异常检测：算法、应用与实践

赵越
卡内基梅隆大学

Carnegie Mellon University

关于我

研究与工作方向:

异常检测、自动机器学习（AutoML）、机器学习系统（MLSystem）。开发超过10个机器学习工具库与系统，GitHub星标9,000，被下载超过300万次。

教育背景:

博士（在读）：卡内基梅隆大学

硕士：多伦多大学

本科：辛辛那提大学

高中：山西省实验中学

工作:

2021：斯坦福大学 – 访问/实习 (with Prof. Jure Leskovec)

2020：艾坤伟（IQVIA）– 实习

2017-2019年：普华永道（高级数据科学家）

一键三连不迷路

Yue Zhao
yzhao062

Ph.D. Student @ CMU. Scalable ML Systems/Algorithms | AutoML | Anomaly/Outlier Detection | Information Systems Twitter@ yzhao062

Edit profile

1.4k followers · 20 following · ☆ 66

Carnegie Mellon University
Pittsburgh, PA, USA
zhaoy@cmu.edu
https://www.andrew.cmu.edu/user/yuez...

Highlights
Arctic Code Vault Contributor
PRO

Carnegie Mellon University

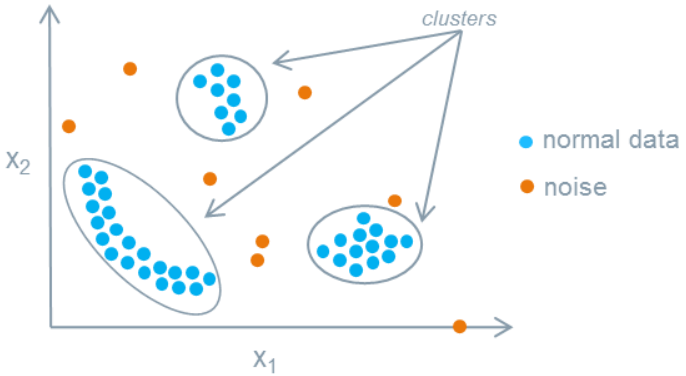
内容概览

1. 什么是异常检测?
2. 异常检测有什么具体应用?
3. 异常检测的工具概览？如何用10行Python代码进行异常检测?
4. 异常检测算法概览与主流模型介绍
5. 面对各种各样的模型，如何选择和调参?
6. 未来的异常检测研究方向
7. 异常检测相关的资源汇总（书籍、讲座、代码、数据等）

异常检测（Anomaly Detection）

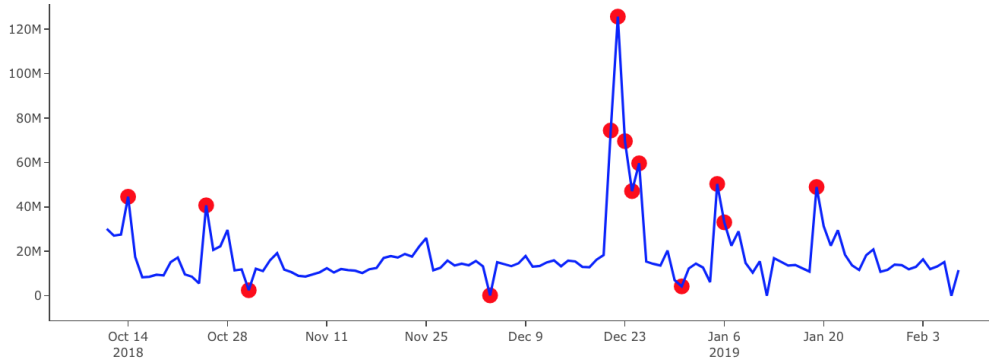
什么是异常值、离群点（anomaly）？

异常一般指的是与标准值（或期待值）有偏离的样本，也就是说跟绝大部分数据"长的不一样"。



表格式数据中的异常

Source: https://developer.mindsphere.io/apis/analytics-anomalydetection/api-anomalydetection-overview.html



时间序列上的异常

Source: https://towardsdatascience.com/anomaly-detection-with-isolation-forest-visualization-23cd75c281e2

Carnegie Mellon University

异常检测（Anomaly Detection）

异常检测的一些特点：

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

异常检测应用

Intel ControlFlag：

“基于 10 亿条包含各种错误的未标记生产质量代码的机器学习培训，ControlFlag 得以通过“异常检测”技术，对传统编程模式展开筛查。无论使用的是哪种编程语言，它都能够有效地识别代码中可能导致任何错误的潜在异常。”

Amazon AWS CloudWatch:

“今天，我们将通过一项新功能增强 CloudWatch，它将帮助您更有效地使用 CloudWatch 警报。... 我们的用户可以构建自定义的控制面板，设置警报并依靠 CloudWatch 来提醒自己影响其应用程序性能或可靠性的问题。”

Google：

“Google Analytics（分析）会选择一段时期的历史数据来训练其预测模型。要检测每天的异常情况，训练期为 90 天。要检测每周的异常情况，训练期为 32 周。”

