Beijing Forest Studio

北京理工大学信息系统及安全对抗实验中心

异常检测算法

孤立森林（Isolation Forest）

刘宇　硕士

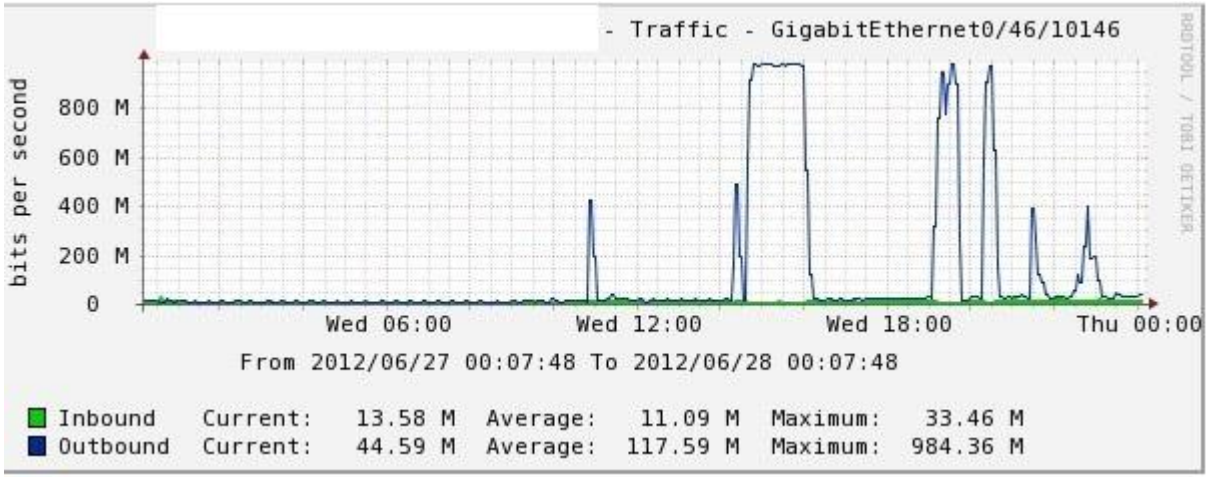
2017年11月26日

内容提要

- 背景简介
- 基本概念
- 算法原理
- 优劣分析
- 应用总结
- 参考文献

背景简介

- 异常
  - 异常是与正常情况具有不同数据特征的数据模式
  - 异常通常在各种应用领域提供关键和可操作的信息



要求：
准确度高
执行速度快

背景简介

- 现存的异常检测方法
  - 基于分类/聚类的方法
    - RNN、one-class SVM、RF
    - 构建正常实例的模型，将不符合模型的实例识别为异常
  - 基于密度/距离的方法
    - ORCA、LOF

- 现存方法的主要缺陷
  - 没有针对检测异常进行优化，误报率和漏报率较大
  - 由于其原有算法的传统，许多现有的方法只适用于低维数据和小数据量

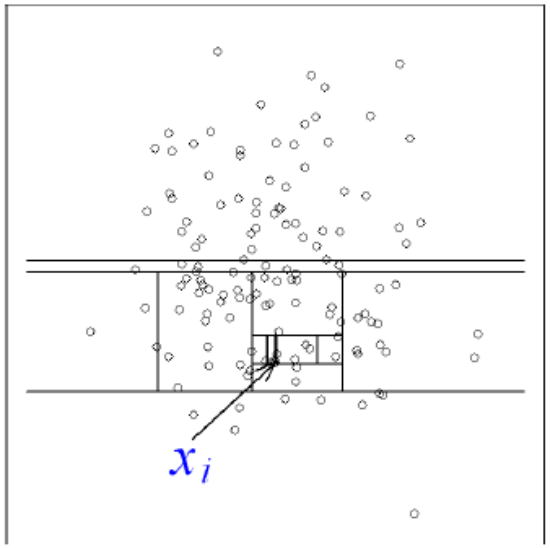
背景简介

- 孤立森林IForest
  - 不依赖任何距离或密度的方法
  - 利用异常的两种定量属性——少而不同
    - the minority consisting of few instances
    - attribute-values that are very different from those of normal instances
  - 采用二叉树的数据结构
  - 异常实例同正常实例相比更靠近根节点
  - 有效和高效的异常检测算法

