Anomaly Detection on Subway Operational Data

基于运营指标数据的异常分析模型

曹晨迪

caocd@ksu.edu

KANSAS STATE UNIVERSITY

Computer Science

卡斯柯 CASCO

Overview 简介

- Motivation 研究动机
- Introduction 相关介绍
- Experimental Method 实验方法
- Implementation 方法实施
- Result and Conclusion 结果和结论

Motivation 研究动机

- Anomaly analysis model based on subway operational indicator data.
  基于地铁运营指标数据的异常分析模型
  - Automatic detection and forecast. 自动识别和预测
  - Optimized the model based on real-time operational data. 根据实时运营数据优化模型
- Construct an outlier detection or anomaly detection model.
  建立离群点检测或是异常点检测模型
  - Model analysis of the relationship between delay time range, train, and destination.
    模型分析延误时间范围，列车，目的地等相关因素的相互关系

Introduction 相关介绍

- Knowledge discovery in databases, commonly referred to as data mining, is generating enormous interest in both the research and software arenas. [1]
  数据库中的知识发现，通常被称为数据挖掘，在研究和软件领域都产生了巨大的兴趣。[1]

- Python Outlier Detection (PyOD) is a comprehensive Python toolkit to identify outlying objects in multivariate data with both unsupervised and supervised approaches.[2] Python异常值检测（PYOD）是一种综合的Python工具包，用于在无监督和监督的方法中识别多变量数据中的离群对象。[2]

[1] 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).
[2] https://github.com/yzhao062/Pyod

Anomaly Detection Models[1] - I 异常检测模型

1. Linear Models for Outlier Detection: 离群点检测的线性模型
   I. **PCA**: Principal Component Analysis (use the sum of weighted projected distances to the eigenvector hyperplane as the outlier scores) (12)
   II. **MCD**: Minimum Covariance Determinant (use mahalanobis distances as outlier scores) (10)
   III. **One-Class Support Vector Machines** (11)
2. Proximity-Based Outlier Detection Models: 基于邻近度的离群点检测模型
   I. **LOF**: Local Outlier Factor (9)
   II. **CBLOF**: Clustering-Based Local Outlier Factor (2)
   III. **HBOS**: Histogram-based Outlier Score (4)
   IV. **kNN**: **k Nearest Neighbors** (use the distance to the kth nearest neighbor as outlier score) (6)
   V. **Average kNN or kNN Sum Outlier Detection** (use the average distance to k nearest neighbors as the outlier score or sum all k distances) (7)
   VI. **Median kNN Outlier Detection** (use the median distance to k nearest neighbors as the outlier score) (8)

