# 第9章 异常检测

异常检测又叫离群点检测，通常用于结构缺陷检测、医疗疾病发现、文本错误纠正等。

**异常检测和聚类分析是两项高度相似的任务，但目的不同。聚类分析发现数据集中的模式，而异常检测则试图捕捉那些显著偏离多数模式的异常情况。**基于聚类的异常点检测是用聚类方式将数据划分为不同的簇，计算簇内每个点与簇中心的相对距离，相对距离较大的点被视为异常点。

异常检测在风控领域的应用非常广泛，主要包括建模前的**样本清洗**、**个体欺诈检测模型**、**PreA预筛选模型**和**冷启动模型**等。如刷单、作弊等难以直接通过标签验证的场景，初期常基于专家经验挑选特征（从业务角度、欺诈手法出发），并通过异常检测模型进行欺诈识别，以在保障平台权益的前提下进行数据积累。

对于**样本清洗**，可以采用异常检测模型剔除极端的异常值，比较剔除前后的数据集建模结果的好坏，如果变好可以剔除，如果变差不能剔除。

对于**欺诈检测**，可以细分为个体欺诈检测与团伙欺诈检测。其中，个体欺诈具有占比极小、与整体显著不同的特点，这与离群点的性质相同，因此常将异常检测技术用于个体欺诈检测。在实践中，配合相关的业务经验，可以达到较好的效果。团伙欺诈检测的中心思想为团伙发现，在金融领域，聚集就意味着风险，因此通常使用基于图的社区发现算法进行团伙欺诈检测。

对于欺诈检测，可以从业务角度、欺诈手法出发，找出可能相关的欺诈变量，做无监督后有监督，一般可以提高模型的泛化能力。

对于**冷启动**，可以根据经验挑选一些有区分度的特征，使用异常检测模型进行无监督训练，得到集欺诈检测与信用评级功能于一体的冷启动模型。

无监督模型的建模难点并不在于模型，而在于**特征的选取**。由于没有标签，特征的构造并不能通过数据分析手段进行，因此通常需要结合领域知识进行精准地特征构造。

异常检测的**特点**：

（1）异常不一定代表是坏的事情，但往往是有价值的事情，我们对异常的成因感兴趣。

（2）异常检测往往是在**无监督的模式下**完成的——历史数据中没有标签，我们不知道哪些数据是异常。因此无法用监督学习去检测。

异常检测的**应用**：

（1）金融行业的反欺诈、信用卡诈骗检测：把欺诈或者金融风险当做异常；

（2）罕见病检测：把罕见病当做异常，比如检测早发的阿兹海默症；

（3）入侵检测：把网络流量中的入侵当做异常；

（4）机器故障检测：实时监测发现或预测机械故障；

（5）图结构、群体检测：比如检测疫情的爆发点等；

异常检测的常用**算法**：

（1）线性模型：PCA

（2）基于相似度度量的算法：KNN、LOF、HBOS

（3）基于概率的算法：COPOD

（4）集成检测算法：孤立森林（IF）、XGBOD

（5）神经网络算法：自编码器（AutoEncoder）

## 9.1 LOF

异常局部因子（LOF）是一种基于密度的异常检测方法。LOF通过局部可达密度刻画数据密度，并用它衡量样本的异常程度。

### 9.1.1 算法流程

LOF算法与KNN类似，不过度量方法不同，它将每一个样本到其k近邻样本的距离量化为一种**密度**的概念，称为**局部可达密度**。在该算法中，需要确定近邻样本个数k，然后根据相应公式得到异常分数。

LOF算法的实现流程如下：

（1）首先对样本空间进行去重，分别计算每一个样本到样本空间内其余点的距离。

（2）将步骤1中的距离升序排列。

（3）指定近邻样本个数k，对于每个样本点，寻找其k近邻样本，然后计算LOF分数，作为异常分数，这种异常分数是由局部可达密度计算得到的。

### 9.1.2 原理

（1）求k-近邻距离：在距离样本点p最近的n个点中（n<k），第k个最近点跟点p之间的距离称为点p的k-近邻距离，记为k-distance(p)。

（2）求k-近邻：样本点p的k-近邻，表示和样本点p的距离小于等于k-distance(p)的样本点。

（3）求可达距离：样本点p到样本点o的可达距离reach\_dist(p,o)，为样本点o的k-近邻距离和样本点p与o之间的直接距离d(p,o)二者中较大的值。

（4）求局部可达距离：样本点p的局部可达距离是样本点p与其k-近邻的平均可达距离的倒数。

（5）求局部异常因子：样本点p的局部相对密度（局部异常因子）为样本点p的k近邻的平均局部可达密度与样本点p的局部可达密度的比值。所以，样本点p的局部可达密度相比于其k-近邻的平均局部可达密度越小，其异常程度越大。

