基于 iForest 异常值检测法的反欺诈研究

导读

本文主要介绍异常值检测领域一种高效的非监督算法——孤立森林：该算法构建多棵孤立树将每个样本单独分成一类，通过衡量分类时各样本点被区分开的难易程度来寻找异常值。以往异常值检测方法多用有监督模型，需要使用历史数据对交易行为进行分类，所以这些模型只能识别历史数据中已有的诈骗手段，新的欺诈方式却难以识别。而孤立森林非监督的检测原理可以很好的克服这一问题，而且该算法还拥有高精度和线性时间复杂度的特点，非常适用于金融行业的反欺诈领域。

一、引言

孤立森林（Isolation Forest,简称 iForest）是刘飞博士在莫纳什大学就读期间在陈开明教授和周志华教授的指导下发表的一种高效的无监督异常值检测方法，具有线性时间复杂度和高精准度，适用于金融交易欺诈检测。

该方法主要思想就是对数据集建立很多棵分类树将数据集中的样本点分开，由于异常点在数据集中占比比较少，且在一些特征上与正常点有差异，所以将样本点进行分类时异常点更容易被区分开。而正常样本点由于占比较大且彼此之间特征差异很小故很难区分开。iForest 就是通过衡量分类时各样本点被区分开的难易程度来寻找异常值的，这一难易程度主要通过分类树中分类样本时所需的路径长度来衡量的，路径越长说明越难分开。这里举一个简单的例子说明这一原理：

	特征 1	特征 2	特征 3	特征 4	特征 5
正常点 1	1	1000	2	1	3
正常点 2	1	1005	1.9	1	4
正常点 3	1	998	2.2	1	2.9
异常点	0	11	0.03	-1	100

假设有 4 个样本，其中 1 个异常点，另外三个是正常点，每个样本有 5 个特征，且异常点在这些特征上都明显与正常点不一样。现在我们要构造一个分类树将这些样本分开，保证每个样本被单独分成一类（即一个叶节点只有一个样本）：任选一个特征，如特征 2，特征 2 取值区间为[11,1005]；再从特征 2 的取值区间内任选一个分割点，此时选取的分割点有极大的可能会来自区间[11,998]，而不太可能来自[998,1005]，所以这一特征的分割结果就是将异常点单独分为一类，3 个正常点分成一类；再继续选择特征和分割点进行切分，直至每个样本被单独分成一类。我们可以看到在这个例子中，由于异常点在各个特征上与正常点明显不一样，所以只需要任选一个特征（即在分类树中用一个节点进行切分）就可以和正常点分开。但正常点之间的特征很相似，需要多次切分才能被单独分开。iForest 就是通过衡量分类时各样本点被区分开的难易程度来寻找异常值。下面我们将介绍 iForest 的理论模型。

二、《数据结构》中的树

1.满二叉树

在正式介绍 iForest 的模型之前需要先简单介绍《数据结构》中的“满二叉树”的相关知识。在《数据结构》中“树”是数据的一种结构，从图像上看类似于机器学习中的“决策树”，它是由n个有限结点组成一个具有层次关系的集合。例如，图 1 就是《数据结构》中某种树的结构。

《数据结构》中“树”的类型有多种，其中有一种被称为满二叉树的结构，这里采用国外的定义，满二叉树被定义为树中的所有节点都有 2 个子节点或没有子节点，如图 2 所示就是满二叉树。若满二叉树的最后一层有n个节点，则其余层节点共有$n-1$个，整棵树总共有$2n-1$个节点。我们称二叉树最开始的那个节点为根节点，最后一层没有子节点的节点为叶节点。

