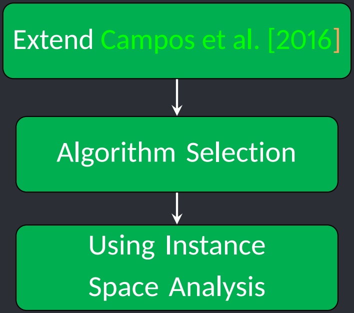
Instance space analysis for outlier detection

Sevvandi Kandanaarachchi, Mario A Muñoz, Kate Smith-Miles

Monash University & University of Melbourne

Overview

Guilherme O Campos, Arthur Zimek, Jörg Sander, Ricardo JGB Campello, Barbora Micenková, Erich Schubert, Ira Assent, and Michael E Houle. On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study. *Data Mining and Knowledge Discovery*, 30(4):891–927, 2016.



Outlier detection

- It means different things to different people
- What is the definition of an outlier?
- Hawkins : an observation that deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism
- We focus on ground truth when labelling outliers

Ground truth as an outlier

- Blue dot - a security breach, red dots - normal activity

Figure 1

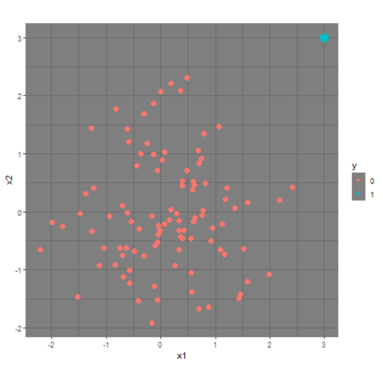
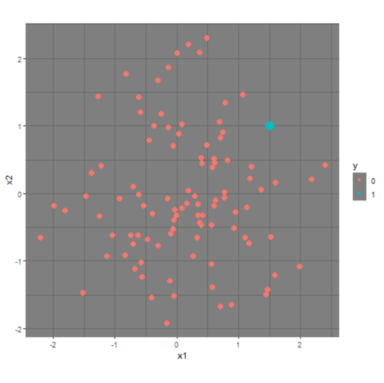