一句话解释LOF：一个点的密度相比于k-近邻点的密度越小，那么这个点越可能是异常点。

## 9.2 IF

孤立森林（IF）是一种基于空间随机划分思想的集成算法，由多颗二叉树并行得到，再将输出结果进行加权平均。IF的每颗孤立树（iTree）中，特征及特征值的选择是完全从数据中随机选取的。

### 9.2.1 算法过程

（1）从样本空间中随机选择一部分样本，从特征空间随机选择一个特征，即行列采样；

（2）在现有特征维度上随机选取一个特征值作为划分节点，即阈值；

（3）分化决策树，左枝放入小于等于该阈值的样本，右枝放入大于该阈值的样本；

（4）重复上述过程，直到数据不再可分，或者当前树的分化达到了开始设定的二叉树深度。

### 9.2.2 原理

（1）路径长度

样本xi在当前有T个样本的孤立树上的路径长度h(xi)的计算公式为：



其中，ei表示样本xi从一颗孤立树的根部游走到当前节点的边的个数，C(T)表示使用T个样本训练的二叉树的平均路长。

（2）异常分



其中，表示xi在所有孤立树上的路径长度的均值，表示一棵孤立树上训练样本的个数，表示用个样本训练的二叉树的平均路径长度，作为归一化项。

从IF的异常分计算方式来看，数据x在多颗孤立树中的平均路径长度越短，得分越接近1，表明数据x越异常；数据x在多颗孤立树中的平均路径长度越长，得分越接近0，表示数据x越正常。数据x在多颗孤立树中的平均路径长度越接近整体均值，则打分会在0.5附近。

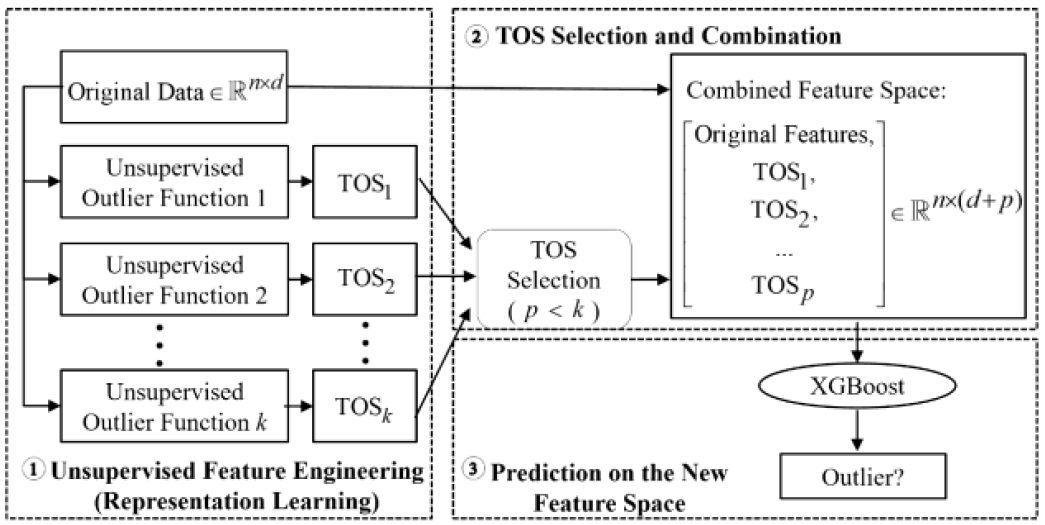
## 9.3 XGBOD

### 9.3.1 算法过程

XGBOD是一种使用无监督表示学习提升有监督异常点检测的方法。该方法首先采用无监督异常检测算法从数据中提取丰富的数据表示，然后将这些新产生的特征与原始特征组合起来构成增强后的特征空间，最后在增强后的特征空间上使用XGBoost分类器。