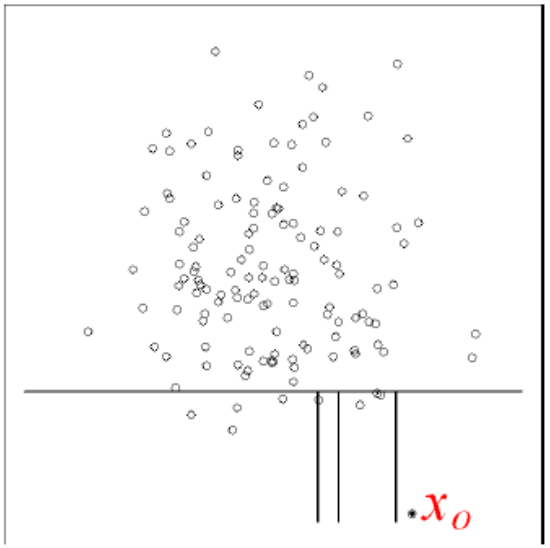
基本概念

基本概念

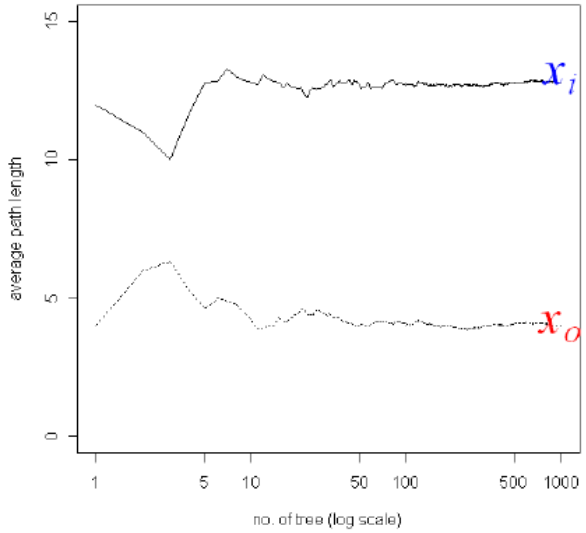
- Isolation
  - separating an instance from the rest of the instances



(a) Isolating x_i



(b) Isolating x_o



(c) Average path lengths converge

基本概念

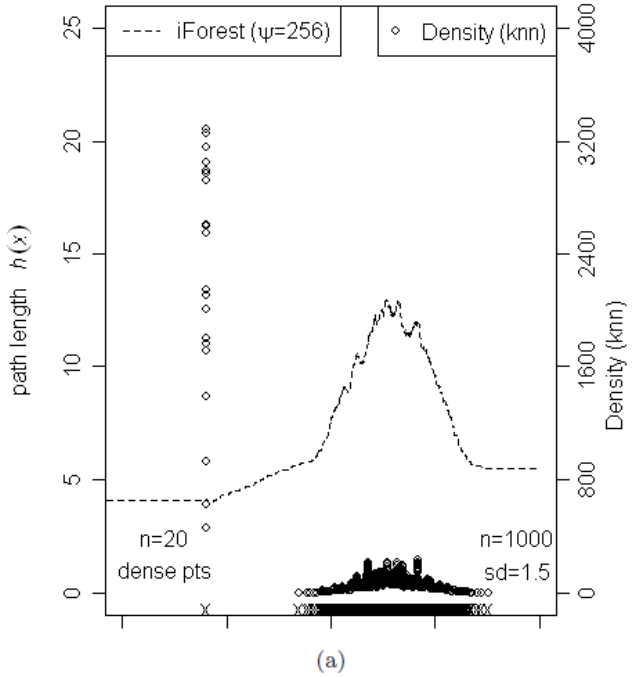
- Isolation Tree
  - iForest 由t个iTree孤立树组成
  - 每个iTree是一个二叉树结构

