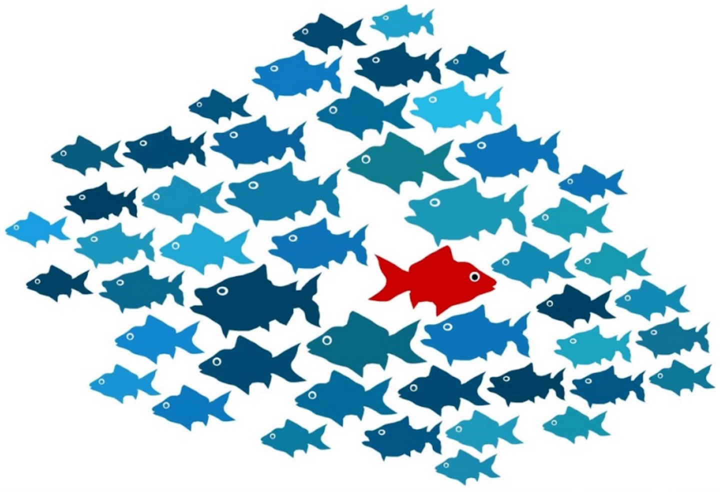
Anomaly Detection Part I

Advanced Research Topics – 7PAM2016

Dr Vanessa Graber (based on slides by Dr William Alston)

Learning outcomes

After the two lectures, you will:



- Understand anomaly detection problems.
- Understand the methods used for anomaly detection.
- Be able to identify, which algorithm to use for a particular problem set.
- Be able to implement this approach in Python.

Reading materials

Two PDFs can be found on Canvas

Anomaly Detection: A Survey

VARUN CHANDOLA, ARINDAM BANERJEE, and VIPIN KUMAR

University of Minnesota

Anomaly detection is an important problem that has been researched within diverse research areas and application domains. Many anomaly detection techniques have been specifically developed for certain application domains, while others are more generic. This survey tries to provide a structured and comprehensive overview of the research on anomaly detection. We have grouped existing techniques into different categories based on the underlying approach adopted by each technique. For each category we have identified key assumptions, which are used by the techniques to differentiate between normal and anomalous behavior. When applying a given technique to a particular domain, these assumptions can be used as guidelines to assess the effectiveness of the technique in that domain. For each category, we provide a basic anomaly detection technique, and then show how the different existing techniques in that category are variants of the basic technique. This template provides an easier and more succinct understanding of the techniques belonging to each category. Further, for each category, we identify the advantages and disadvantages of the techniques in that category. We also provide a discussion on the computational complexity of the techniques since it is an important issue in real application domains. We hope that this survey will provide a better understanding of the different directions in which research has been done on this topic, and how techniques developed in one area can be applied in domains for which they were not intended to begin with.

15

Revisiting Time Series Outlier Detection: Definitions and Benchmarks

Kwei-Herng Lai
Rice University
khlai@rice.edu

Daochen Zha
Rice University
daochen.zha@rice.edu

Junjie Xu
Penn State University
jmx5097@psu.edu

Yue Zhao
Carnegie Mellon University
zhaoy@cmu.edu

Guanchu Wang
Rice University
hegsns@rice.edu

Xia Hu
Rice University
xiahu@rice.edu

Abstract

Time series outlier detection has been extensively studied with many advanced algorithms proposed in the past decade. Despite these efforts, very few studies have investigated how we should benchmark the existing algorithms. In particular, using synthetic datasets for evaluation has become a common practice in the literature, and thus it is crucial to have a general synthetic criterion to benchmark algorithms. This is a non-trivial task because the existing synthetic methods are very different in different applications and the outlier definitions are often ambiguous. To bridge this gap, we propose a behavior-driven taxonomy for time series outliers and categorize outliers into point- and pattern-wise outliers with clear context definitions. Following the new taxonomy, we then present a general synthetic criterion and generate 35 synthetic datasets accordingly. We further identify 4 multivariate real-world datasets from different domains and benchmark 9 algorithms on the synthetic and the real-world datasets. Surprisingly, we observe that some classical algorithms could outperform many recent deep learning approaches. The datasets, preprocessing and synthetic scripts, and the algorithm implementations are made publicly available at https://github.com/datamllab/tods/tree/benchmark.