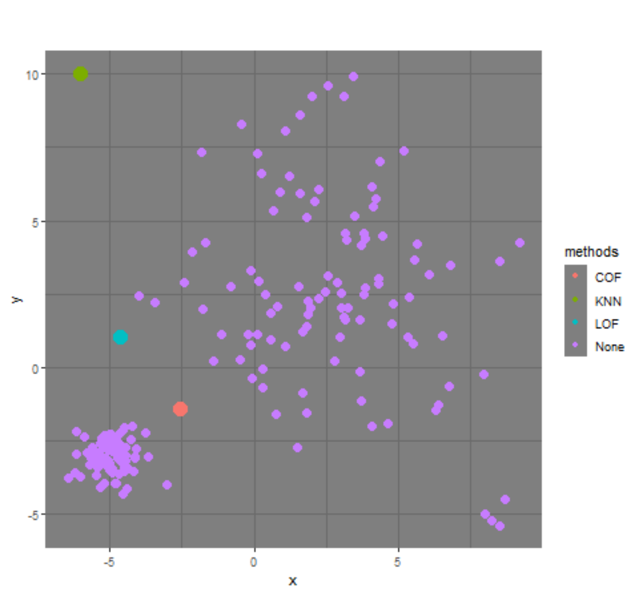
Figure 2



Basic Mechanism

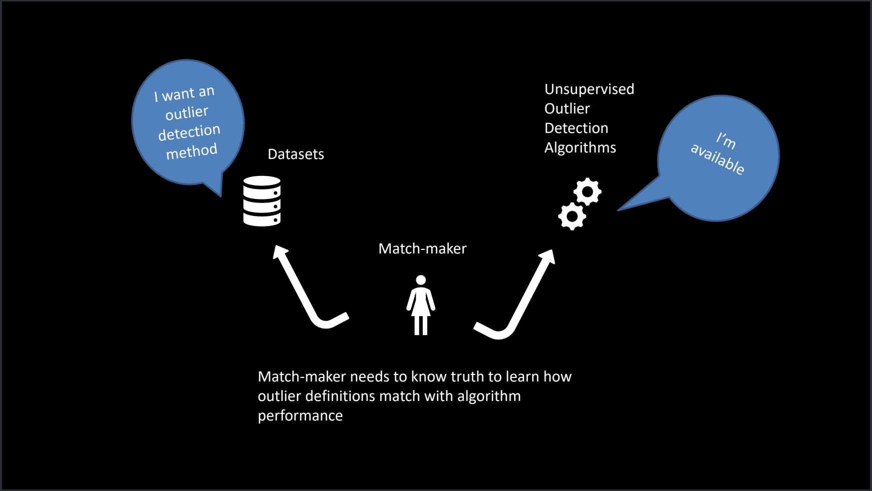
- data in $\mathbb{R}^n$
- A mapping $f : \mathbb{R}^n \longrightarrow \mathbb{R}$
- such that outliers have different $f(x)$ compared with other points

Due to different definitions and mechanisms

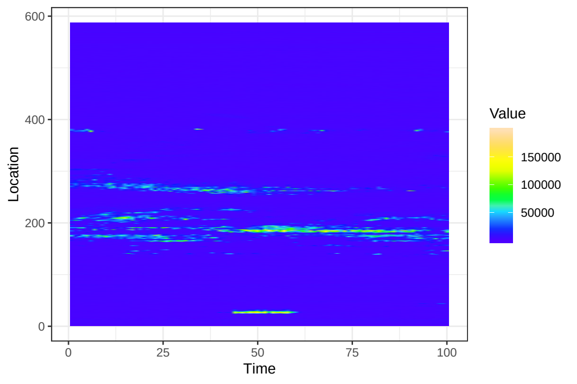


What is the best outlier detection method for my problem?

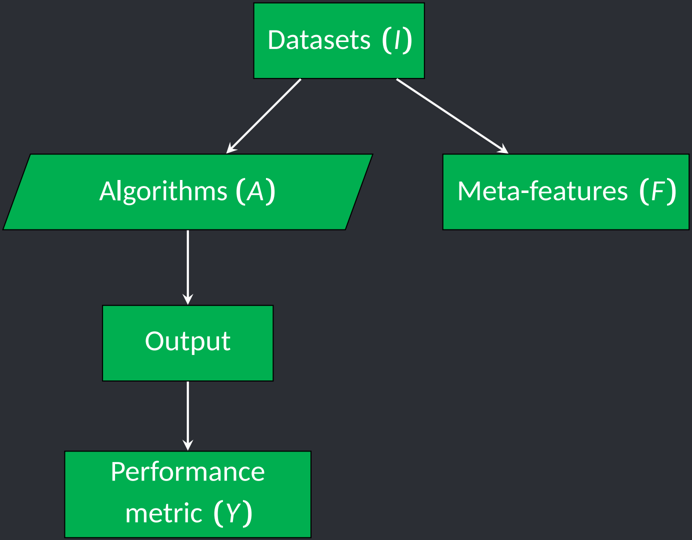
Algorithm Selection



To match we need to know the outlier labels



- Common in industry applications to have labeled data and develop methods to detect these outliers


Datasets (I)
Algorithms (A)
Meta-features (F)
Output
Performance metric (Y)

Our experiment with real data

- 12000+ datasets
  - Around 200 sources, many variants
- 12 outlier detection methods
- Meta-features
  - Describe a dataset
  - Each dataset can be represented as a feature vector
  - A way to compare datasets
  - 178 meta-features
- Matching outlier methods with datasets