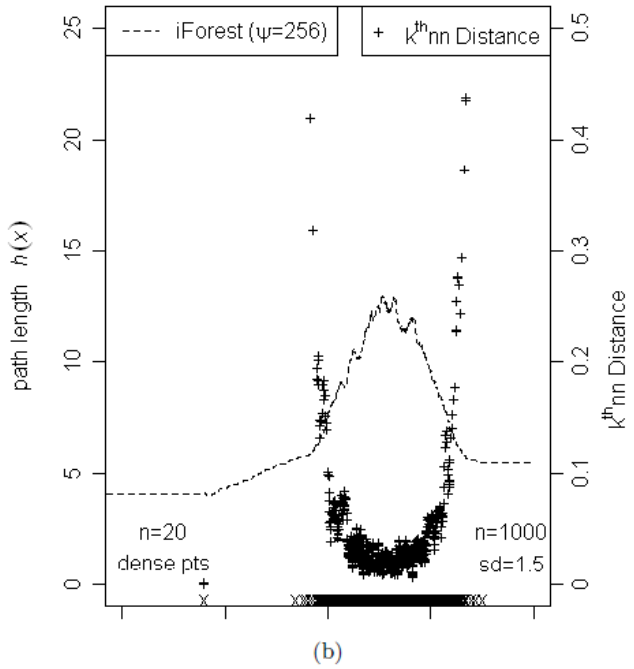
iTree	BST
Proper binary trees	Proper binary trees
External node termination	Unsuccessful search
Not applicable	Successful search

- Path Length
  - 从根节点开始，到外部节点终止，所经过的边的数量
    - 短路径意味着高度易孤立
    - 长路径意味着低度难孤立

基本概念

- 孤立、密度、距离
  - 基于密度的方法
    - 基本假设：正常点出现在密集区，异常点出现在稀疏区
  - 基于距离的方法
    - 基本假设：正常点与相邻点距离近，异常点与相邻点距离远