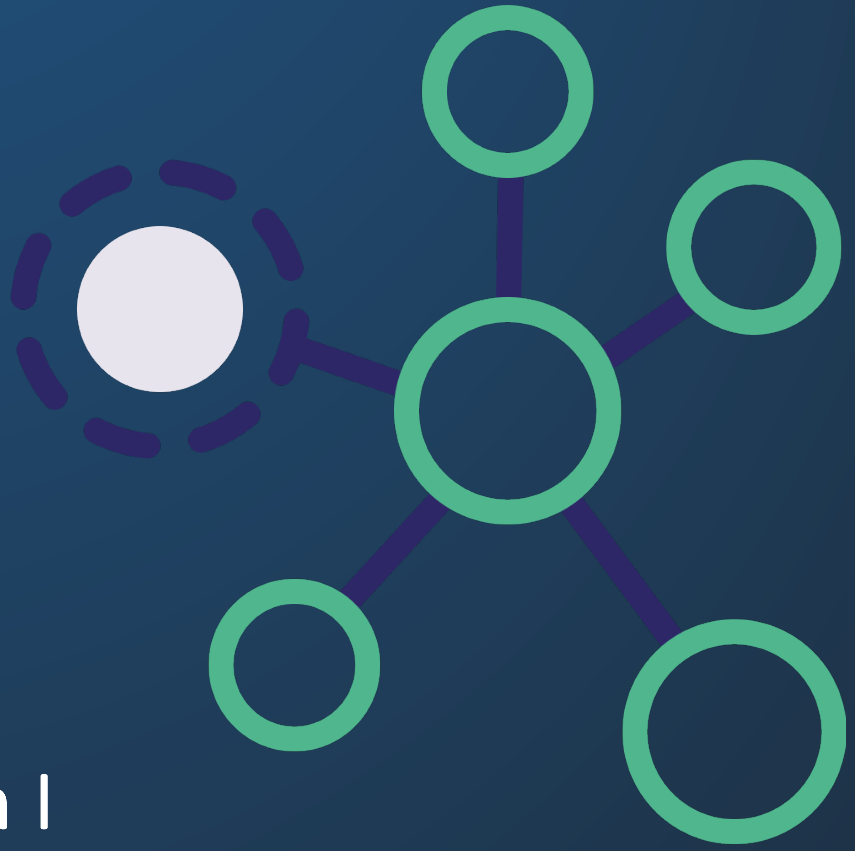
Introduction

Key questions

Applications

Techniques for anomaly detection I

Summary



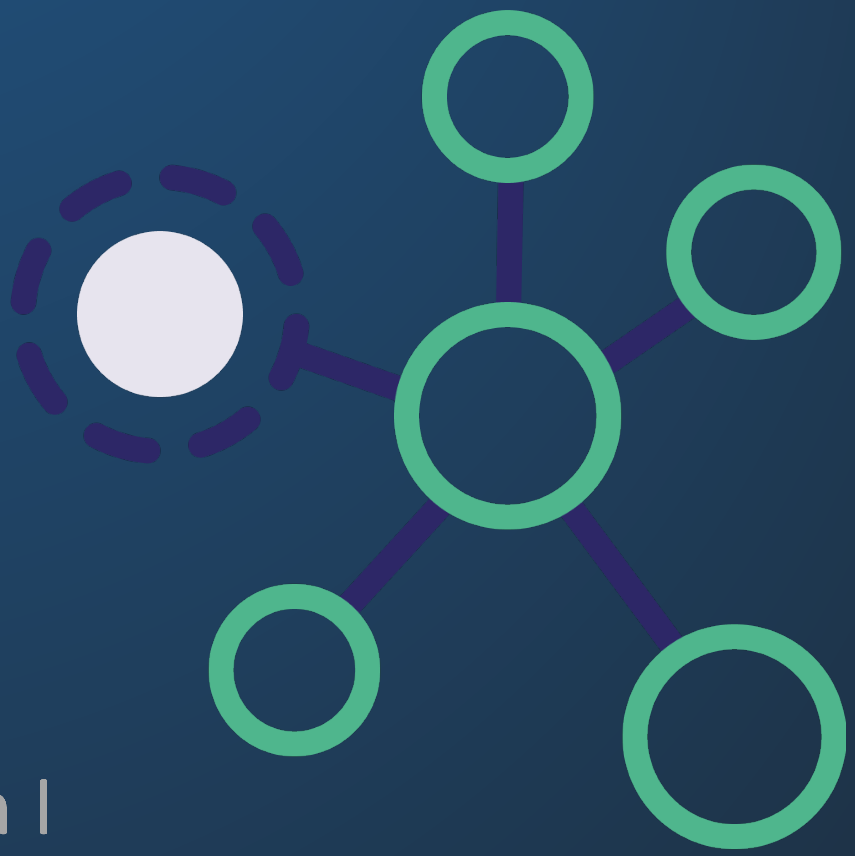
Introduction

Key questions

Applications

Techniques for anomaly detection I

Summary



Anomalies

And what they mean

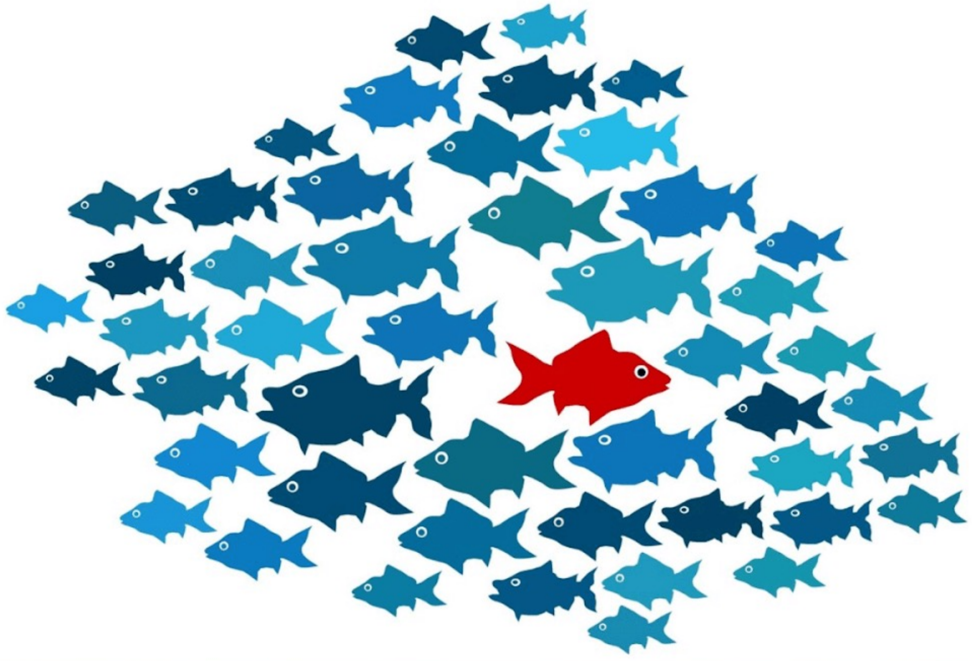
- "We are drowning in information, while starving for wisdom." Consilience: The Unity of Knowledge (1998), biologist E. O. Wilson
- Anomalous events occur relatively infrequently.
- However, when they do occur, their **consequences** can be **dramatic** and often **negative**.

Anomalies can be like needles in a haystack.

Anomaly detection

"The odd one out"

- An anomaly is a pattern in the data that does **not conform with the expected behaviour**. They are also referred to as outliers, exceptions, peculiarities, surprises, novelties, incongruences, etc.
- Historically, the **field of statistics** dealt with anomalies to find and remove outliers to improve analyses. There are now many fields, where **anomalies are the topic** of greatest **interest**.



To detect them, we need to identify objects, events, etc. that are different from most other objects, events, etc.

Real-world anomalies
and related concepts

- Anomalies translate to significant (often critical) real life events:
  - Credit card fraud: an abnormally high purchase on a credit card
  - Cyber intrusion: webserver involved in FTP (File Transfer Protocol) traffic

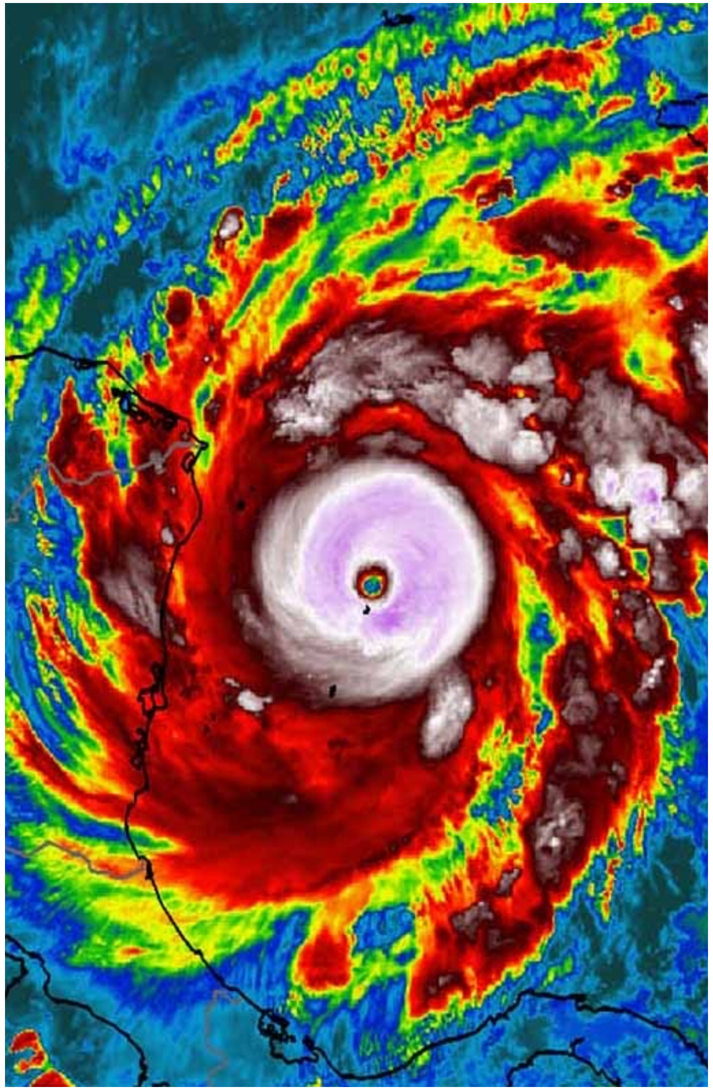
Anomaly detection is related to concepts like rare class mining, chance discovery, novelty detection, exception mining, noise removal and black swan events*

* Theory developed by Nassim N. Taleb starting in 2001.