异常检测的挑战与难点

异常检测面临很多挑战：

1. 大部分情况下是无监督学习，没有标签信息可以使用
2. 数据是极端不平衡的（异常点仅占总体数据的一小部分），建模难度大
3. 检测方法往往涉及到密度估计，需要进行大量的距离/相似度计算，运算开销大
4. 在实际场景中往往需要实时检测，这比离线检测的技术难度更高
5. 在实际场景中，我们常常需要同时处理很多案例，运算开销大
6. 解释性比较差，我们很难给出异常检测的原因，尤其是在高维数据上。但业务方需要了解异常成因
7. 在实际场景中，我们往往有一些检测的历史规则，如何与学习模型进行整合

以往几十年的经验如何与现在的模型相融合

异常检测工具概览

Python：

1. PyOD：超过30种算法，从经典模型到深度学习模型一应俱全，和sklearn的用法一致
2. Scikit-Learn：包含了4种常见的算法，简单易用
3. TODS：与PyOD类似，包含多种时间序列上的异常检测算法

Java：

1. ELKI: Environment for Developing KDD-Applications Supported by Index-Structures
2. RapidMiner异常检测扩展

R：

1. outliers package:
2. AnomalyDetection:

异常检测工具库：PyOD （JMLR‘ 19）

Python Outlier Detection Toolbox（https://github.com/yzhao062/pyod）：

- GitHub上约4000星标，下载量150万次，是异常检测任务的主流工具
- 论文发表在期刊JMLR上，被引用过百次。Zhao, Y., Nasrullah, Z. and Li, Z., 2019. PyOD: A Python Toolbox for Scalable Outlier Detection. *Journal of machine learning research (JMLR)*, 20(96), pp.1-7.
- 所有模型（约30个）均配备有统一且易用的API、文档和示例，使用并行化和JIT加速

```python
from pyod.models.knn import KNN   # kNN detector

# train kNN detector
clf_name = 'KNN'
clf = KNN()
clf.fit(X_train)

# get the prediction label and outlier scores of the training data
y_train_pred = clf.labels_  # binary labels (0: inliers, 1: outliers)
y_train_scores = clf.decision_scores_  # raw outlier scores

# get the prediction on the test data
y_test_pred = clf.predict(X_test)  # outlier labels (0 or 1)
y_test_scores = clf.decision_function(X_test)  # outlier scores
```

- 详细介绍：https://zhuanlan.zhihu.com/p/58313521

Carnegie Mellon University

异常检测算法

异常检测算法可以大致被分为：

- 线性模型（Linear Model）：PCA
- 基于相似度的度量的算法 （Proximity-based Model）：kNN, LOF，HBOS
- 基于概率的算法（Probabilistic Model）：COPOD
- 集成检测算法（Ensemble Model）：孤立森林（Isolation Forest），XGBOD
- 神经网络算法（Neural Networks）：自编码器（AutoEncoder）

评估方法也不能简单用准确度（accuracy），因为数据的极端不平衡

- ROC-AUC曲线
- Precision @ Rank k：top k的精准
- Average Precision：平均精准度

异常检测方法1：k-近邻检测

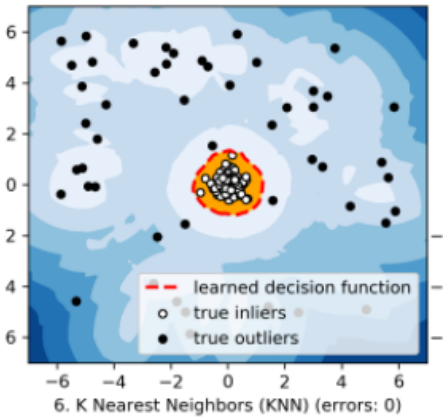
思路：计算每个点距离第k个近邻的距离，如果这个距离越大，就越可能是异常点。

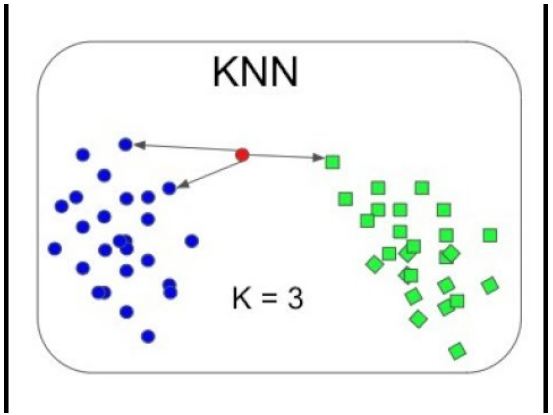
调用方法：from pyod.models.knn import KNN

优点：简单易懂、实现简单

缺点：因为需要计算距离，运算开销大，不适合高维大量数据

决策边界





Ramaswamy, S., Rastogi, R. and Shim, K., 2000, May. Efficient algorithms for mining outliers from large data sets. *ACM Sigmod Record*, 29(2), pp. 427-438.