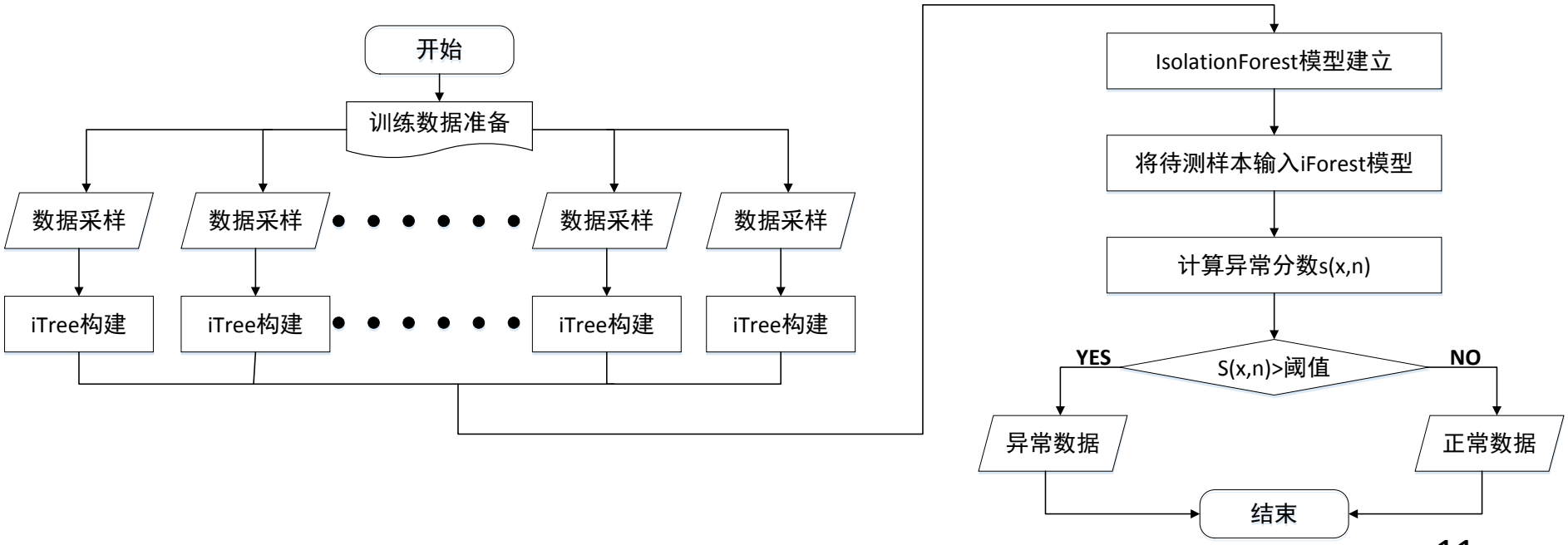




算法原理

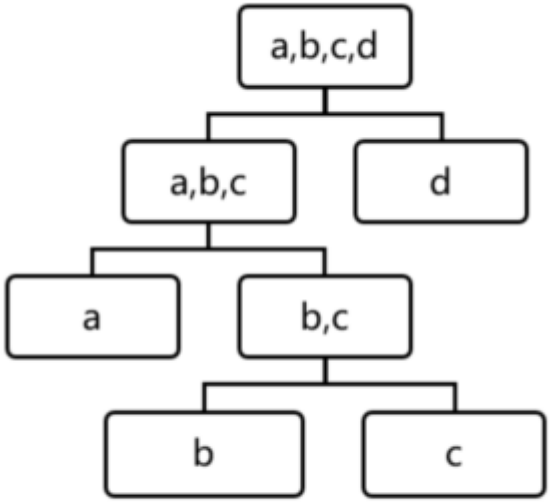
算法原理

- IForest算法流程图
  - 训练阶段
    - 获得t棵iTree，构建IsolationForest模型
  - 评估阶段
    - 用生成的iForest来评估测试数据



算法原理

- 训练阶段
  - Isolation Tree
    - 节点T
      - 没有子节点的外部节点
      - 有两个子节点的内部节点（T_l，T_r）
    - 属性值q和分裂值p
    - 完满二叉树



b和c的高度为3，a的高度是2，d的高度是1。

算法原理

- 算法实现
  - 使用给定训练集的子采样构建孤立树
  - 参数设置
    - 子采样大小 ψ
    - 树的数量 t

Algorithm 1 : $iForest(X, t, \psi)$

Inputs: X - input data, t - number of trees, ψ - subsampling size
Output: a set of t *iTrees*

```
Initialize Forest
for i = 1 to t do
    X' ← sample(X, ψ)
    Forest ← Forest ∪ iTree(X')
end for
return Forest
```

算法原理

Algorithm 2 : $iTree(X')$

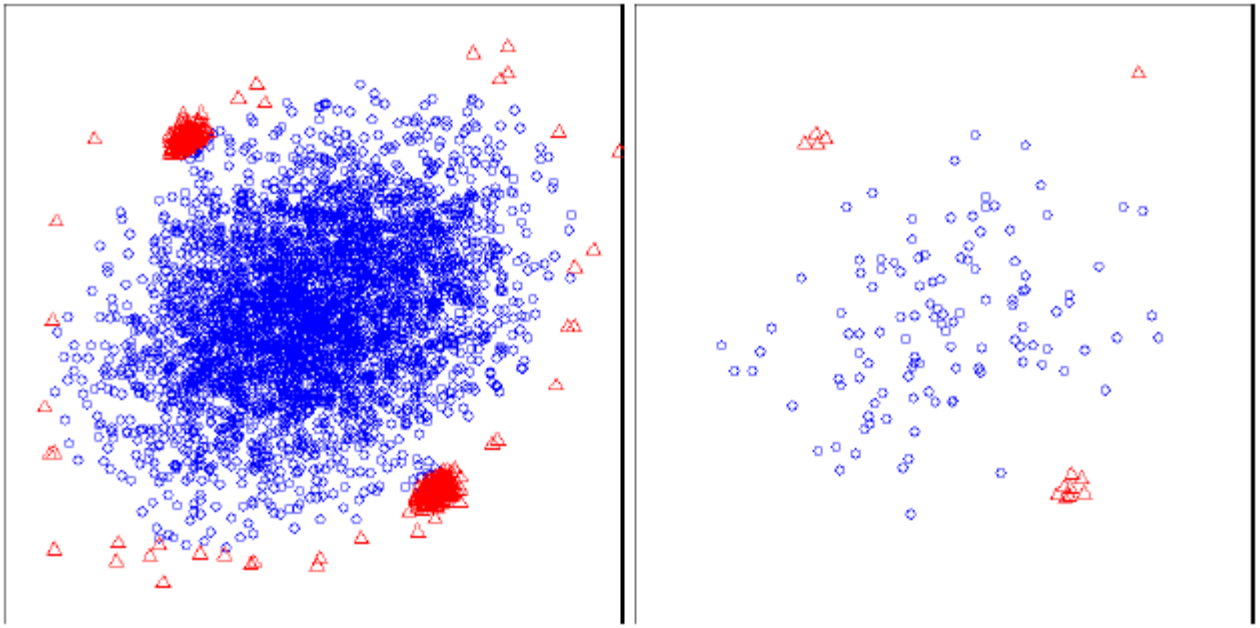
Inputs: X' - input data
Output: an iTree