Causes of anomalies

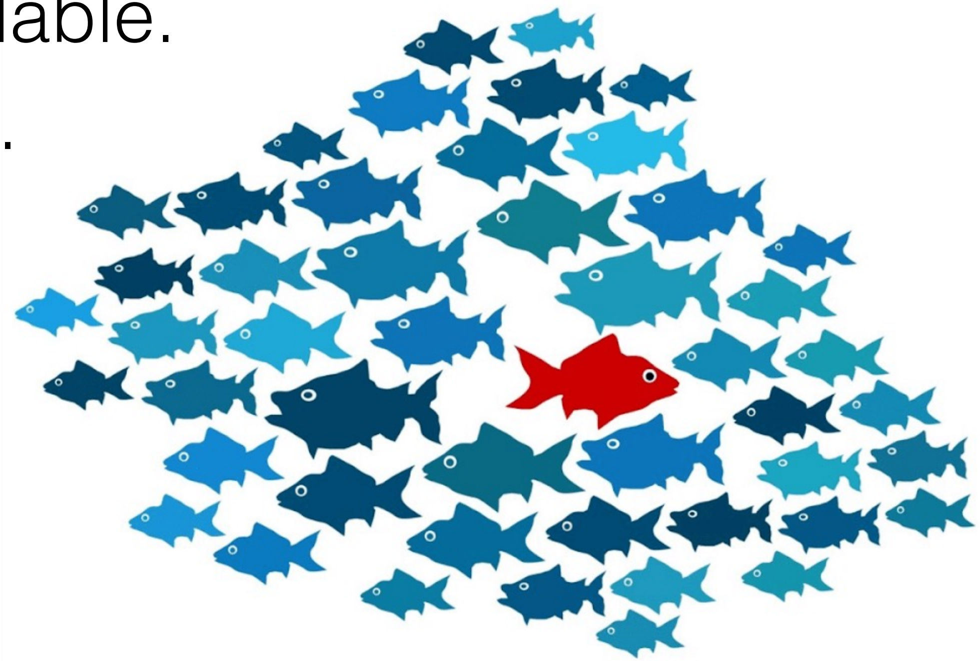
Several primary causes

- Some data elements are part of a different class of objects or produced by a different underlying mechanism (e.g., disease vs. no disease or fraud vs. no fraud).
- Data elements can originate from the tails of an underlying Gaussian distribution.
- The underlying process has a natural variation (e.g., extreme weather events).
- The underlying data was measured, and collection errors were made (e.g., human, equipment).



Key challenges in anomaly detection

- Defining a representative normal region is challenging.
- Boundary between normal & outlying behaviour is often imprecise.
- Exact notion of an outlier varies between application domains.
- Data labels for training/validation unavailable.
- Malicious adversaries are unpredictable.
- Normal behaviour evolves with time.
- Data might contain noise.
- Selection of relevant features is difficult.



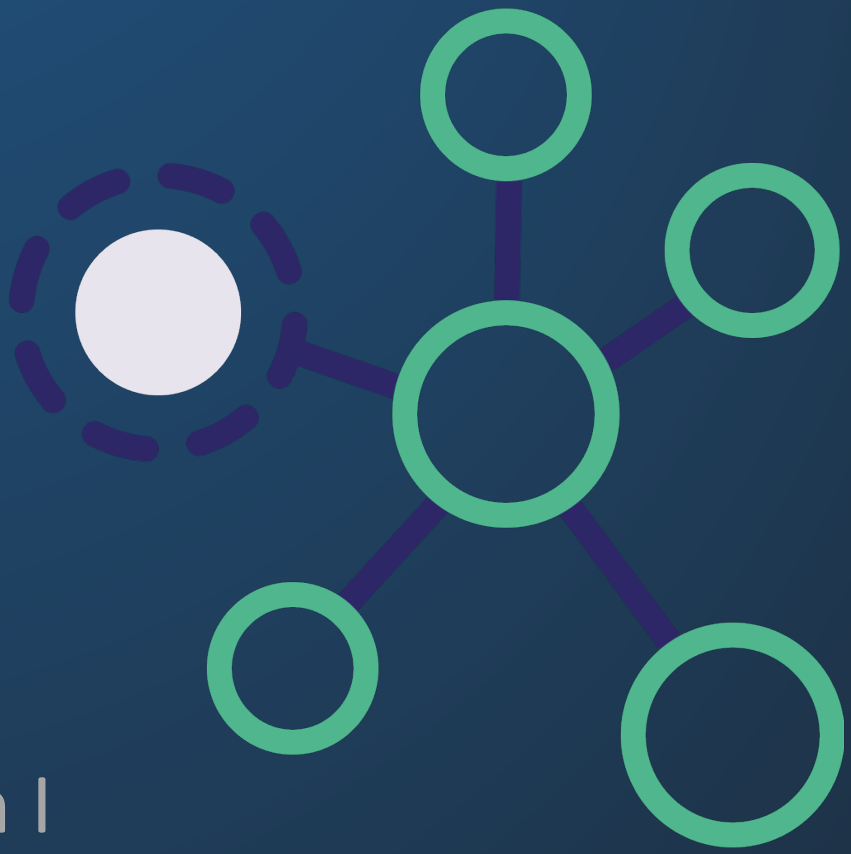
Introduction

Key questions

Applications

Techniques for anomaly detection I

Summary



Key questions in anomaly detection

We will answer these in our lectures.

- What is the nature of the **input data**?
- Can we perform **supervised learning**, i.e., is labelled training data available?
- What **type of anomaly** are we trying to detect?
  - We will distinguish point, contextual, and structural anomalies.
- What does the **output of anomaly detection** look like?
- How can we **evaluate anomaly detection** techniques?

Input data: types

- Most common data form handled by anomaly detection techniques is **record data**. We distinguish **univariate and multivariate data**.

Univariate data

Engine temperature (°C)
88
75
81
95
10
90
89
72

Multivariate data

Id	Src IP	Start time	Dest IP	Dest port	Number of bytes	Attack
1	206.135.38.95	11:07:20	160.94.179.223	139	192	No
2	206.163.37.95	11:13:56	160.94.179.219	139	195	No
3	206.163.37.95	11:14:29	160.94.179.217	139	180	No
4	206.163.37.95	11:14:30	160.94.179.255	139	199	No
5	206.163.37.95	11:14:32	160.94.179.254	139	19	Yes
6	206.163.37.95	11:14:35	160.94.179.253	139	177	No
7	206.163.37.95	11:14:36	160.94.179.252	139	172	No
8	206.163.37.95	11:14:44	160.94.179.249	139	163	Yes

Input data: nature of attributes

- We can generally distinguish the following **kinds of attributes**:
  - Binary
  - Categorical
  - Continuous
  - Hybrid

What kinds of attributes can you see in our multivariate data example from earlier?