异常检测方法2：LOF

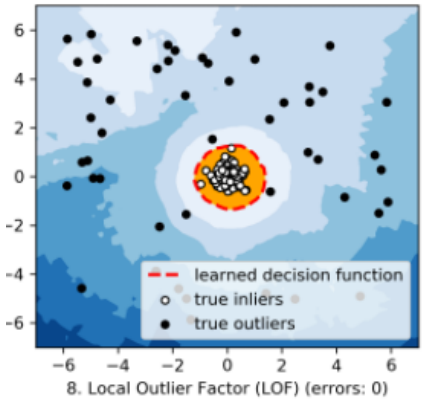
思路：计算每个点在局部区域上（local region）的密度和其邻域点的密度来判断异常程度

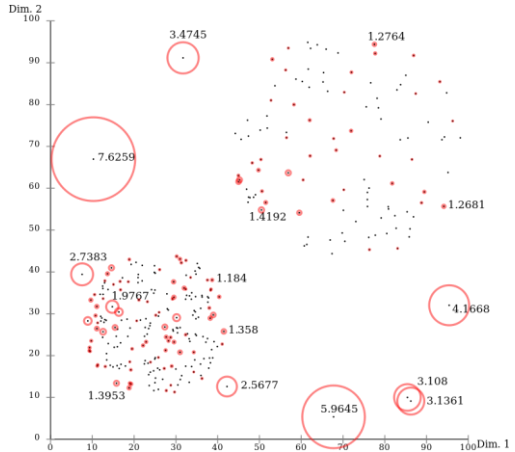
调用方法：from pyod.models.lof import LOF

优点：简单易懂、实现简单

缺点：因为需要计算距离，运算开销大，不适合高维、大数据

决策边界





Breunig, M.M., Kriegel, H.P., Ng, R.T. and Sander, J., 2000, May. LOF: identifying density-based local outliers. *ACM Sigmod Record*, 29(2), pp. 93-104.

Carnegie Mellon University

异常检测方法3：HBOS

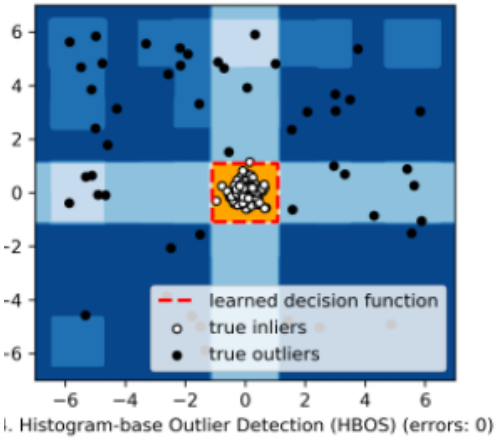
思路：假设每个维度间都是独立的，分别计算一个样本在不同维度上所处的密度空间，并叠加结果。

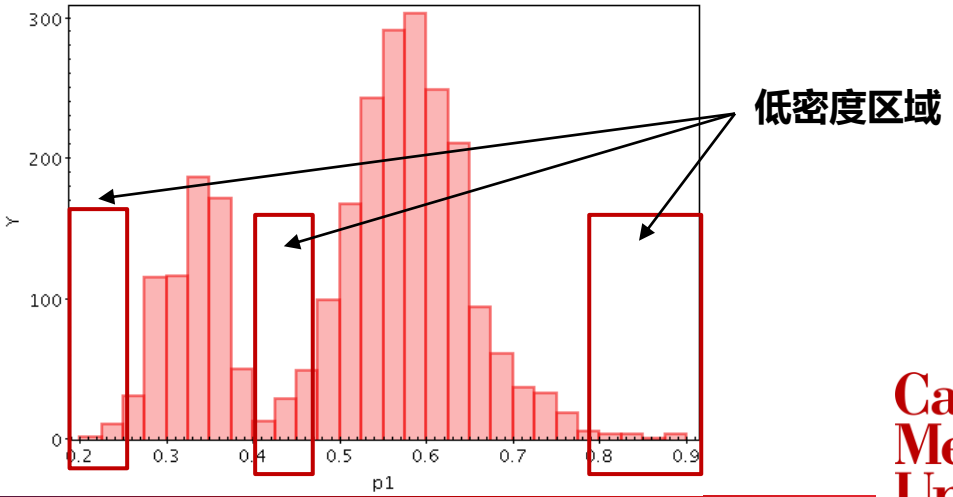
调用方法：from pyod.models.hbos import HBOS

优点：简单、开销小，可并行式计算，适用于大量数据

缺点：无法考虑不同特征间的关系（dependency），虽然在实际使用中效果很好

决策边界





Goldstein, M. and Dengel, A., 2012. Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm. In *KI-2012: Poster*

Carnegie Mellon University

异常检测方法4：COPOD (ICDM’ 20)

大致思路：计算一个（多维）样本在每个维度上是异常点的可能性，再合并所有的维度上的结果

调用方法：from pyod.models.copod import COPOD

优点：运行快、无需调参、有一定的解释性（比如哪个特征更有解释性）

缺点：决策边界比较复杂

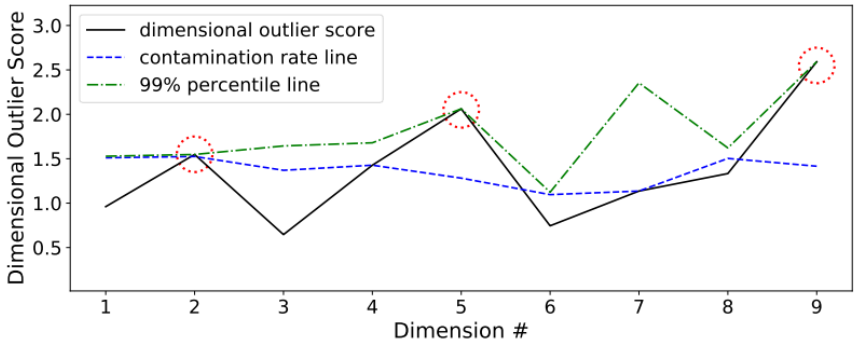


Fig. 2. *Dimensional Outlier Graph* for sample 70 (row 70); dimension 2, 5, 9 are identified as outlier subspaces.