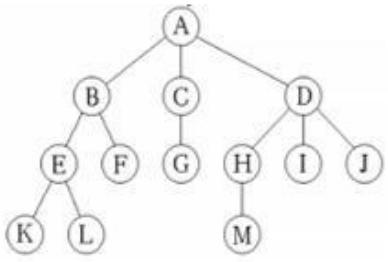


图1《数据结构》中树举例图

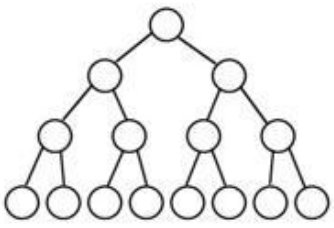


图2 满二叉树图

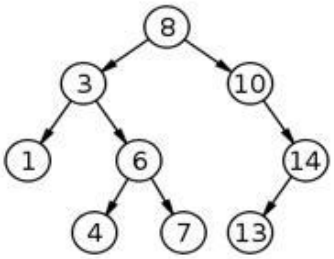


图3 二叉搜索树图

2. 二叉搜索树

二叉搜索树也是《数据结构》中“树”的一种，它或者是一棵空树，或者是具有下列性质的二叉树：若它的左子树不空，则左子树上所有结点的值均小于它的根结点的值；若它的右子树不空，则右子树上所有结点的值均大于它的根结点的值；它的左、右子树也分别为二叉排序树。例如，图 3 就是《数据结构》中二叉搜索树的结构。

其实可以把二叉搜索树理解为一种存储数据的方式，可以通过二叉搜索树来查找存储在其中的数据，这样也就对应了查找数据所需要的次数。例如，我们想通过图 3 的二叉搜索树查找 13，第一步发现 8 比 13 小，则往 8 的右边搜索；第二步发现 10 比 13 小，则往 10 的右边搜索；第三步发现 14 比 13 大，则往 14 的左边搜索，从而找到了 13。这时搜索是成功的，需要 3 步。再例如，我们想通过图 3 的二叉搜索树查找 15，第一步发现 8 比 15 小，则往 8 的右边搜索；第二步发现 10 比 15 小，则往 10 的右边搜索；第三步发现 14 比 15 小，则往 14 的右边搜索，但 14 的右边为空，这时将在 14 的右边插入 15 这个值。这时搜索是不成功的，也需要 3 步。

从而我们可以发现，二叉搜索树中查找数据时有成功和不成功两种结果，若成功则返回对应的值，若不成功则在相应位置插入我们所查找的值作为新值。成功或不成功都有查找次数这个值。

三、孤立树

1. 算法思路

孤立森林是由很多棵孤立树（Isolation Tree,简称 iTree）组成的一种集成的模型，每棵 iTree 都是一棵满二叉树，树中的所有节点都有 2 个子节点或没有子节点。每棵 iTree 的建立过程如下：

给定含 n 个样本的集合$X=\{x_1 x_2 ... x_n\}$通过随机选择数据集的特征q和随机选择特征的分裂值p来递归样本集X，从而建立 iTree。递归过程直到满足以下三个条件之一才停止：①iTree 的深度达到限定的最大值；②某次递归后 iTree 的节点只有一个样本；③某次递归后 iTree 的节点所包含的数据都有相同的值。在假定样本集$X=\{x_1 x_2 ... x_n\}$中所有点都不同的情况下，当 iTree 充分生长时，样本集中的每个样本x_i都会是 iTree 的叶节点。iTree 的算法思路如下所示：

Algotithm：*iTree*(X, e, l)

Inputs: X - input data, e – current tree height, l- height limit Output: an iTree

1: if $e \geq 1$ or $|X| \leq 1$ 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 from 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 $inNode\{Left \leftarrow iTree(X_l, e+1, l)$