Id	Src IP	Start time	Dest IP	Dest port	Number of bytes	Attack
1	206.135.38.95	11:07:20	160.94.179.223	139	192	No
2	206.163.37.95	11:13:56	160.94.179.219	139	195	No
3	206.163.37.95	11:14:29	160.94.179.217	139	180	No
4	206.163.37.95	11:14:30	160.94.179.255	139	199	No
5	206.163.37.95	11:14:32	160.94.179.254	139	19	Yes

Input data: nature of attributes

- We can generally distinguish the following **kinds of attributes**:
  - Binary
  - Categorical
  - Continuous
  - Hybrid

What kinds of attributes can you see in our multivariate data example from earlier?

binary

Id	Src IP	Start time	Dest IP	Dest port	Number of bytes	Attack
1	206.135.38.95	11:07:20	160.94.179.223	139	192	No
2	206.163.37.95	11:13:56	160.94.179.219	139	195	No
3	206.163.37.95	11:14:29	160.94.179.217	139	180	No
4	206.163.37.95	11:14:30	160.94.179.255	139	199	No
5	206.163.37.95	11:14:32	160.94.179.254	139	19	Yes

Input data: nature of attributes

- We can generally distinguish the following **kinds of attributes**:
  - Binary
  - Categorical
  - Continuous
  - Hybrid

What kinds of attributes can you see in our multivariate data example from earlier?

continuous

binary

Id	Src IP	Start time	Dest IP	Dest port	Number of bytes	Attack
1	206.135.38.95	11:07:20	160.94.179.223	139	192	No
2	206.163.37.95	11:13:56	160.94.179.219	139	195	No
3	206.163.37.95	11:14:29	160.94.179.217	139	180	No
4	206.163.37.95	11:14:30	160.94.179.255	139	199	No
5	206.163.37.95	11:14:32	160.94.179.254	139	19	Yes

Input data: nature of attributes

- We can generally distinguish the following **kinds of attributes**:
  - Binary
  - Categorical
  - Continuous
  - Hybrid

	categorical		categorical	categorical	continuous	binary
Id	**Src IP**	**Start time**	**Dest IP**	**Dest port**	**Number of bytes**	**Attack**
1	206.135.38.95	11:07:20	160.94.179.223	139	192	No
2	206.163.37.95	11:13:56	160.94.179.219	139	195	No
3	206.163.37.95	11:14:29	160.94.179.217	139	180	No
4	206.163.37.95	11:14:30	160.94.179.255	139	199	No
5	206.163.37.95	11:14:32	160.94.179.254	139	19	Yes

What kinds of attributes can you see in our multivariate data example from earlier?

Input data: nature of attributes

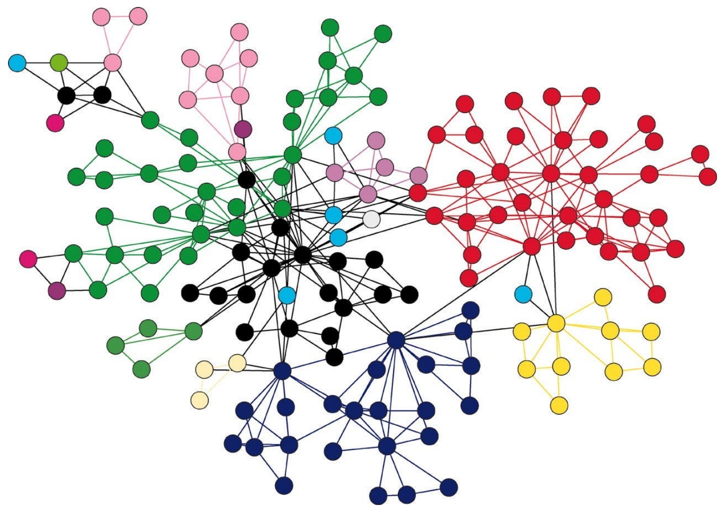
- We can generally distinguish the following **kinds of attributes**:
  - Binary
  - Categorical
  - Continuous
  - Hybrid

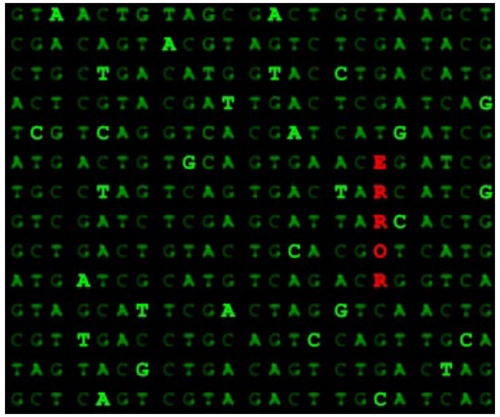
What kinds of attributes can you see in our multivariate data example from earlier?

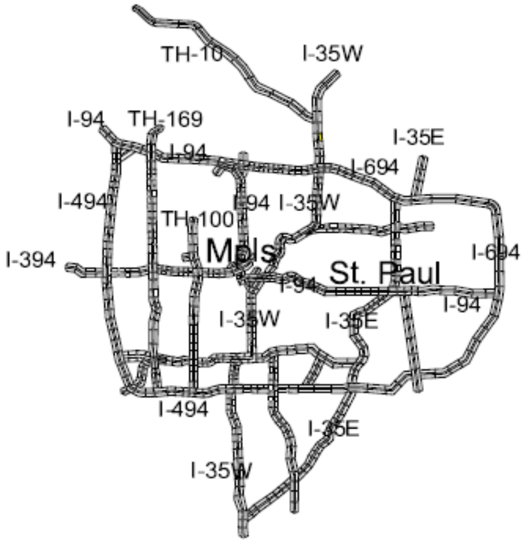
Id	Src IP (categorical)	Start time (categorical, continuous)	Dest IP (categorical)	Dest port (categorical)	Number of bytes (continuous)	Attack (binary)
1	206.135.38.95	11:07:20	160.94.179.223	139	192	No
2	206.163.37.95	11:13:56	160.94.179.219	139	195	No
3	206.163.37.95	11:14:29	160.94.179.217	139	180	No
4	206.163.37.95	11:14:30	160.94.179.255	139	199	No
5	206.163.37.95	11:14:32	160.94.179.254	139	19	Yes

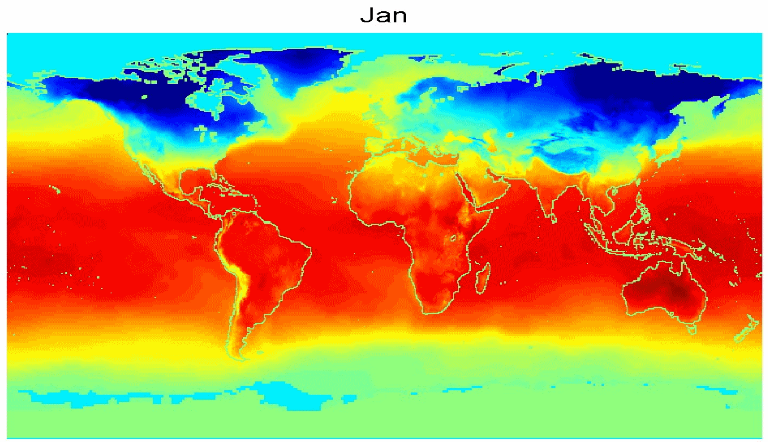
Input data: relationships

- Data instances can have relationships:
  - Sequential (e.g., temporal)
  - Spatial
  - Spatio-temporal
  - Graph