Algorithm 1 Copula Outlier Detector

Inputs: input data X
Outputs: outlier scores $O(X)$

for each dimension d **do**
  Compute left tail ECDFs: $\hat{F}_d(x) = \frac{1}{n}\sum_1^n \mathbb{I}(X_i \leq x)$
  Compute right tail ECDFs: $\hat{F}_d(x) = \frac{1}{n}\sum_1^n \mathbb{I}(-X_i \leq -x)$
  Compute the skewness coefficient according to Equation 11.
end for
for each i in $1, \ldots n$ **do**
  Compute empirical copula observations
    $\hat{U}_{d,i} = \hat{F}_d(x_i)$,
    $\hat{V}_{d,i} = \hat{F}_d(x_i)$
    $\hat{W}_{d,i} = \hat{U}_{d,i}$ if $b_d < 0$ otherwise $\hat{V}_{d,i}$
  Calculate tail probabilities of X_i as follows:
    $p_l = -\sum_{j=1}^d \log(\hat{U}_{j,i})$
    $p_r = -\sum_{j=1}^d \log(\hat{V}_{j,i})$
    $p_s = -\sum_{j=1}^d \log(\hat{W}_{j,i})$
  Outlier Score $O(x_i) = \max\{p_l,\ p_r,\ p_s\}$
end for
Return $O(X) = [O(x_1), \ldots, O(x_d)]^T$

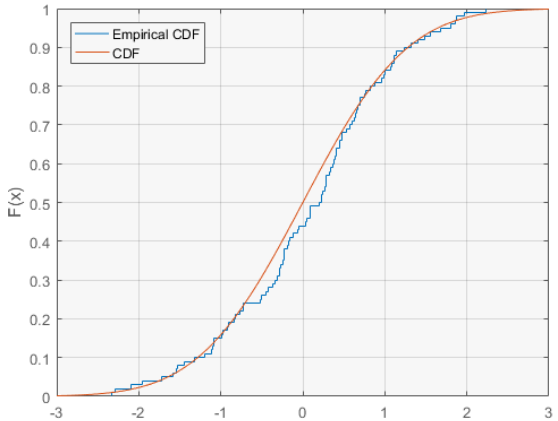
详细介绍：https://zhuanlan.zhihu.com/p/338189299

Carnegie Mellon University

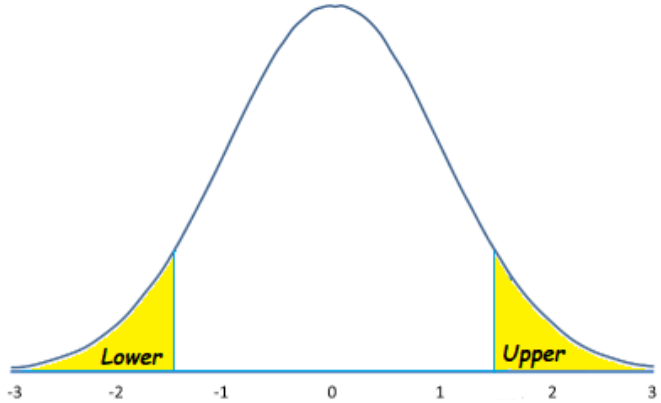
Li, Z., Zhao, Y., Botta, N., Ionescu, C. and Hu, X. COPOD: Copula-Based Outlier Detection. *IEEE International Conference on Data Mining (ICDM)*, 2020.

异常检测方法4：COPOD (ICDM’ 20)

核心思路：传统的统计方法会把特征空间上的每个维度单独建模，然后估算一个样本在这个维度上的“异常程度”（比如是否在2个σ以外），但这种思想忽略了不同维度间并不独立（特征点并非是完全独立的）。**因此通过copula函数（一种概率模型），我们可以对所有的特征一并联合建模**，估算出“经验累积分布函数”（empirical CDF），并用它来估计每个样本在不同维度上有多么“异常”。



计算经验累积分布函数



估算一个样本属于tail的概率

Carnegie Mellon University

Li, Z., Zhao, Y., Botta, N., Ionescu, C. and Hu, X. COPOD: Copula-Based Outlier Detection. *IEEE International Conference on Data Mining (ICDM)*, 2020.