1: **if** X' cannot be divided **then**
2: return $exNode\{Size \leftarrow |X'|\}$
3: **else**
4: let Q be a list of attributes in X'
5: randomly select an attribute $q \in Q$
6: randomly select a split point p between the max and min values of attribute q in X'
7: $X_l \leftarrow filter(X', q < p)$
8: $X_r \leftarrow filter(X', q \geq p)$
9: return $inNode\{Left \leftarrow iTree(X_l),$
10: $Right \leftarrow iTree(X_r),$
11: $SplitAtt \leftarrow q,$
12: $SplitValue \leftarrow p\}$
13: **end if**

- 终止条件
  - 树高达到高度限制
  - |X|=1 或 X中的所有数据具有相同的数值

算法原理

- 误报与漏报的解决
  - 二次采样，子采样减少了对异常隔离的干扰



(a) Original sample
(4096 instances)

(b) Sub-sample
(128 instances)

算法原理

- 评估阶段
  - 单条路径长度可以通过从根节点到终止节点的边的数量计算
  - 当前路径长度超过预定义的高度限制时，需要改进h(x)的计算
  - 最坏情况下，时间复杂度为$O(nt\psi)$

Algorithm 3 : $PathLength(x, T, hlim, e)$

Inputs : x - an instance, T - an iTree, $hlim$ - height limit, e - current path length; to be initialized to zero when first called

Output: path length of x

```
1: if T is an external node or e ≥ hlim then
2:     return e + c(T.size) {c(.) is defined in Equation 1}
3: end if
4: a ← T.splitAtt
5: if x_a < T.splitValue then
6:     return PathLength(x, T.left, hlim, e + 1)
7: else {x_a ≥ T.splitValue}
8:     return PathLength(x, T.right, hlim, e + 1)
9: end if
```

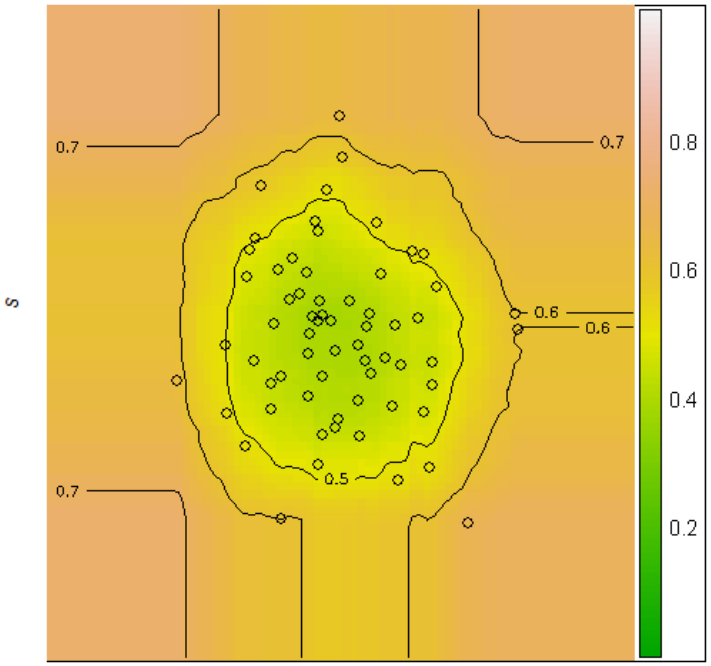
算法原理

- 异常分数
  - 借助二叉搜索树BST

$$c(\psi)=\begin{cases}2H(\psi-1)-2(\psi-1)/n & \psi>2\\ 1 & \psi=2\\ 0 & otherwise\end{cases}$$