Data labels

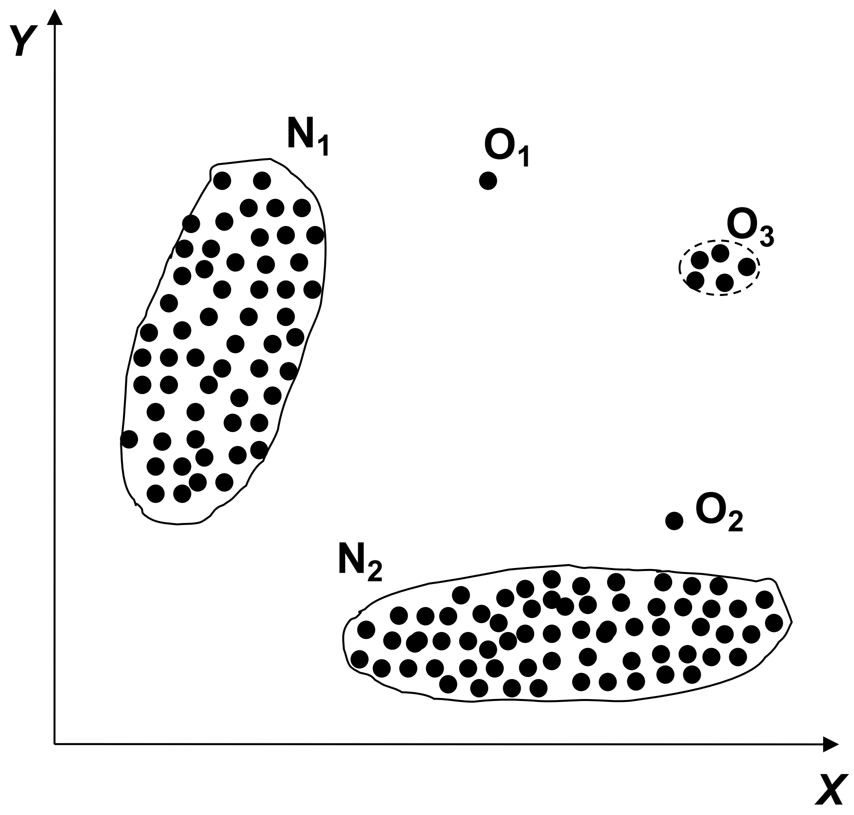
A distinction between three cases

- Supervised anomaly detection:
  - Labels are available for both normal data and anomalies.
  - Similar to rare class mining.
- Semi-supervised anomaly detection (**novelty detection**):
  - Labels are available only for normal data, but not for anomalies.
- Unsupervised anomaly detection (**outlier detection**):
  - No labels are assumed.
  - Assumption: anomalies rare compared to normal data.
  - Understand “normal” behaviour (e.g., summary statistics).

Novelty detection: The training data is not polluted by outliers. We want to detect whether a new observation is an outlier. In this context, an outlier is called a novelty.

Outlier detection: The training data contains outliers (observations far from others). Algorithms try to fit the regions of concentrated training data, ignoring the deviating observations.

Types of anomalies
Point anomalies

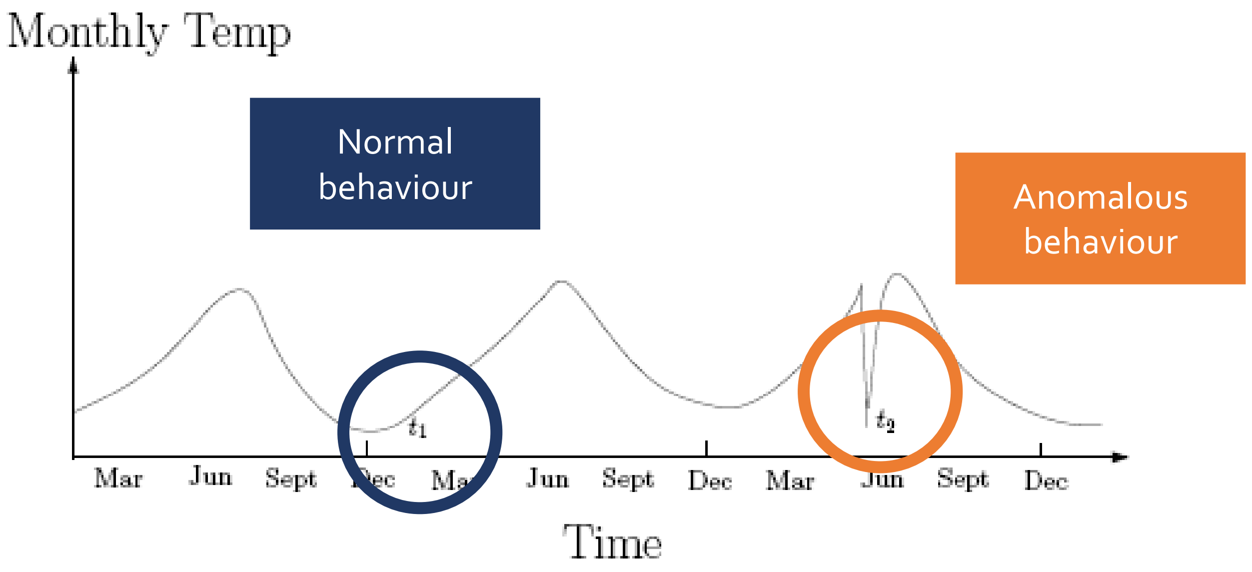


- **Point anomaly**: An individual data instance is anomalous with respect to the remaining data in a dataset.

- In the example on the left, we have:
  - N_1 and N_2 are regions of normal behaviour.
  - Points O_1 and O_2 are anomalies w.r.t. N_1 and N_2.
  - Points in region O_3 are also anomalies.

Types of anomalies
Contextual anomalies

- **Contextual anomaly**: An individual data instance is anomalous within a specific context. This requires a **notion of context**.
- Such anomalies are also referred to as **conditional anomalies** (see e.g., Song et al., Conditional Anomaly Detection, IEEE,19, 5, 2006).

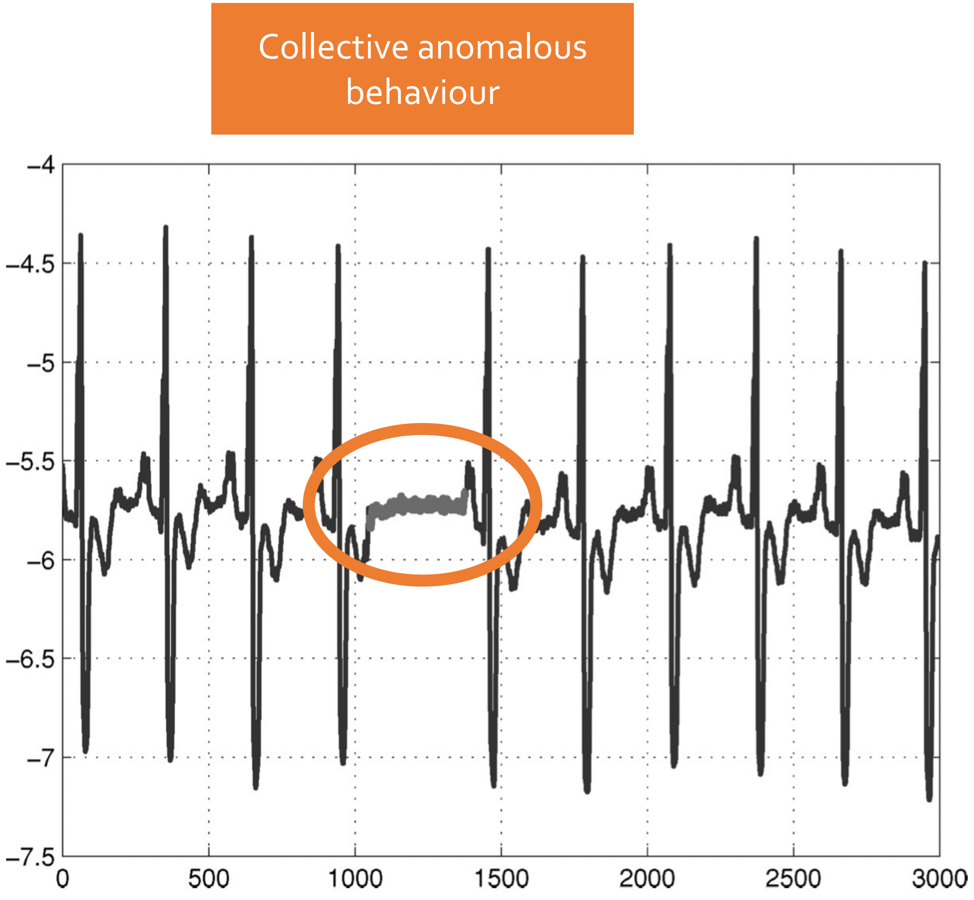


Types of anomalies

Collective anomalies

- **Collective anomaly**: A collection of related data instances is anomalous.
- This requires a (sequential, spatial, or graph) relationship among data instances.