[1] https://github.com/yzhao062/Pyod

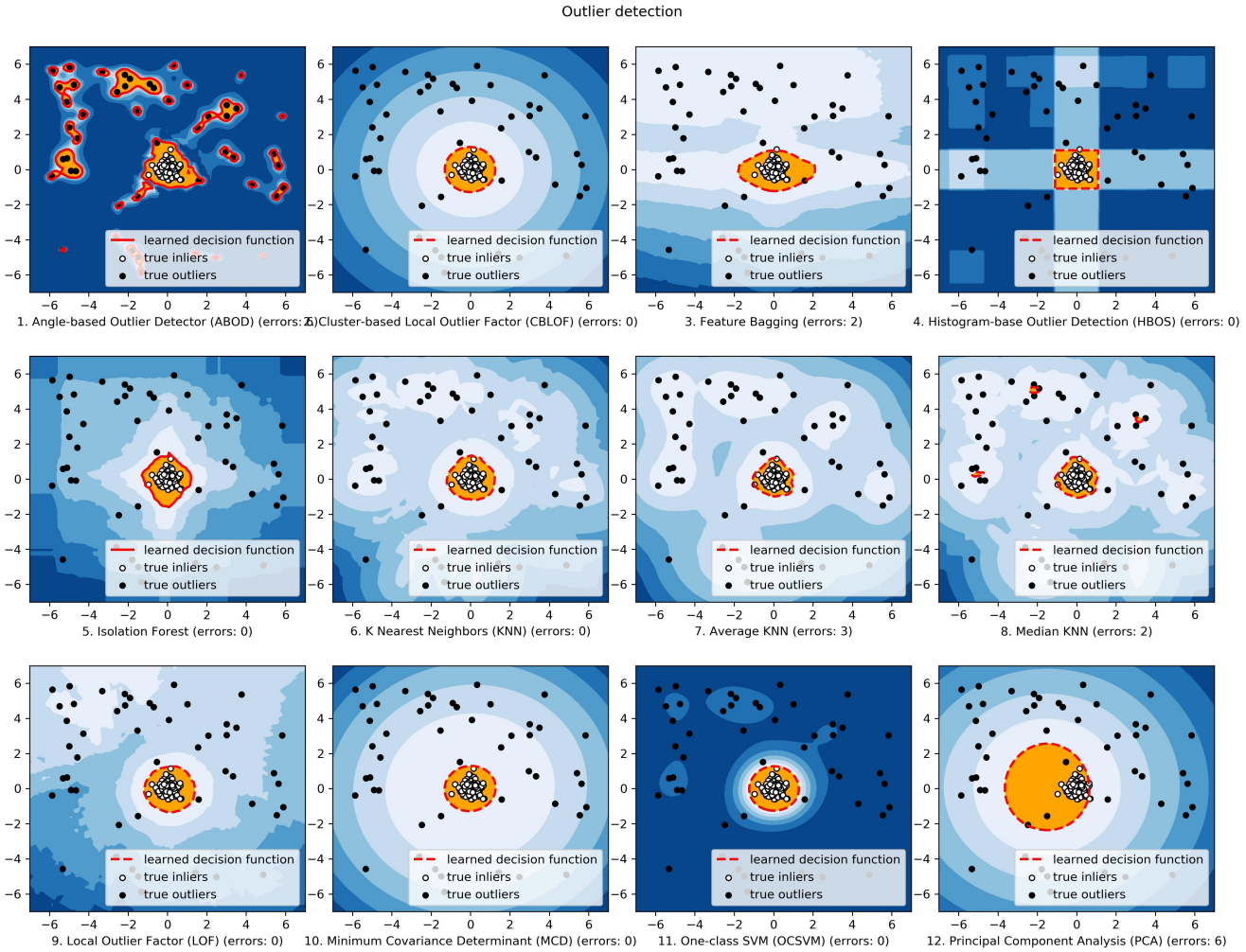
Anomaly Detection Models[1] - II 异常检测模型

3. Probabilistic Models for Outlier Detection: 异常值检测的概率模型
   I. **ABOD**: Angle-Based Outlier Detection (1)
4. Outlier Ensembles and Combination Frameworks: 离群集合与组合框架
   I. **Isolation Forest** (5)
   II. **Feature Bagging** (3)

[1] https://github.com/yzhao062/Pyod

Comparison of All Implemented Models

所有实现模型的比较



kNN: k Nearest Neighbor Algorithms K近邻算法

- The principle is to find a predefined number of training samples closest in distance to the new point and predict the label from these. The number of samples can be a user-defined constant (**k-nearest neighbor learning**).[1]
  原理是找到距离新点最近距离的预定义数量的训练样本，并从这些样本中预测标签。 样本数可以是用户定义的常数（k-最近邻居学习）。

- Neighbors-based methods are known as non-generalizing machine learning methods since they simply “remember” all of its training data.
  基于邻居的方法被称为非泛化机器学习方法，因为它们只是“记住”其所有训练数据。