10: $Right \leftarrow iTree(X_r, e+1, l)$

11: $SplitAtt \leftarrow q$

12: $SplitValue \leftarrow p$

13: end if

---

2. iTree 中的评价指标

（1）路径长度

这里定义 iTree 中节点 x的路径长度 $h(x)$为从根节点贯穿到 x 所在的叶节点的边的数量。

（2）路径长度平均值

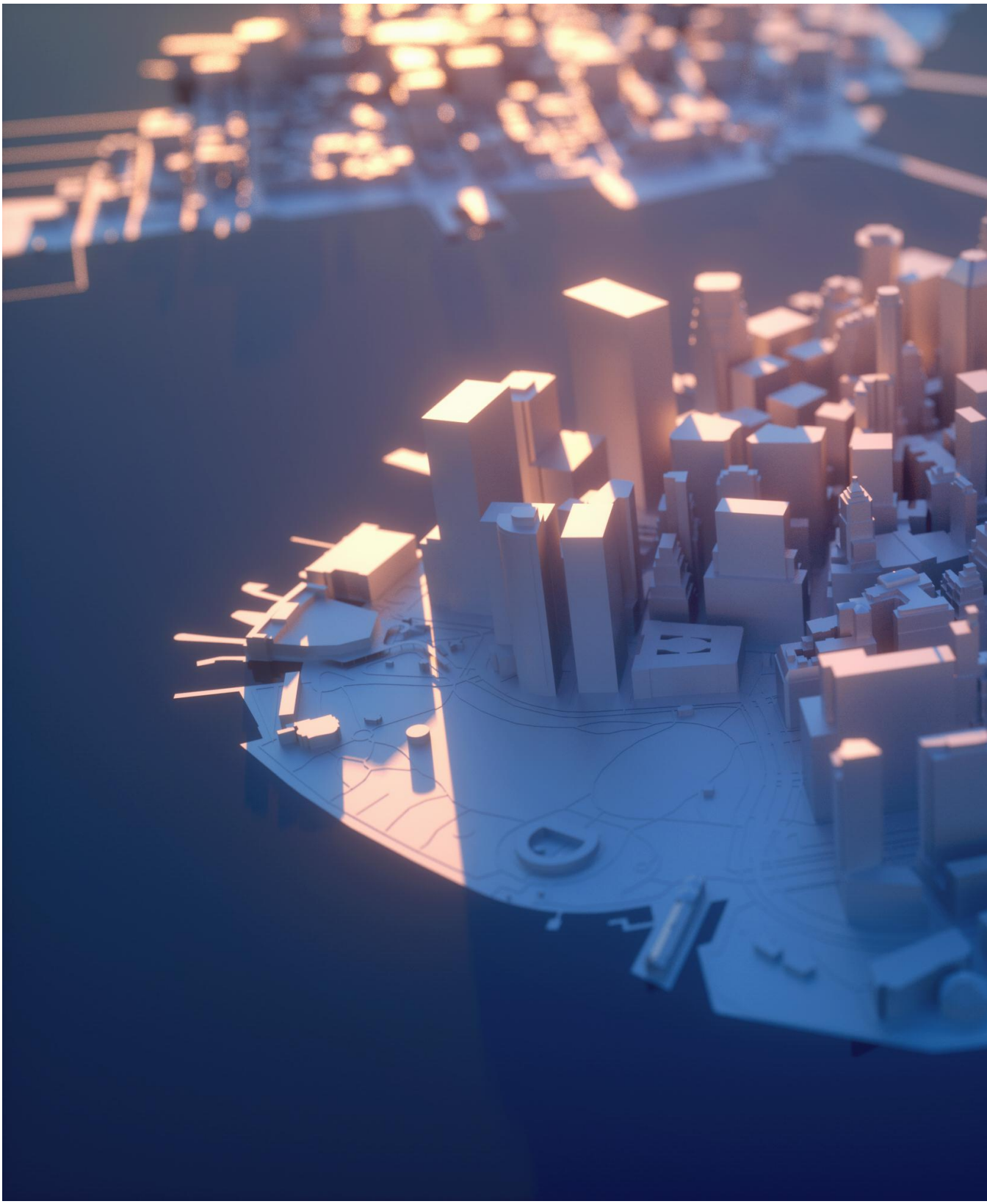
iTree 中叶节点的平均路径长度 $h(x)$与二叉搜索树不成功查找时所需的路径长度相同，所以这里借用二叉搜索树的相关知识来定义 iTree 中叶节点的平均路径长度 $h(x)$。给定一个含 n 个样本的数据集，二叉搜索树中不成功查找的平均路径长度为：