$$H(i)=\ln(i)+0.5772156649$$

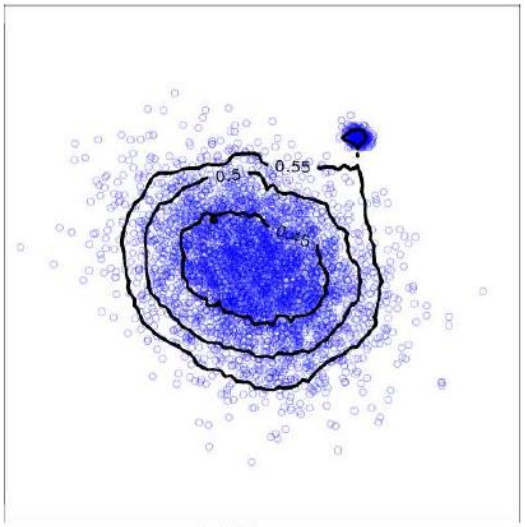
$$s(x,\psi)=2^{-\frac{E(h(x))}{c(\psi)}}$$



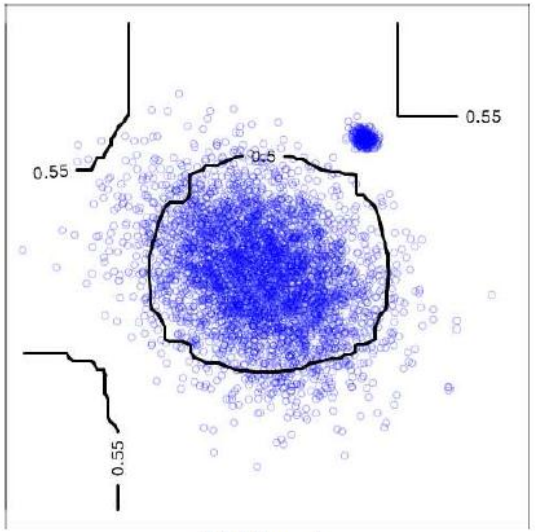
  - 异常分数的取值
    - S接近于1，更可能为异常
    - S小于0.5，更可能为正常
    - S接近0.5，没有明显的异常

算法原理

- 异常分数粒度的调整
  - 调整评估阶段的树高限制参数实现
  - 解决异常集群问题



(a) $hlim = 6$,
points surrounding both clusters are treated as anomalies.
Note that there is a contour line of 0.55 inside the small dense cluster.



(b) $hlim = 1$,
the small isolated cluster and the surrounding points of the large cluster are treated as anomalies.

优劣分析

优劣分析

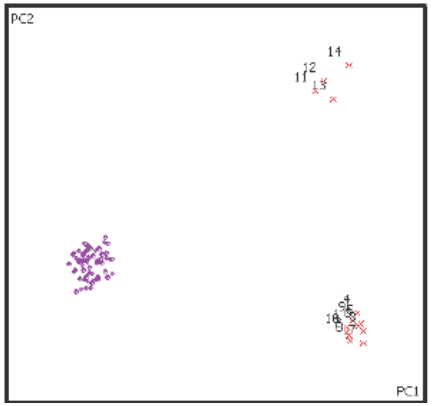
- ORCA
  - 基于距离的异常检测算法
- SVM
  - 基于分类模型的异常检测算法
- LOF（Local Outlier Factor）
  - 基于密度的经典异常检测算法
- RF（Random Forest）
  - 基于分类模型的异常检测算法

	n	d	anomaly class
Http	567497	3	attack(0.4%)
ForestCover	286048	10	class 4 (0.9%) vs.class 2
Mulcross	262144	4	2 clusters(10%)
Smtp	95156	3	attack(0.03%)
Shuttle	49097	9	classes 2,3,5,6,7(7%)
Mammography	11183	6	class 1(2%)
Annthyroid	6832	6	classes 1,2(7%)
Satellite	6435	36	3 smallest classes(32%)
Pima	768	8	pos(35%)
Breastw	683	9	malignant(35%)
Arrhythmia	452	274	classes 03,04,05,07,08,09,14, 15(15%)
Ionosphere	351	32	bad(36%)
hbk	75	4	14 points(36%)
wood	20	6	6 instances(30%)