- A lot of time on the Monash Cluster

Datasets = Instances = $I \in \{I, F, Y, A\}$

- Campos et al. datasets (only $\leq 5\%$ outliers)
- Goldstein-Uchida datasets from *A Comparative Evaluation of Unsupervised Anomaly Detection Algorithms for Multivariate Data*
- Muñoz et al. classification datasets from *Instance Spaces for Machine Learning Classification*
  - From UCI repository
  - Prepare for outlier detection
  - Downsampling each class - several variants
  - Convert categorical attributes numerical
  - Remove duplicate observations
  - Attend to missing values

Meta-features $= F \in \{I, F, Y, A\}$

- Simple features
  - Number of obs., attributes, binary attributes, ...
- Statistical features
  - skewness, kurtosis, mean to sd ...
- Information theoretic features
  - entropy, mutual information, ...
- Density based features
  - DBSCAN, Kernel density estimates ...
- Residual based features
- Graph based features

Breakdown of features

Feature category	Number of features
Generic : Simple, Statistical and Information theoretical	25
Density based	77
Residual based	35
Graph based	41
Total	178

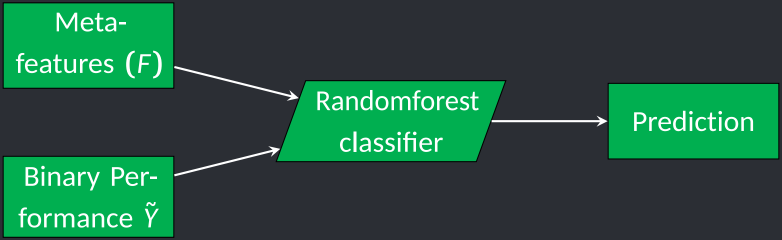
We use outlier labels to compute some features.

Outlier Detection Methods $= A \in \{I, F, Y, A\}$

- Distance based
  - KNN
  - KNNW
  - ODIN
- Density based
  - LOF
  - LDF
  - LDOF
  - LOOP
  - COF
  - SIMLOF
  - KDEOS
  - INFLO
- Angle based
  - FAST ABOD

Evaluation metric $= Y \in \{I, F, Y, A\}$

- Area under ROC, Precision-Recall curve, Precision@n
- We use area under ROC as the evaluation metric Y
- To validate meta-features
  - Define good performance as Area under ROC $> 0.8 \longrightarrow \tilde{Y}$
  - One model for each outlier detection method

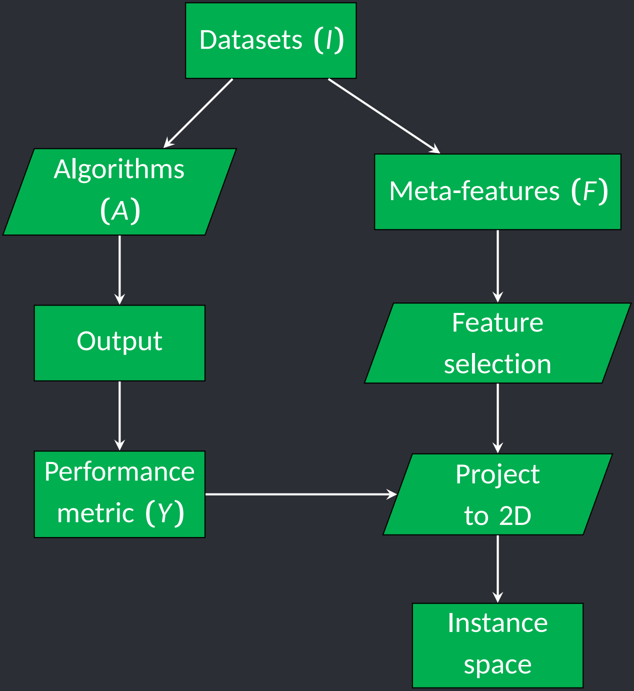


Validating the meta-features

Outlier detection method	Default accuracy(%)	Prediction accuracy(%)
COF	75.58	83.48
FAST ABOD	67.77	86.07
INFLO	83.22	89.29
KDEOS	90.96	92.81
KNN	68.16	86.56
KNNW	67.13	86.13
LDF	75.65	85.28
LDOF	80.08	87.36
LOF	74.63	84.07
LOOP	77.19	85.88
ODIN	79.09	87.00
SIMLOF	75.85	85.21