XGBOD框架包括三步：

* 1. 使用无监督异常检测算法提升数据表示；
  2. 通过贪心算法提取有用的数据表示；
  3. 在增强后的特征空间上使用XGBoost分类器进行预测；



### 9.3.2 原理

（1）无监督表示学习

对于原始特征空间X，采用不同的异常检测算法计算异常分数：



（2）TOS选择

X与新产生的数据表示组合得到新的特征空间：



然后可以使用三种方法筛选有用的数据表示：

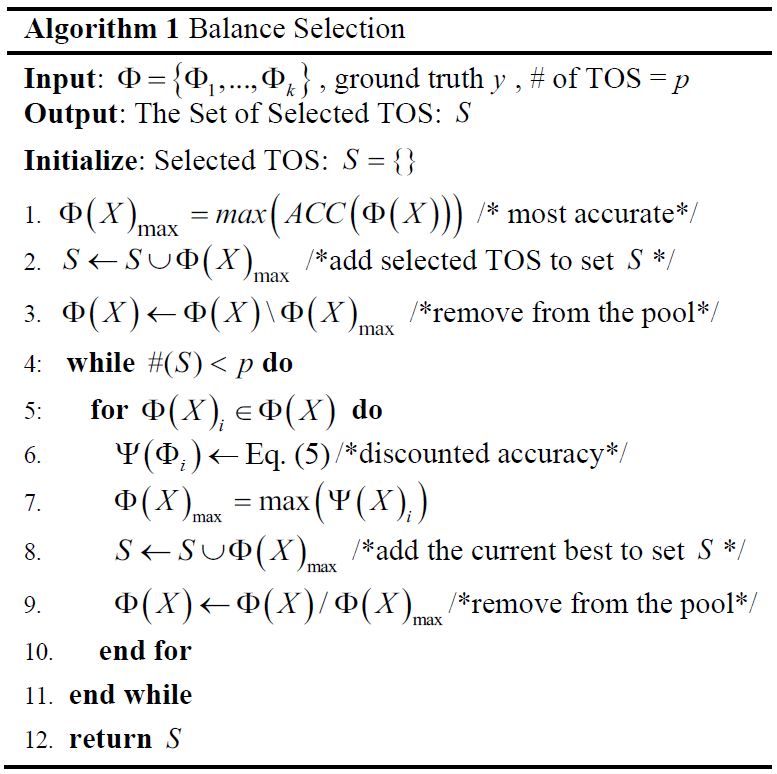
* 随机选择：从产生的TOS中随机选择p个加入S；
* 精确选择：从中选择最准确的TOS，即ROC值最大的TOS：



* 平衡选择：从平衡多样性和准确性的角度选择TOS：



其中，表示一个TOS与S中已经选择的TOS之间的皮尔森相关系数，相关系数越大，平衡分数越小。



（3）XGBoost预测

XGBoost分类器在新的特征空间上预测，产生最终的输出。

（4）限制

平衡选择更适用于在高维特征空间中提取有用信息，准确选择更适用于低维空间数据，随机选择有时候更有用，然而，最终的结果一般无法预测，可根据需要尝试。

## 9.4 Auto Encoder

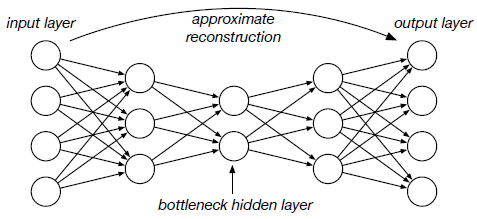
### 9.4.1 算法过程

Auto Encoder (AE，自动编码器)是一种无监督的学习数据表示的神经网络方法，同PCA类似，AE可以通过计算重建误差检测数据中的异常点。

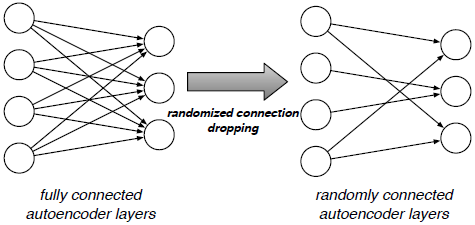
AE的输出层的节点数与输入层的节点数相同，并且结构是分层且对称的， AE的目标是训练输出以尽可能紧密地重建输入。由于AE创建了低维的数据表示，因此该算法很自然地可以应用于异常点发现，这里的基本思想是离群点比正常点更难准确地表示。因此，在重建异常值时，误差会很大，这提供了对数据点进行评分的方法。

Auto Encoder算法包括三步：

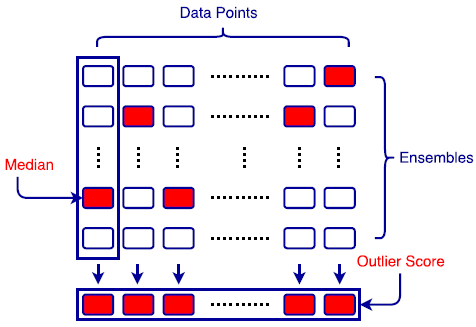
（1）构建全连接的自动编码器。



（2）对自动编码器采用集成学习方法获得更高的准确性，也就是采用Dropout方法得一系列完全独立的神经网络，然后将结果组合起来。



（3）样本点在所有的集成子模型上进行打分，每个样本点取得分的中位数作为异常分数。



### 9.4.1 原理

（1）构建自动编码器

首先构建全连接的自动编码器，在网络的两端使用sigmoid函数，并在其他层使用ReLU。这样有两个好处：一方面，使用sigmoid函数将确保即使所有中间层ReLU神经元都已死亡，至少我们在网络的两端有两层正常工作，这可以确保改善最坏的情况。另一方面，ReLU死亡的问题通常在梯度太大时发生，由sigmoid函数引起的梯度消失实际上有助于防止ReLU单元在反向传播过程中死亡。