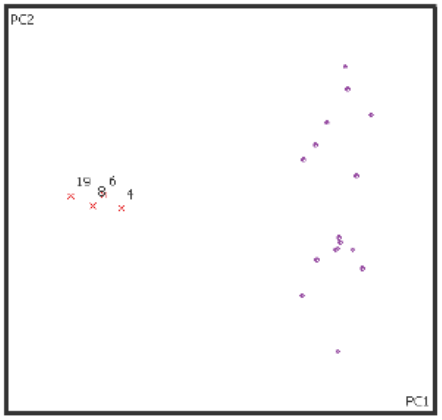
优劣分析

- 统计数据集Hbk和Wood

rank#	iForest	Orca	SVM	LOF	RF
1	**14**	**14**	**1**	**14**	**12**
2	**12**	**12**	**3**	47	**14**
3	**13**	**11**	**4**	68	**13**
4	**11**	**13**	**7**	16	**11**
5	**10**	**7**	**10**	61	**4**
6	**4**	47	**11**	34	**9**
7	**7**	**4**	**12**	53	**1**
8	**1**	**3**	**13**	**12**	**10**
9	**2**	**1**	**14**	52	**7**
10	**8**	**10**	16	42	**6**
11	**9**	**6**	43	37	**3**
12	**3**	70	44	75	**8**
13	**5**	53	47	44	**2**
14	**6**	25	49	36	**5**
15	47	68	60	31	43
16	52	43	68	70	22
17	68	**8**	**2**	62	61
18	53	**2**	**5**	30	20
19	43	62	**6**	35	38
20	60	**5**	**8**	43	74

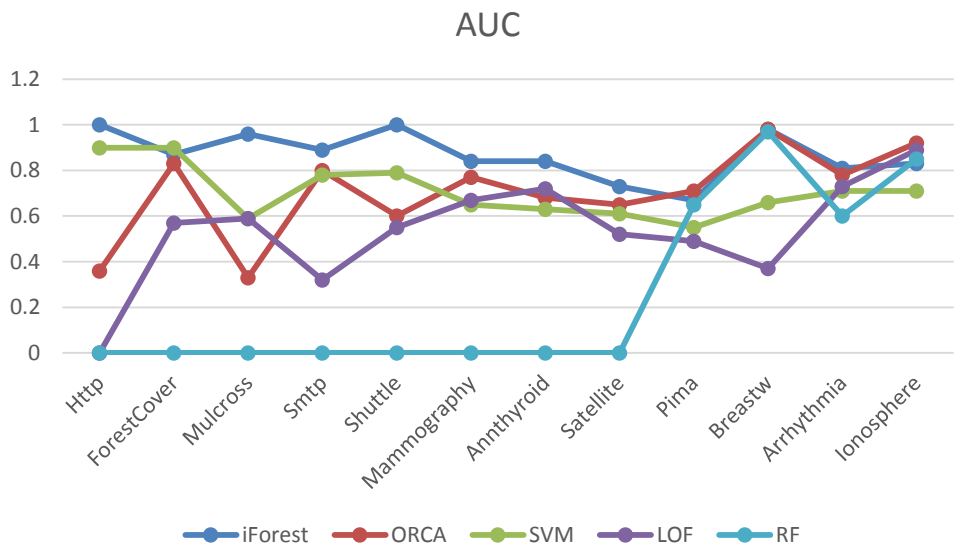


rank#	iForest	Orca	SVM	LOF	RF
1	**19**	**19**	10	**19**	11
2	**8**	**8**	11	**8**	10
3	10	**6**	12	**6**	13
4	**4**	**4**	**19**	**4**	**8**
5	**6**	10	1	7	7
6	20	7	2	10	**4**
7	12	12	3	12	12
8	7	11	**4**	18	9
9	11	9	5	13	**19**
10	13	13	**6**	11	**6**