Understanding strengths and weaknesses of algorithms

- Instance space methodology
- Visually represent the datasets and the algorithm performances
- Understand the relative strengths and weaknesses


Datasets (I)
Algorithms (A)
Meta-features (F)
Output
Feature selection
Performance metric (Y)
Project to 2D
Instance space

Feature selection

- Select 7 features from 178
- Discard features with a small number of unique values
- Discard features that are highly correlated with each other and un-correlated with performance
- Cluster the remaining features
- Select the best combination of features

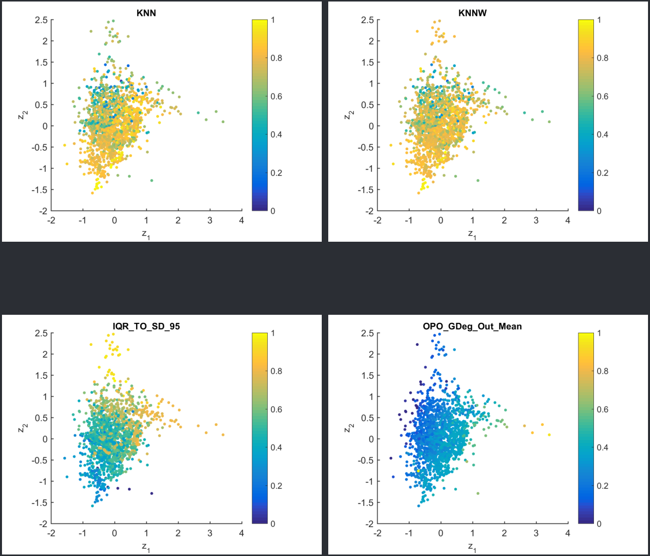
Chosen features

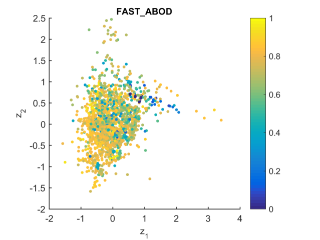
- Mean_Entropy_Attr (generic)
  - Mean entropy of attributes
- IQR_TO_SD_Ratio_95 (generic)
  - 95% of IQR to Standard deviation ratio of attributes
- OPO_Res_ResOut_Median (residual)
  - Median proxi-outlier residuals/ median non-PO residuals
- OPO_Res_Out_Mean (residual)
  - Mean of outlier residuals / mean of non-outlier residuals
- OPO_Den_Out_95P (density)
  - 95% of density of non-outliers / 95% of density of outliers
- OPO_GDeg_PO_Mean (graph)
  - Mean graph degree inner points/mean graph degree non-inner points
- OPO_GDeg_Out_Mean (graph)
  - Mean graph degree of outliers / mean graph degree of non-outliers

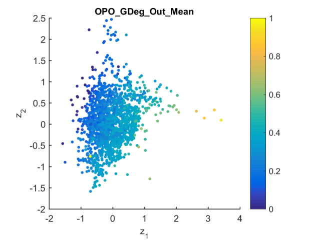
The projection

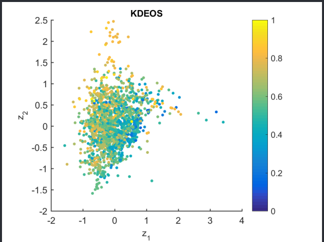
- PBLDR : Prediction Based Linear Dimensionality Reduction
- Finds a projection with most linear trends in algorithm performance and feature values