The individual instances within a collective anomaly are not anomalous by themselves.

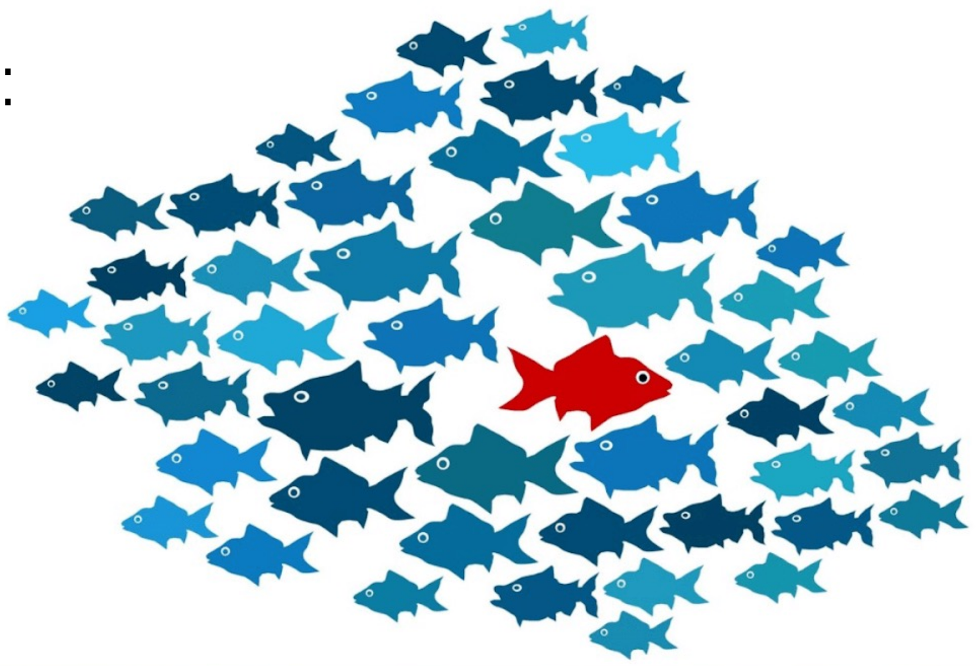


Output of anomaly detection

Two different types

- Anomaly detection can output a **label**:
  - Each test instance is given a “normal” or “anomaly” label.
  - This is typically used for classification-based approaches.

- Anomaly detection can output a **score**:
  - Each test instance is assigned an anomaly score.
  - This allows the output to be ranked.
  - The approach however requires an additional threshold parameter.



Evaluation of anomaly detection

F-scores

- **Accuracy** is an insufficient metric for anomaly detection.
  - Consider, for example, a network traffic dataset with 99.9% of normal data and 0.1% of intrusions. A trivial classifier that labels everything as normal and detects no anomalies can achieve 99.9% accuracy!!

- Consider instead **f-score** (combination of precision and recall):

True positives (TP)	False positives (FP)
False negatives (FN)	True negatives (TN)

$$F_\beta = (1+\beta^2) * \frac{PR * RC}{\beta^2 * PR + RC} = \frac{(1+\beta^2) * TP}{(1+\beta^2) * TP + \beta^2 * FN + FP}$$

$$PR = \frac{TP}{TP+FP}, \qquad RC = \frac{TP}{TP+FN}$$

Evaluation of anomaly detection

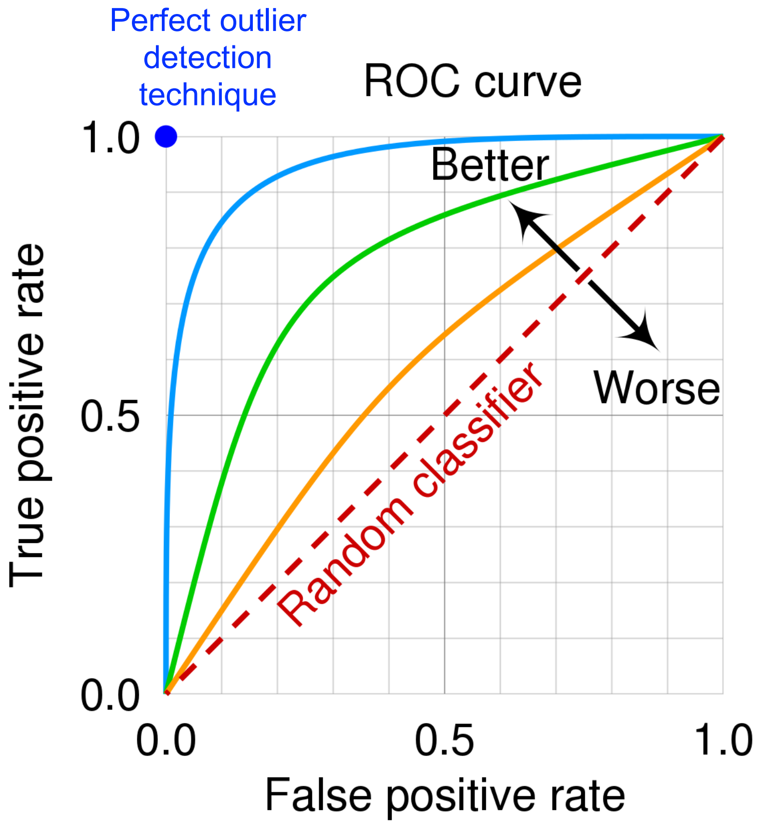
ROC (receiver operating character.) curve, AUC (area under ROC curve)

- Alternatively, we can use ROC curves and AUC. We obtain the former by plotting the **true positive rate** (TPR; also called recall or sensitivity) against the **false positive rate** (FPR):
  - TPR - ratio between correctly detected anomalies and total anomalies (TP + FP).
  - FPR - ratio between false positive count (normal instances misclassified as anomalies) and the number of ground truth negatives (FP + TN).

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}},$$

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}}$$

ROC curve is a trade-off between detection rate and false alarm rate. AUC can be computed via trapezoid rule.