$$c(n) = 2H(n-1) - \frac{2(n-1)}{n}$$

其中，$H(i)$是调和级数，$H(i)=ln(i)+0.57721566 49$（欧拉常数）。$c(n)$也正是给定n个样本时，一棵 iTree 路径长度 $h(x)$的平均值。在样本量固定 n 时，不同 iTree$c(n)$其实是相同的。

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

其中，iForest 中有很多棵 iTree，每棵 iTree 的叶节点x都有一个路径长度$h(x)$，$E(h(x))$是 iForest 中这些路径长度的平均值。从异常值得分$s(x,n)$的定义可知，$s(x,n)$是一个关于$h(x)$的单调函数，$0<s(x,n)<1$，且$s(x,n)$越大则对应的样本点x越有可能是异常点。



四、孤立森林

算法思路：iForest 就是由多棵 iTree 组成的一种集成模型。根据刘飞博士等的研究表明，与传统的机器学习算法要求大样本相比， iForest 在小样本时的表现更好。所以根据他们的建议，在使用 iForest 检测异常值时，先对较大的原始样本量进行多次抽样（设 t 次）， 每次随机抽出一部分数据（设Ψ个数据） 建立一棵 iTree，多次抽样可以建立多棵 iTree（t 棵）组合成 iForest。iForest 的算法思路如下所示：

Algotithm:*iForest*(X, t, Ψ)

Input:X-input data,t-number of trees,Ψ-smapling size

Output:a set of t *iTrees*

1: Initialize *Forest*

2: set height limit $l = ceiling(\log_2 \Psi)$

3: for $i = 1$ to t do

4: $X' \leftarrow sample(X, \Psi)$

5: $Forest \leftarrow Forest \cup iTree(X', 0, l)$

6: end for

7: return Forest

当 iForest 模型建立好之后，通过再计算各个样本点的异常值得分来判断该样本点是否为异常值，得分越高越有可能是异常值。现在 Python 的 sklearn 库里已经集成了 iForest 的算法，有需要的同学可以联系本文的作者向笔者索取或自行百度搜索相关代码。

金融领域的欺诈行为往往在某些地方与正常行为不一样，这些欺诈行为可以被认为是数据中的异常值，以往的检测方法一般是构建有监督的分类模型来进行分析：一次交易或一个客户是一个样本，有风险是一类，无风险是另一类，交易行为的一些信息是样本的特征，通过构建分类模型来识别这次交易（或这个客户）是否欺诈。

这些模型的数据来自已有的数据，这导致这些模型只能识别历史数据中已经存在的诈骗手段，而新的欺诈方式由于不在历史数据中故难以识别，这极大的制约了金融交易欺诈检测领域数据挖掘模型的应用。

而 iForest 是一种无监督的方法，只需要知道样本点的特征即可，无需确定样本点的所属类别，这样一来传统有监督的金融反欺诈模型只能识别历史已有的欺诈模式这一局限性就大大减小了，这特别适用于金融领域的欺诈识别。而且 iForest 在保证高精度的同时还只具有线性的时间复杂度，模型的计算量也较小。

参考资料

[1] Aggarwal, Charu C. Outlier Analysis, 2nd ed[M]. Berlin, Germany:Springer. 2016.

[2]Liu F T , Ting K M , Zhou Z H . Isolation Forest[C]// Data Mining, 2008. ICDM '08. Eighth IEEE International Conference on. IEEE, 2009.

[3]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.

[4]B. R. Preiss. Data Structures and Algorithms with ObjectOriented Design Patterns in Java. Wiley, 1999.