异常检测方法5：孤立森林（Isolation Forest）

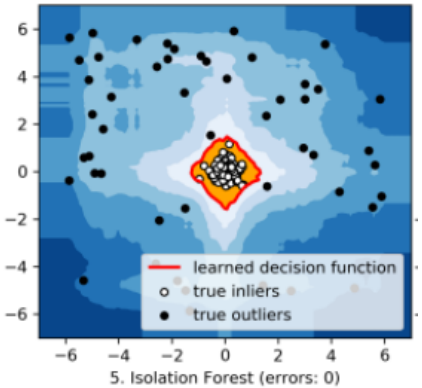
大致思路：利用决策树的特点，多次对特征空间进行划分，然后观察"孤立"（隔离）一个点难易程度。异常点往往在密度比较低的地方，更容易被隔离出来（决策树的深度比较低）。

调用方法：from pyod.models.iforest import IForest

优点：运行快、效果好，适用于大量数据，容易并行

缺点：需要调的参数比较多多，结果有一定的随机性

决策边界

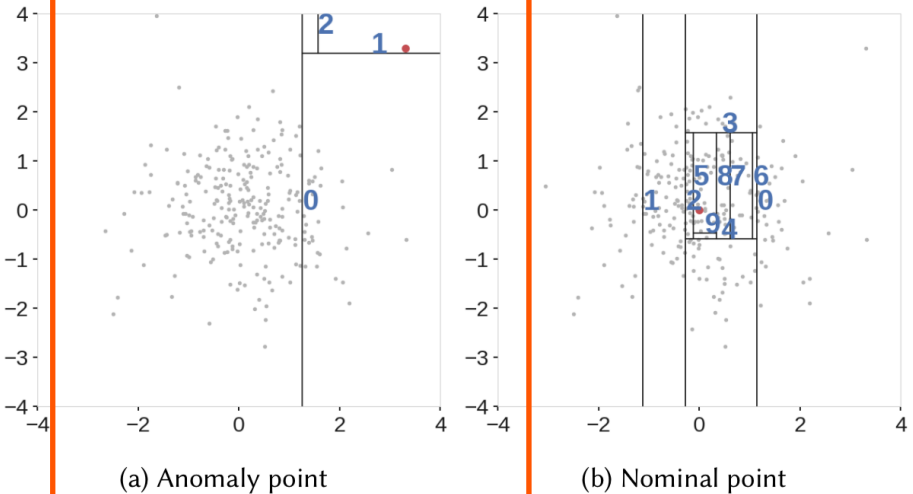


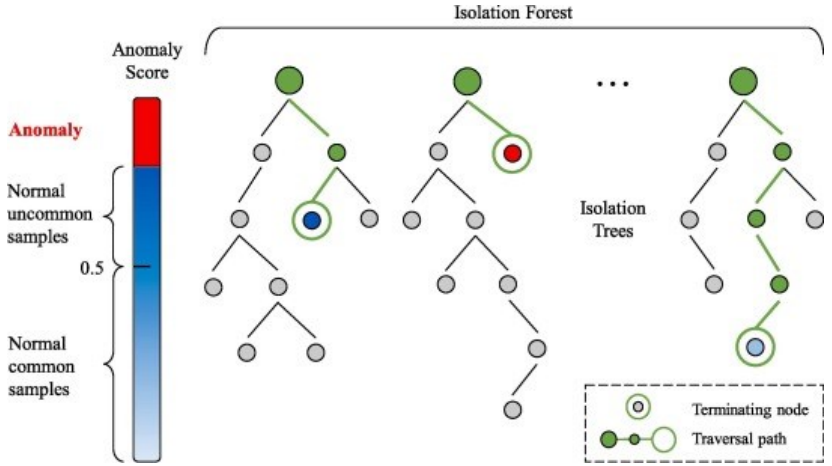
Carnegie Mellon University

Liu, F.T., Ting, K.M. and Zhou, Z.H., 2008, December. Isolation forest. In *International Conference on Data Mining (ICDM)*, pp. 413-422. IEEE.

异常检测方法5：孤立森林（Isolation Forest）

决策过程是不断对特征空间进行划分，异常点一般处于树的上面（离root更近）。





Carnegie Mellon University

Liu, F.T., Ting, K.M. and Zhou, Z.H., 2008, December. Isolation forest. In *International Conference on Data Mining (ICDM)*, pp. 413-422. IEEE.

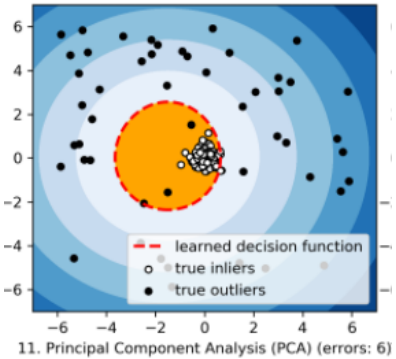
异常检测方法6：PCA

大致思路：对全部数据计算特征向量（eigenvectors），异常样本距离特征向量的距离比较远，因此这个距离可以被用作异常值。

调用方法：from pyod.models.pca import PCA

优点：直觉上简单，运算开销适中

缺点：作为一种线性模型，效果往往有一定的局限性



11. Principal Component Analysis (PCA) (errors: 6)

Shyu, M.L., Chen, S.C., Sarinnapakorn, K. and Chang, L., 2003. A novel anomaly detection scheme based on principal component classifier. MIAMI UNIV CORAL GABLES FL DEPT OF ELECTRICAL AND COMPUTER ENGINEERING.

