部异常因子算法（LOF）
- 基于聚类的局部异常因子算法（CBLOF）
- 总结

基于聚类的局部异常因子算法 (1)

- **基本概念**

  聚类：将数据分为多个簇，尽可能使簇内相似度大、簇间相似度小

  异常检测：检测数据中不符合预期行为的异常数据

  联系：常见的聚类算法扩展后都能应用于异常检测

- **基本思想**

  (1) 对数据样本进行聚类得到簇集合

  (2) 由每个簇中样本的数量将簇分为大簇和小簇

  (3) 计算异常得分，也就是一个样本到最近的大簇中心的距离

  (4) 按各样本的异常得分判断该样本是否属于异常点

基于聚类的局部异常因子算法 (2)

算法步骤

数据样本集$X=\{x_1,x_2,\cdots,x_i,\cdots,x_n\},x_i=\{x_i^1,x_i^2,\cdots,x_i^t\}$

- Step1. 聚类

  将X聚类为m个簇，$C=\{c_1,c_2,\cdots,c_m\},|c_m|$为第$m$个簇中的样本数量

- Step2. 划分簇

  将m个簇按样本数量进行排序，假设$|c_1|\geq|c_2|\geq\cdots\geq|c_m|$

  ✓ 绝对多数：$(|c_1|+|c_2|+\cdots+|c_b|)\geq|X|\times\alpha,\alpha\in\{0.5\sim1\}$ 默认为0.9，$LC=\{c_i,i\leq b\}$为大簇集合

  ✓ 突降：$|c_b|/|c_{b+1}|\geq\beta$（默认为5，突降5倍），$SC=\{c_j,j>b\}$为小簇集合

基于聚类的局部异常因子算法 (3)

- Step3. 计算数据样本的异常得分（ x_l样本为例）：

✓若x_l是大簇里面的样本，直接计算它到簇中心的距离即可

✓若x_l是小簇里面的样本，分别计算其到所有大簇的距离，并选取最小的值

$$CBLOF(x_l) = \begin{cases} |c_i| \times \min\left(distance(x_l, c_j)\right) & x_l \in c_i, c_i \in SC, c_j \in LC \\ |c_i| \times distance(x_l, c_i) & x_l \in c_i, c_i \in LC \end{cases}$$

- Step4. 选取异常样本：

异常得分较大的样本即判定为异常数据

基于聚类的局部异常因子算法 (4)

$X =$

ID	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
sales	61	36	3083	12	84	7	544	30	146	277	409	30	8159	2	55	14	1188
profit	11	6	-1665	2	14	2	91	6	34	45	68	5	-1359	-3	10	2	-950

示例: X共包含17个样本，每个样本有2个属性，以k-mean聚类算法为基础演示CBLOF算法（簇个数$k = 3, \alpha = 0.8, \beta = 5$）

簇号	ID
0	1, 2, 4, 5, 6, 7, 8, 9, 10, 11, 12, 14, 15, 16
1	3, 13
2	17

绝对多数

→

突降

大簇	小簇
0	1, 2

基于聚类的局部异常因子算法 (5)

ID	1	2	3	4	5	6	7	8	
异常得分	0.102	0.158	18.591	0.206	0.077	0.212	0.768	0.170	
ID	9	10	11	12	13	14	15	16	17
异常得分	0.147	0.267	0.522	0.174	15.257	0.276	0.123	0.206	10.711

- 异常样本的比例（一般为1%，此例样本较少，可调节为15%）

分类	ID
正常样本	1, 2, 4, 5, 6, 7, 8, 9, 10, 11, 12, 14, 15, 16
异常样本	3, 13, 17

基于聚类的局部异常因子算法 (6)

- **算法** **AD_CBLOF**$(\boldsymbol{X}\{\boldsymbol{x_1},\boldsymbol{x_2},\cdots,\boldsymbol{x_n}\},\boldsymbol{\alpha},\boldsymbol{\beta})$

$C\leftarrow\{k-means(X,n\_clusters=m)\}$
$C\leftarrow\{c_1,c_2,\cdots,c_{m-1},c_m||c_m|\le|c_{m-1}|\le\cdots\le|c_2|\le|c_1|\}$
for $b\leftarrow1$ to m do
  $\mathrm{LC}=\{c_i,i\le b\},(|c_1|+|c_2|+\cdots+|c_b|)\ge|X|\times\alpha$
  $\mathrm{SC}=\{c_j,j>b\},|c_b|/|c_{b+1}|\ge\beta$
end for
$U\leftarrow\emptyset$
for $l\leftarrow1$ to n do
  if $x_l\in c_i$ and $c_i\in\mathrm{SC}$ then
    $CBLOF(x_l)\leftarrow|c_i|\times min\{distance(x_l,c_j)\}$ $//c_j\in\mathrm{LC}$
    $U\leftarrow U\cup\{CBLOF(x_l)\}$
  else
    $CBLOF(x_l)|c_i|\times distance(x_l,c_i)$ $//c_i\in\mathrm{LC}$
    $U\leftarrow U\cup\{CBLOF(x_l)\}$
  end if
end for
return U

时间复杂度: $O(n)$
空间复杂度: $O(n)$

基于聚类的局部异常因子算法 (7)

◆ **优点**

不需要监督，易适应在线或增量模式，适用于时空数据的异常检测。若选择聚类算法的时间和空间复杂度是线性的或接近线性的，基于这类算法的异常检测技术对大规模数据集也是有效的。

◆ **缺点**

没有任何一种聚类算法适用于所有数据集，不同数据集需要采用不同的聚类算法。当聚类算法的选取不合适时，样本不能创建任何有意义的簇，那么该方法可能会失败。针对高维空间中的稀疏数据，任意两个样本间的距离可能会非常相似，聚类算法可能不会得到有意义的簇。

提纲

- 引例
- 异常检测的概述
- 异常检测的分类
- 局部异常因子算法（LOF）
- 基于聚类的局部异常因子算法（CBLOF）
- 总结

总结

- 异常检测的基本思想、分类方法，所解决的主要问题
- 异常检测算法解决问题的一般方法和步骤、及其优缺点
- 异常检测的重要算法实例：

  - 基于密度的LOF算法

  - 基于聚类的CBLOF算法

结语

谢谢！