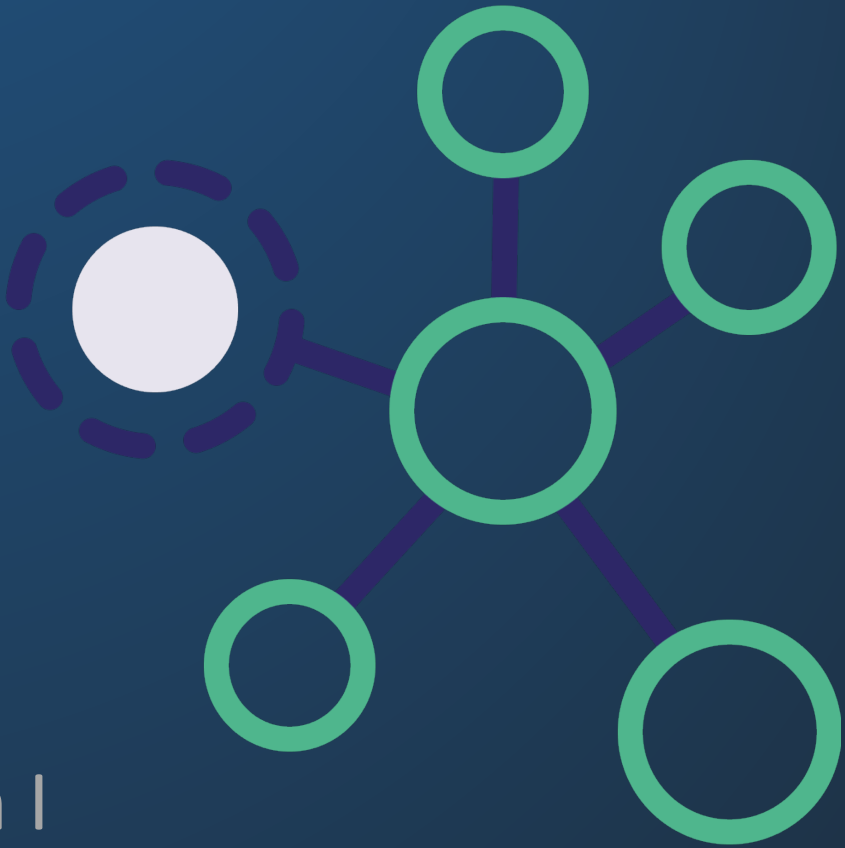
Introduction

Key questions

Applications

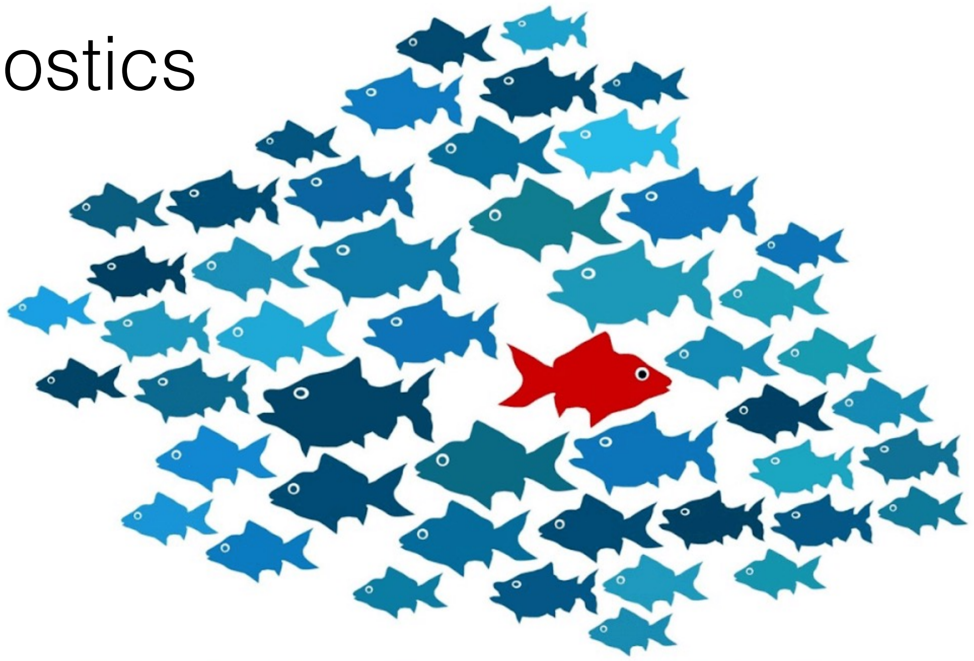
Techniques for anomaly detection I

Summary



An overview of applications

- Anomaly detection has a **broad range of applications**:
  - Network intrusion detection
  - Insurance / credit card fraud detection
  - Healthcare informatics / medical diagnostics
  - Industrial damage detection
  - Image processing / video surveillance
  - Novel topic detection in text mining
  - … and many more



Intrusion detection

- This refers to the process of monitoring events occurring in a computer system or network and analysing them for intrusions. Intrusions are attempts to **bypass the security mechanisms** of a computer or network.

- Traditional approaches rely on **recognising signatures of known attacks**. They cannot detect emerging cyber threats and are prone to delays deployments of new signatures.

Anomaly detection can alleviate these limitations.

Fraud detection

- This refers to **detection of criminal activities** in various settings. For example, what looks like a malicious user might be a real customer or could indeed be someone else posing as a customer (identity theft).

- Anomaly detection can be applied to a **range of frauds**, like credit card fraud, insurance claim fraud, mobile phone fraud, insider trading, etc.

- **Key challenges** of fraud detection are fast and accurate real-time anomaly detection because misclassification costs are usually very high.

Health(care) informatics

- Healthcare informatics concerns the use of algorithms to improve communication, understanding, and **management of medical information**. For example, we want to **detect anomalous patient** records to indicate disease outbreaks, instrumentation errors, etc.

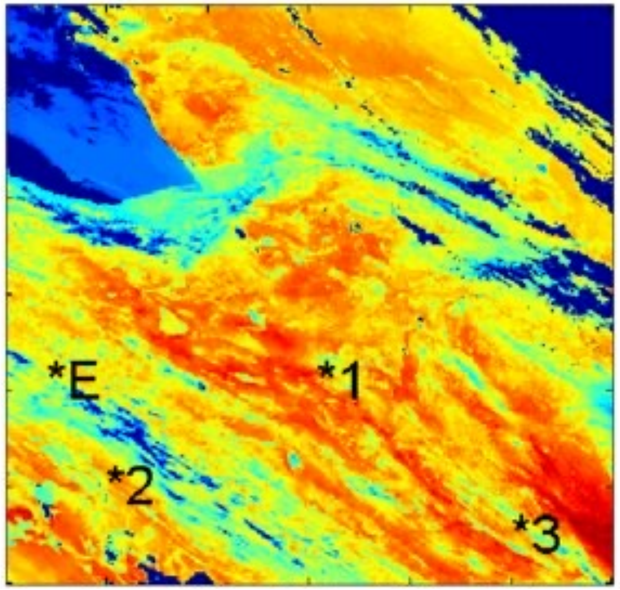
- **Key challenges** in the field include
  - Only normal labels are available.
  - Data can be very complex (spatio-temporal, highly multivariate, …).
  - Misclassification costs are very high.

Industrial damage detection

- This refers to the detection of **faults and failures in complex industrial systems**, structural damages, or abnormal energy consumption. An example is **aircraft safety**, where we want to look at anomalous aircraft usage, anomalies in engine combustion data or total aircraft health.

- **Key challenges** in this area include
  - Extremely large, noisy and unlabelled datasets.
  - Most applications exhibit temporal behaviour.
  - Detecting anomalies often requires immediate and far-reaching intervention.
  - Misclassification costs are very high

Image processing



- Here, anomaly detection often takes **two forms**:
  - Finding outliers in an image/video monitored over time.
  - Detecting anomalous regions within an image.
- For example, we might want to detect
  - Thermal anomalies in satellite images before an earthquake.
  - Potentially dangerous luggage pieces at an airport (video surveillance).
  - Cell abnormalities in mammographic images.
- **Key challenges** in this area include the detection of collective anomalies and the handling of very large datasets.