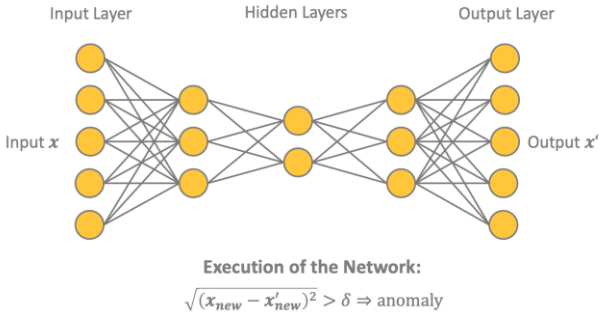
异常检测方法7：自编码器（Autoencoder）

大致思路：跟PCA相似，但扩展到了非线性的模型上。思路大概是通过自编码器的先压缩（encoder）再还原（decoder），异常点会有更大的重建误差（reconstruction error, $||X - X'||$）

调用方法：from pyod.models.auto_encoder import AutoEncoder

优点：理解简单，是PCA类型的模型的非线性扩展，数据量大、特征多时适用

缺点：神经网络模型往往有一定的随机性，运算开销比较大



Aggarwal, C.C., 2015. Outlier analysis. In Data mining (pp. 237-263). Springer, Cham.

Carnegie Mellon University

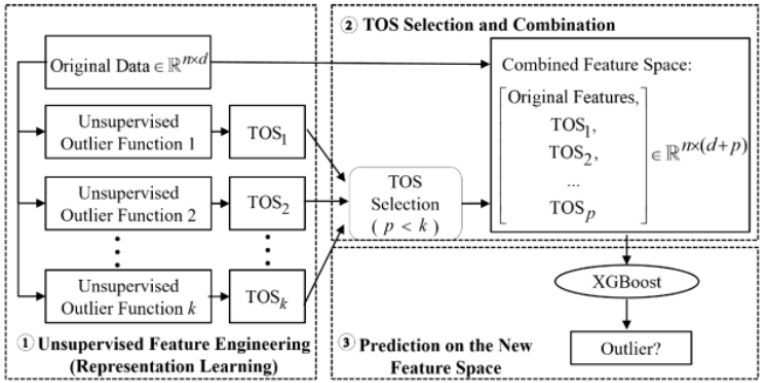
异常检测方法8（监督学习）：XGBOD（IJCNN‘18）

大致思路：如果数据有标签，那么是否可以结合xgboost与无监督的检测方法（把无监督的方法当做增强特征空间的方法）

调用方法：from pyod.models.xgbod import XGBOD

优点：可以同时利用数据中自带的标签，以及无监督学习带来的更好的表达

缺点：需要标签信息，运算开销比较大



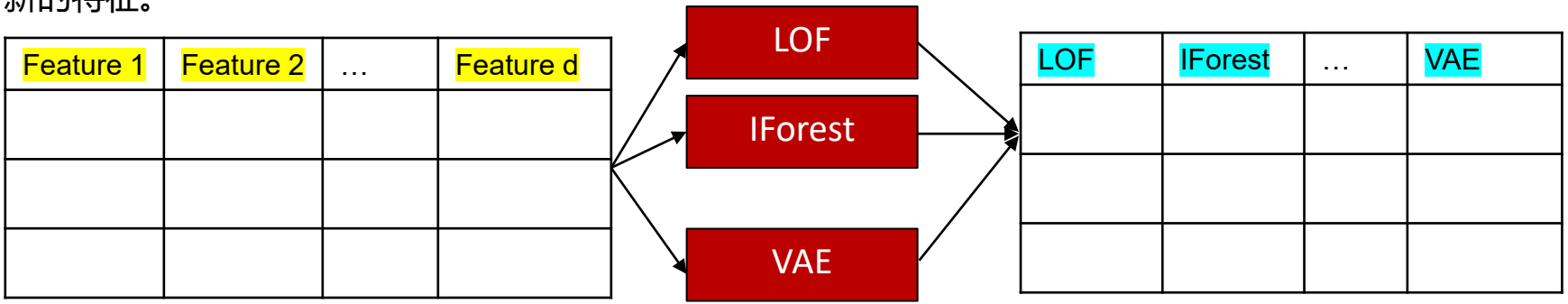
Zhao, Y. and Hryniewicki, M.K. XGBOD: Improving Supervised Outlier Detection with Unsupervised Representation Learning. *IEEE International Joint Conference on Neural Networks (IJCNN)*, 2018.

Carnegie Mellon University

异常检测方法8（监督学习）：XGBOD（IJCNN‘18）

Stacking

核心想法：机器学习的目标是得到好的数据表示，使得下游的分类任务变得容易。因此我们是否能从数据增强和提升的角度对原始数据集进行提升。一个简单的想法就是使用各种无监督学习的异常值来作为新的特征。



Feature 1	Feature 2	…	Feature d	LOF	IForest	…	VAE

Carnegie Mellon University

Zhao, Y. and Hryniewicki, M.K. XGBOD: Improving Supervised Outlier Detection with Unsupervised Representation Learning. *IEEE International Joint Conference on Neural Networks (IJCNN)*, 2018.

如何合并和选择模型?