优劣分析

- AUC和处理时间

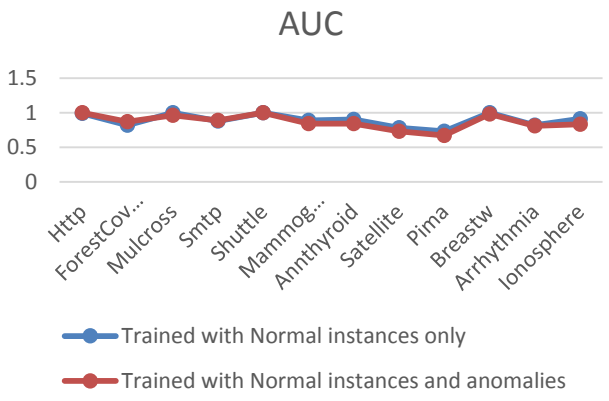


	Time(seconds)						
	iForest			ORCA	SVM	LOF	RF
	Train	Eval.	Total				
Http	0.25	15.33	**15.58**	9487.47	35872.09	*	**
ForestCover	0.76	15.57	**16.33**	6995.17	9737.81	224380.19	**
Mulcross	0.26	12.26	**12.52**	2512.2	7342.54	156044.13	**
Smtp	0.14	2.58	**2.72**	267.45	986.84	24280.65	**
Shuttle	0.3	2.83	**3.13**	156.66	332.09	7489.74	**
Mammography	0.16	0.5	**0.66**	4.49	10.8	14647	**
Annthyroid	0.15	0.36	**0.51**	2.32	4.18	72.02	**
Satellite	0.46	1.17	**1.63**	8.51	8.97	217.39	**
Pima	0.17	0.11	0.28	**0.06**	**0.06**	1.14	4.98
Breastw	0.17	0.11	0.28	**0.04**	0.07	1.77	3.1
Arrhythmia	2.12	0.86	2.98	0.49	**0.15**	6.35	2.32
Ionosphere	0.33	0.15	0.48	**0.04**	**0.04**	0.64	0.83

优劣分析

- 只使用正常样例训练
  - 训练集中只含有正常实例，评估集中含有正常和异常实例
  - 结果表明，包含异常与否对检测性能几乎没有影响
  - iForest可以描述数据分布，路径长度值与数据点相对应

	Trained with Normal instances only	Trained with Normal instances and anomalies
Http	0.99	**1.00**
ForestCover	0.82	**0.87**
Mulcross	**1.00**	0.96
Smtp	0.88	**0.89**
Shuttle	**1.00**	**1.00**
Mammography	**0.89**	0.84
Annthyroid	**0.90**	0.84
Satellite	**0.78**	0.73
Pima	**0.73**	0.67
Breastw	**1.00**	0.98
Arrhythmia	**0.82**	0.81
Ionosphere	($\psi = 128$) **0.91**	0.83



优劣分析

- IForest优势总结
  - IForest在运行时间、检测精度、内存需求方面有很好的表现，尤其是在大数据集中
  - 可以降低误报率和漏报率
  - 可以识别集群异常
  - 训练数据集中可以不包含异常
  - 可以提供多种粒度的检测结果，无需重新训练
  - 有能力处理含有不相关属性的高维数据

- IForest改进的方向
  - 对于分类数据的处理
    - 与连续值数据不同，分类数据没有排序信息，可能值有限。在这种情况下选择拆分就成了一个问题。
    - 此外，在隔离模型中存在混合数据类型的情况下，与连续值属性相比，由于分类属性取值可能少很多，导致分类属性的影响较小。
    - 当分类属性相关时，这会对检测性能产生不良影响。
  - 在线异常检测
    - 在数据流的早期开始检测，构建模型时只需要使用小的子样本

应用总结

- 欺诈检测
  - 信用诈骗、电信诈骗、信用卡盗刷等
- 入侵检测
  - 检测入侵计算机系统的行为
- 医疗领域
  - 检测人的健康是否异常
- 制药领域
  - 药物筛选的时候常常需要确定试验结果是否正常
- 数据去噪
  - 一些异常数据可能会导致数据的期望或者方差等严重偏离正常，检测出数据中的噪声通常是数据预处理中很重要的一步

参考文献

- Liu F T, Kai M T, Zhou Z H. Isolation Forest[C]// Eighth IEEE International Conference on Data Mining. IEEE Computer Society, 2008:413-422.
- Liu F T, Ting K M, Zhou Z H. Isolation-Based Anomaly Detection[J]. Acm Transactions on Knowledge Discovery from Data, 2012, 6(1):1-39.
- 源码：http://www.jianshu.com/p/5af3c66e0410

道德经

上善若水。水善利万物而不争，处众人之所恶，故几於道。居善地，心善渊与善仁，言善信，正善治，事善能，动善时。夫唯不争，故无尤。

谢谢！