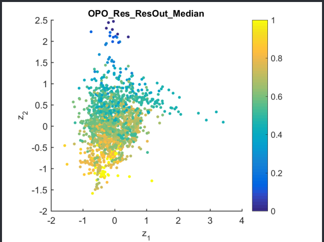
$$\mathbf{Z} = \begin{bmatrix} -0.0862 & -0.2078 \\ 0.1737 & 0.1845 \\ -0.0460 & -0.2847 \\ -0.0938 & -0.2025 \\ 0.1202 & 0.0378 \\ 0.1854 & -0.0822 \\ 0.3543 & -0.1325 \end{bmatrix}^{\top} \begin{bmatrix} \text{Mean\_Entropy\_Attr} \\ \text{IQR\_TO\_SD\_95} \\ \text{OPO\_Res\_ResOut\_Median} \\ \text{OPO\_Res\_Out\_Mean} \\ \text{OPO\_Den\_Out\_95P} \\ \text{OPO\_GDeg\_PO\_Mean} \\ \text{OPO\_GDeg\_Out\_Mean} \end{bmatrix}$$


KNN
KNNW
IQR_TO_SD_95
OPO_GDeg_Out_Mean
z_1
z_2
-2 -1 0 1 2 3 4
-2 -1.5 -1 -0.5 0 0.5 1 1.5 2 2.5
0 0.2 0.4 0.6 0.8 1


FAST_ABOD
z1
z2


OPO_GDeg_Out_Mean
z1
z2

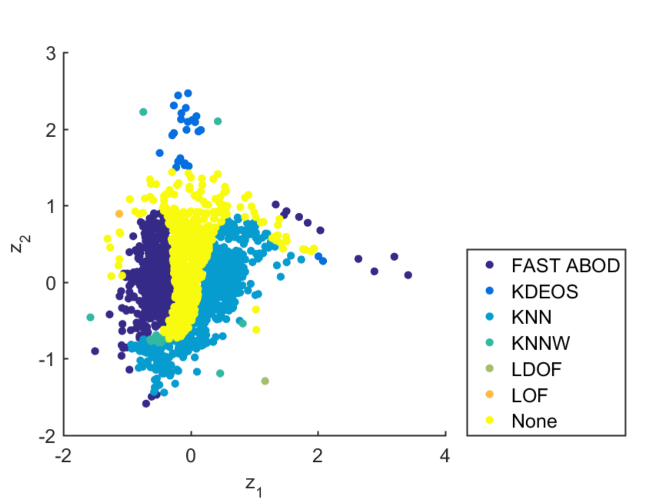

KDEOS
z1
z2


OPO_Res_ResOut_Median
z1
z2

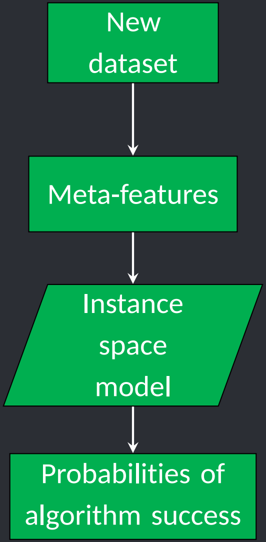
An SVM

- Partition the instance space
- Train an SVM for each outlier method
  - Input: instance coordinates in 2D
  - Binary output: For each method does the instance elicit good performance from the method
- Break ties using prediction probability of the SVM

Instance space



How to use it


New dataset
Meta-features
Instance space model
Probabilities of algorithm success

Takeaway

- No outlier method is superior to all other methods
- Need to find the suitable method for a given problem
- We find the strengths and weaknesses via instance space methodology and predict regions of good performance

Thank you!

- R package *outselect* at https://github.com/sevvandi/outselect
- Datasets at https://monash.figshare.com/articles/Datasets12338zip/7705127/4

THE UNIVERSITY OF MELBOURNE

MONASH University

ACEMS
AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE FOR MATHEMATICAL AND STATISTICAL FRONTIERS