假设两层节点之间有L1\*L2个连接，我们对这些连接进行采样，在采样的L1\*L2个连接中，将有一些冗余（或重复）的连接和一些丢失的连接，保留我们选择的连接形成随机连接自动编码器。

（2）异常点评分

假如我们有m个集成子模型，训练集包含n个点和d个维度，第i个子模型输入的第j个数据点为，自动编码器重建的输出为，故数据点的重建偏差为：



数据点的重建偏差越大，说明越有可能是异常点。

注：异常检测算法可参考工具包[Pyod](https://pyod.readthedocs.io/en/latest/pyod.html)，github参考[yzhao062/pyod](https://github.com/yzhao062/pyod)。

## 9.5 评估方法

评估方法不能简单用准确度（accuracy），因为数据的极端不平衡。我们可以使用如下指标衡量：

（1）ROC-AUC曲线；

（2）Precision @rank-K：top K的精准度

（3）Average Precision：平均精准度

## 9.6 实践技巧

1、**假设数据是有标签的，优先使用监督学习**，比如xgboost。如果数据量不是非常大，也可以尝试xgbod。如果是无监督的情况下，可以参考下面的流程。

2、如果不知道如何选择合适的异常检测模型，可以使用MetaOD进行自动模型选择。如果不知道该选什么模型，优先选择孤立森林。

3、**如果是手动选择模型的话，首先要考虑数据量和数据结构。**当数据量比较大（>10万条，>100个特征），优先选择可扩展性强的算法，比如孤立森林、HBOS和COPOD。

4、如果最终的结果需要一定的可解释性，可以选择孤立森林或者COPOD。

5、如果数据量不大，且追求精度比较高的结果，可以尝试随机训练多个检测模型，并使用 LSCP来进行合并。

6、如果训练和预测过程比较缓慢、开销大的话，可以使用SUOD进行加速。

7、**如果数据量大、特征多，可以尝试用基于神经网络的方法**，并有GPU并行等方法。

## 9.7 实例：识别洗钱客户

（1）建模目标

本赛题旨在提升对于无监督/半监督学习算法的运用能力和特征提取的技巧。建模的目标就是通过充分挖掘对公客户开户时提供的29要素信息，在无好坏标签的情况下，识别出客户是否异常。这里异常是指具有洗钱嫌疑，例如在表现期内出现与洗钱相关的违法犯罪行为或者被行内反洗钱专家和人工规则识别为洗钱客户的情况。

（2）数据简介

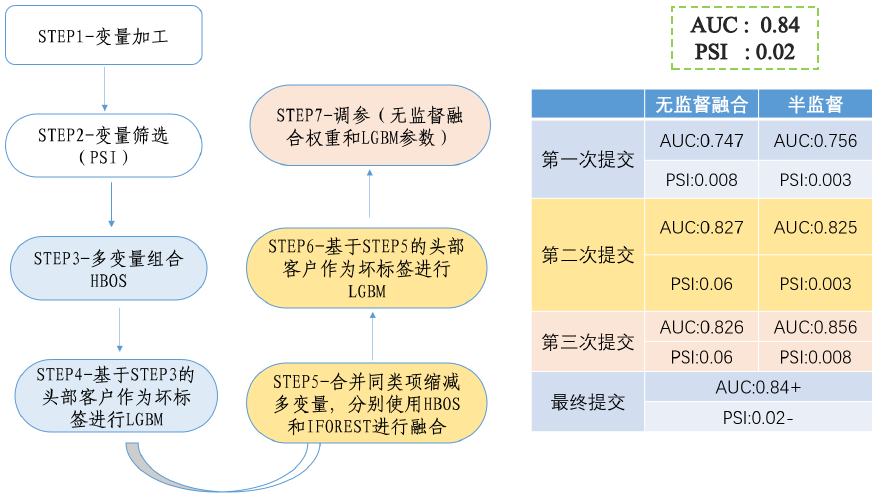
数据集由两类客户组成，一类客户为新开户并且在表现期内正常的公司类客户，另一类客户则为在表现期内异常的公司类客户，即模型需要识别的异常客户。X变量由对公客户类型代码、客户名称、住所和经营范围等29要素组成，并已进行脱敏和初步衍生，Y标签为在表现期内经业务规则判定的好坏标志。出于保密性和比赛设计要求，本次比赛不公布Y标签生成规则。

（3）评价指标

AUC、PSI

（4）解决方案

方案一：



方案二：