[1] http://scikit-learn.org/stable/modules/neighbors.html

Implementation - Data Trimming 实施 - 数据修剪

- Subway operational metadata originally stored in Oracle database.
  地铁运营元数据最初存储在Oracle数据库中。
- Prior to doing data mining, we need to clean and trim the data.
  在进行数据挖掘之前，我们需要清理和修剪数据。

Sample output from Oracle database: Oracle数据库的示例输出

TRAIN_ID DESTINATION_CODE GROUP_TRAIN_ID LOCAL_SUB_ID GLOBAL_SUB_ID TRAIN_ATTRIBUTE STATION PLATFORM ARRIVAL_DEPARTURE_FLAG DATE_VALUE TIME_VALUE DATE_VALUE_EXPECTED TIME_VALUE_EXPECTED TIME_DIFF_FROM_SCHD

"003","314","0437","07","00306","16384","28","1","A","1530720000","45493","1530720000","45483","-10"
"020","384","0427","19","02018","16384","33","1","D","1530720000","45496","1530720000","45493","-3"
"004","014","0412","06","00406","16384","5","2","D","1530720000","45497","1530720000","45477","-20"

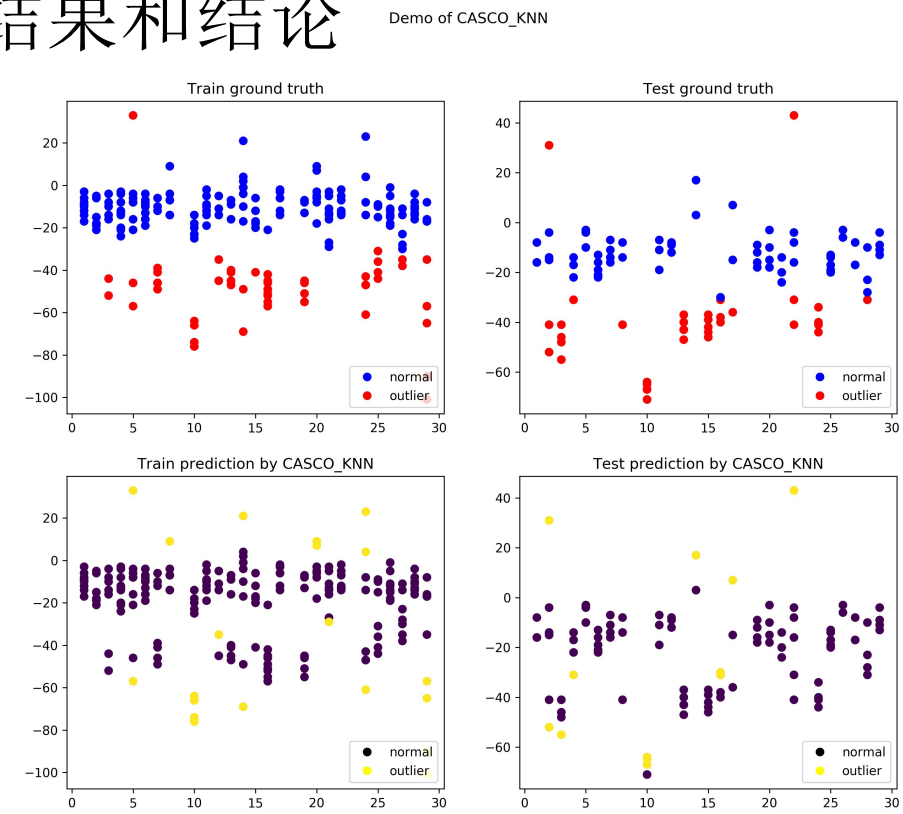
Implementation - Data Trimming 实施 - 数据修剪

- Convert the data format to be suitable for PyOD API.
  将数据转换为适合PyOD API的格式。
- Label the outlier in ground truth if the time is out of the range of ±30 seconds as scheduled. 如果时间超出计划的±30秒，则在基本事实中标注离群点。
- TRAIN_ID, TIME_DIFF_FROM_SCHD(s) and Labels (0: inliers, 1: outliers).
  列车ID，计划时间偏差（秒），和标签 （0：正常值，1：离群值）