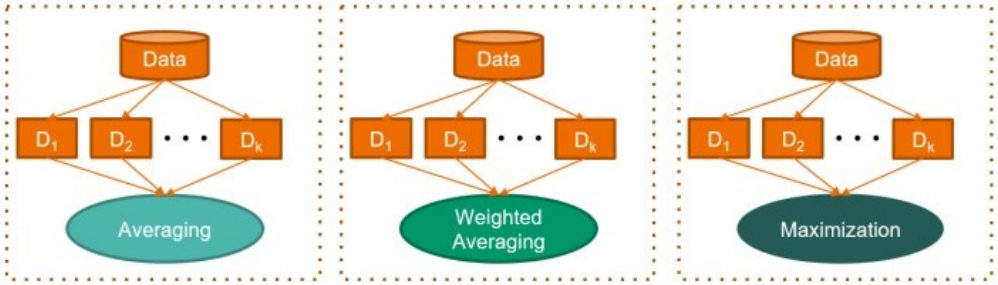
异常检测模型选择与合并：LSCP（SDM‘ 19）

背景：无监督异常检测往往需要训练多个异常检测模型，再去合并（比如做平均）

局限性：

(1) 不存在选择的过程（no selection process），因此只能得到平庸的结果

(2) 忽视了异常检测的局部性特征。**缺乏选择过程的全局性合并过程会使得合并的价值有所降低，得到稳定却中庸的结果。**



Examples of Parallel Detector Combination

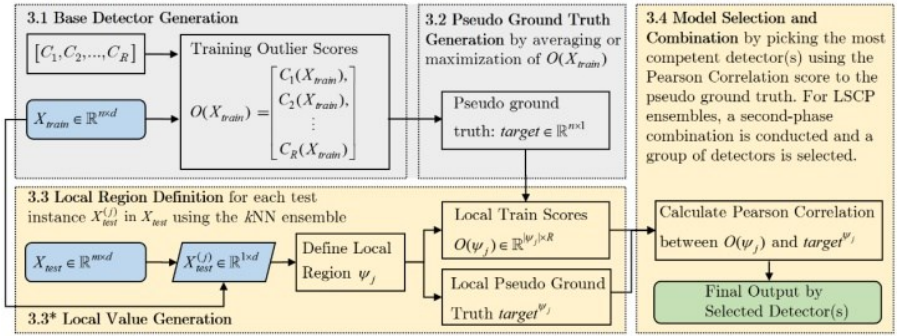
研究目标：提出一种新的合并框架，来“选择”适合于每个测试点的基础异常检测模型？

Zhao, Y., Nasrullah, Z., Hryniewicki, M.K. and Li, Z., 2019, May. LSCP: Locally selective combination in parallel outlier ensembles. *2019 SIAM International Conference on Data Mining (SDM)*, pp. 585-593. Society for Industrial and Applied Mathematics.

异常检测模型选择与合并：LSCP（SDM‘ 19）

解决方案：每当我们获得一个新的样本X进行预测时，先评估哪些检测模型在这个点***附近的区域***上表现良好（也就是在X的邻近区域上），那我们就更可以相信它会在X上表现良好。所以最简单的就是对于X，找到在它附近的训练数据上表现最好的模型 C_i，然后输出$C_i(X)$作为X的结果即可。

流程：（1）训练多个基础异常检测器；（2）生成伪标签用于评估；（3）对于每个测试点生成局部空间，也就是近邻（4）模型选择与合并，即对所有的基模型在我们找到的局部空间上用生产的伪标签进行评估，和伪标签在局部空间上Pearson大的被选做最终输出模型。



调用方法：from pyod.models.lscp import LSCP

Zhao, Y., Nasrullah, Z., Hryniewicki, M.K. and Li, Z., 2019, May. LSCP: Locally selective combination in parallel outlier ensembles. *2019 SIAM International Conference on Data Mining (SDM)*, pp. 585-593. Society for Industrial and Applied Mathematics.

异常检测模型选择：MetaOD（under review）

背景：拿到一个新的数据集想做异常检测，该怎么选择最适合的异常检测模型？

挑战：如何在**无监督的前提下**对任意数据集选择合适的异常检测模型

思路：把无监督的模型选择问题转化为「冷启动下的推荐问题」，给数据集选择模型就像给新用户推荐电影。因此我们可以借用推荐系统的方法，比如矩阵分解进行模型选择（推荐）。

训练与学习方法：需要首先得到大量（有标签）的数据在各种异常检测模型下的表现，称为模型表现矩阵P，之后对P进行矩阵分解。

	kNN	LOF	COPOD	IF
Data1	0.8	0.7	0.3	0.2
Data2	0.5	1	0.6	0.9
Data3	0.3	0.5	1	0.1

$P \in R^{n \times d}$

$\approx$

0.2	0.23
0.1	-1
0.3	0.1

$U \in R^{n \times k}$

$\cdot$

1	0.6	0.8	0.7
0.7	0.3	0.3	0.2

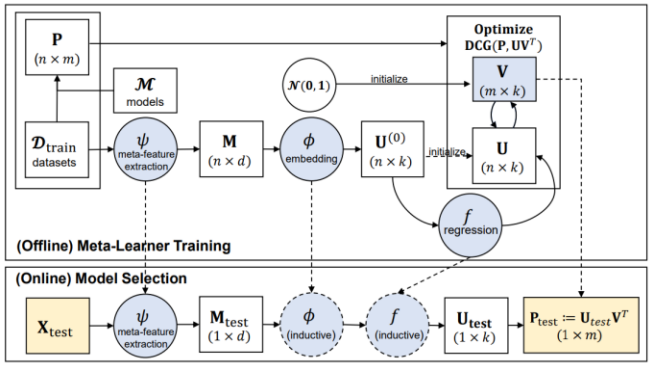
$V \in R^{k \times d}$

Carnegie Mellon University

Zhao, Y., Rossi, R.A. and Akoglu, L., 2020. Automating outlier detection via meta-learning. arXiv preprint arXiv:2009.10606.