29	-65	1	24	-41	1
25	-44	1	13	-43	1
12	-45	1	15	-44	1
.	.	.	.	.	.
25	-41	1	2	31	1
24	-47	1	28	-31	1
12	-45	1	10	-71	1

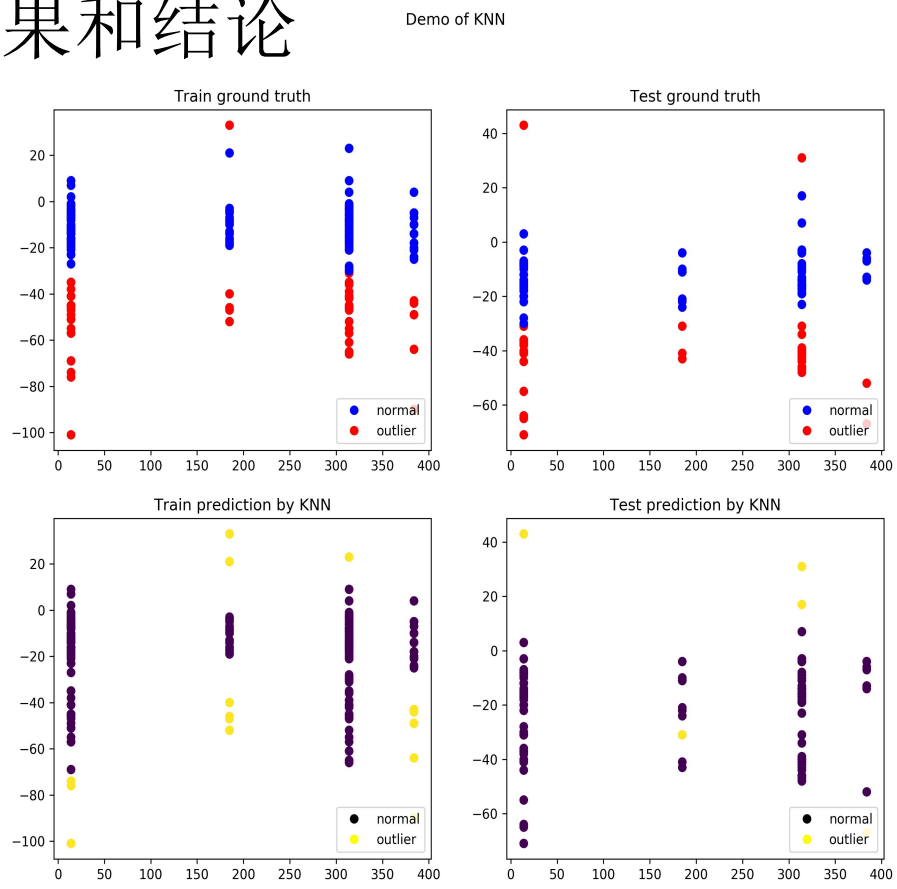
Result and Conclusion 结果和结论

- TRAIN_ID on X-axis ,TIME_DIFF_FROM_SCHD(s) on Y-axis.
  X轴：列车ID，Y轴：计划时间偏差（秒）
- Above graph shows the ground truth, and the below graph shows the prediction.
  上图显示了基本事实，下图显示了预测。



Result and Conclusion 结果和结论

- DESTINATION_CODE on X-axis ,TIME_DIFF_FROM_SCHD (s) on Y-axis
  X轴：目的地编码，Y轴：计划时间偏差（秒）



Thanks for your attention, any questions?

感谢您的关注，有什么问题吗？

All code and implementation has been uploaded on my GitHub: 所有代码已上传

https://github.com/paragon520/Pyod_CASCO

联系邮箱

Email: caocd@ksu.edu