异常检测模型选择：MetaOD（under review）

具体流程：（1）得到历史数据在不同模型上的表现矩阵P （2）对其进行分解成U和V矩阵（3）得到数据的表示（embedding），并训练一个回归模型f使得数据表示回归到矩阵U上（4）当拿到新数据时，先得到起数据表示X'，并使用f得到对应的U'，最终得到预测的模型表现$P' = \langle f(X'), V \rangle$



调用方法： pip install metaod; from metaod.models.predict_metaod import select_model

Zhao, Y., Rossi, R.A. and Akoglu, L., 2020. Automating outlier detection via meta-learning. arXiv preprint arXiv:2009.10606.

Carnegie Mellon University

异常检测训练加速：SUOD (under review)

背景：如果要训练多个无监督的异常检测模型，那么运算开销可能会给出大，尤其是在高维度、大数据上。如何高效的训练大量异常监测模型?

解决方案：在不同的层面上进行加速：

1. **数据层面**： 进行降维，SUOD中使用的 是Johnson-Lindenstrauss (JL) projection，可以在保证数据多样性的前提下降维
2. **模型层面**：无监督的密度估计开销很大，我们使用监督模型来学习（高开销的）无监督模型的训练结果。这个思想跟知识蒸馏有点像，即用小的神经网络去模拟大的神经网络输出。
3. **系统层面**：每个模型的开销各不相同，在做分布式学习时可能每个worker上的负载不同，导致先完成的worker需要等待还未完成的worker。为了解决这个问题，我们尝试去预测不同模型学习所需的时间，并根据预测值进行合理调度。

详细介绍： https://zhuanlan.zhihu.com/p/112588169

调用方法：pip install suod; from suod.models.base import SUOD

Yue Zhao, Xueying Ding, Jianing Yang, Haoping Bai, "Toward Scalable Unsupervised Outlier Detection".

Workshops at the Thirty-Fourth AAAI Conference on Artificial Intelligence, 2020.

异常检测实践中的一些技巧

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。

异常检测落地中的一些考量

1. **不要尝试一步到位用机器学习模型来代替传统模型**
2. 在理想情况下，应该尝试**合并机器学习模型和基于规则的模型**
3. **可以尝试用已有的规则模型去解释异常检测模型**

数据量、数据结构、可解释性、传统模型融合

异常检测的一些研究方向

模型选择与调参（model selection and hyperparameter tuning）

- 无监督学习，缺乏有效的评估与反馈
- 是否能使用“伪监督”(pseudo labels)的方法?
    - 比如把多个模型的结果取平均当伪标签

大规模的异常检测（large-scale anomaly detection）

- 并行化：现在缺少基于Spark的异常检测工具，在大数据上异常检测很难发力
- 如何应对evolving feature（特征是变化的）以及在线（online）检测covariate shift
- 模型压缩与优化：
    - 知识蒸馏：比如SUOD文章中尝试使用监督模型（随机森林）来替代无监督模型（kNN、LOF)
    - Quantization：降低精度

异常检测的一些研究方向

边缘计算（edge computing）：

- 移动端上的异常检测：如何在手机、智能手表上部署异常检测模型。主要还是依靠模型压缩
- 难点在于测试成本比较高，比如android端需要使用java来部署和simulate

FairOD：异常检测中的公平性

- 是否特定的特征（性别、年龄、种族等）会导致某一类群体更容易被认为是异常
- https://arxiv.org/abs/2012.03063

基于深度学习的异常检测工具库

- 越来越多的检测方法走向了深度学习，主要是基于GAN和VAE，但相关的工具库是缺失的
- 开发PyOD V2 for DL

其他研究思路：https://www.zhihu.com/question/324999831/answer/716118043

异常检测的相关资料

论文、教学、书籍：
https://github.com/yzhao062/anomaly-detection-resources

Table of Contents

- 1. Books & Tutorials
  - 1.1. Books
  - 1.2. Tutorials
- 2. Courses/Seminars/Videos
- 3. Toolbox & Datasets
  - 3.1. Multivariate data outlier detection
  - 3.2. Time series outlier detection
  - 3.3. Real-time Elasticsearch
  - 3.4. Datasets
- 4. Papers
  - 4.1. Overview & Survey Papers
  - 4.2. Key Algorithms
  - 4.3. Graph & Network Outlier Detection
  - 4.4. Time Series Outlier Detection
  - 4.5. Feature Selection in Outlier Detection
  - 4.6. High-dimensional & Subspace Outliers
  - 4.7. Outlier Ensembles
  - 4.8. Outlier Detection in Evolving Data
  - 4.9. Representation Learning in Outlier Detection
  - 4.10. Interpretability
  - 4.11. Outlier Detection with Neural Networks
  - 4.12. Active Anomaly Detection
  - 4.13. Interactive Outlier Detection
  - 4.14. Outlier Detection in Other fields
  - 4.15. Outlier Detection Applications
  - 4.16. Automated Outlier Detection
  - 4.17. Emerging and Interesting Topics
- 5. Key Conferences/Workshops/Journals
  - 5.1. Conferences & Workshops
  - 5.2. Journals

一键三连不迷路

Carnegie Mellon University