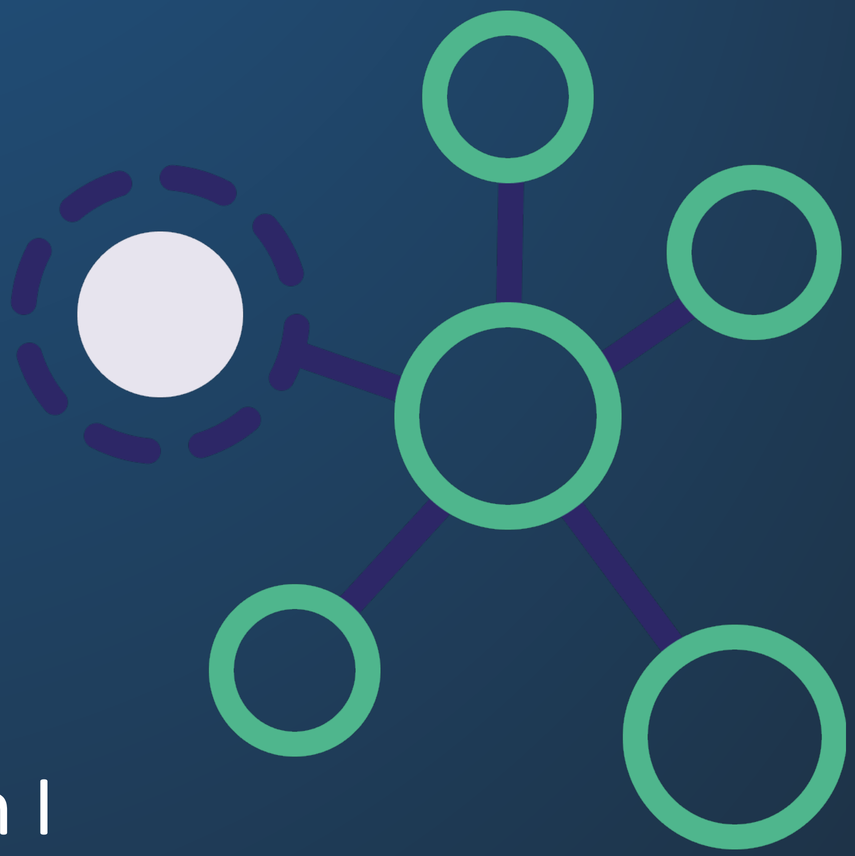
Introduction

Key questions

Applications

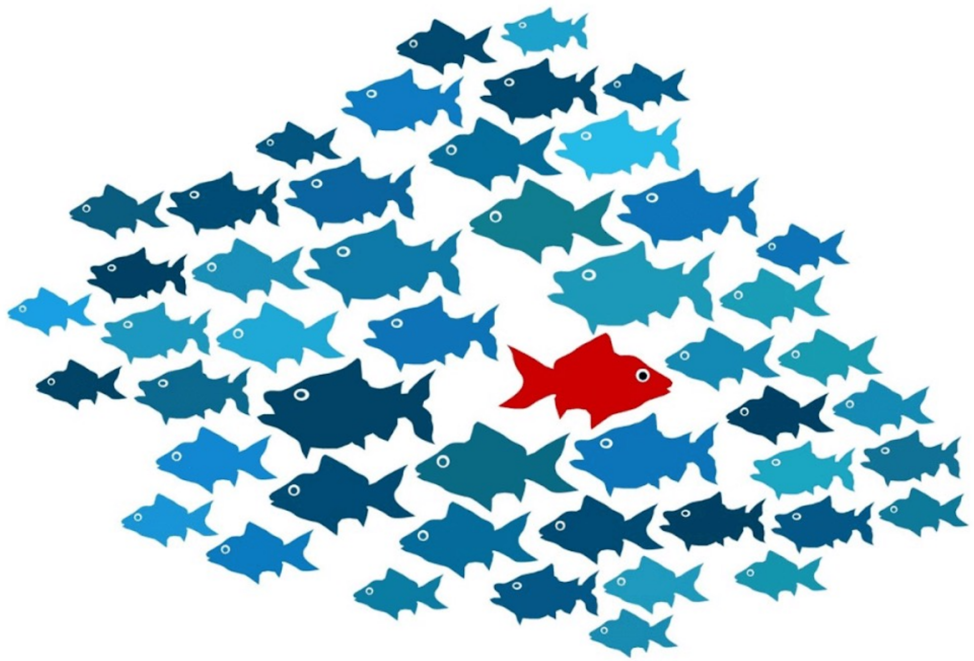
Techniques for anomaly detection I

Summary



Variants of anomaly detection

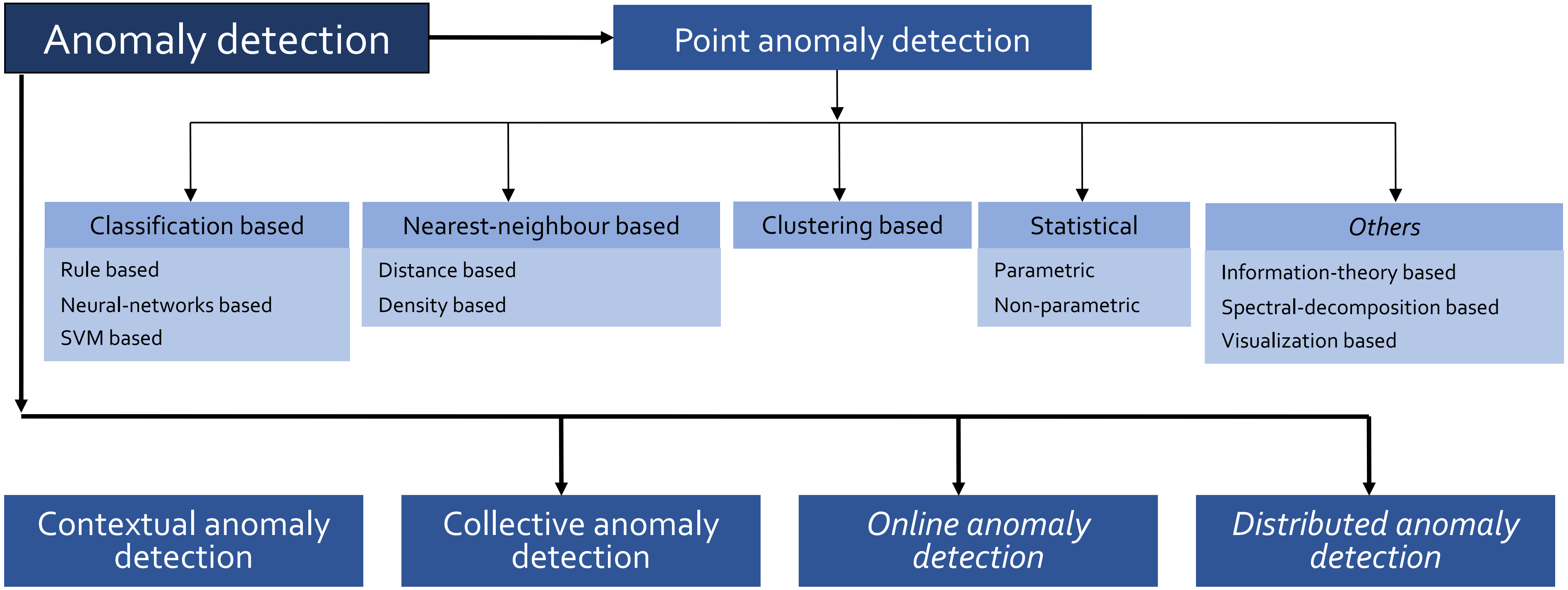
- We will typically focus on **obtaining scores** from our anomaly detection. We can specifically distinguish the following cases:
  - Given a dataset D, we want to find all the data points $\mathbf{x} \in \mathbf{D}$ with anomaly scores greater than some threshold t.
  - Given a dataset D, we want to find all the data points $\mathbf{x} \in \mathbf{D}$ that have the top-n largest values of anomaly scores.
  - Given a dataset D, containing mostly normal data points, and a test point $\mathbf{x}$, we want to compute the anomaly score of $\mathbf{x}$ with respect to D.



Taxonomy of techniques

An overview of anomaly detection approaches

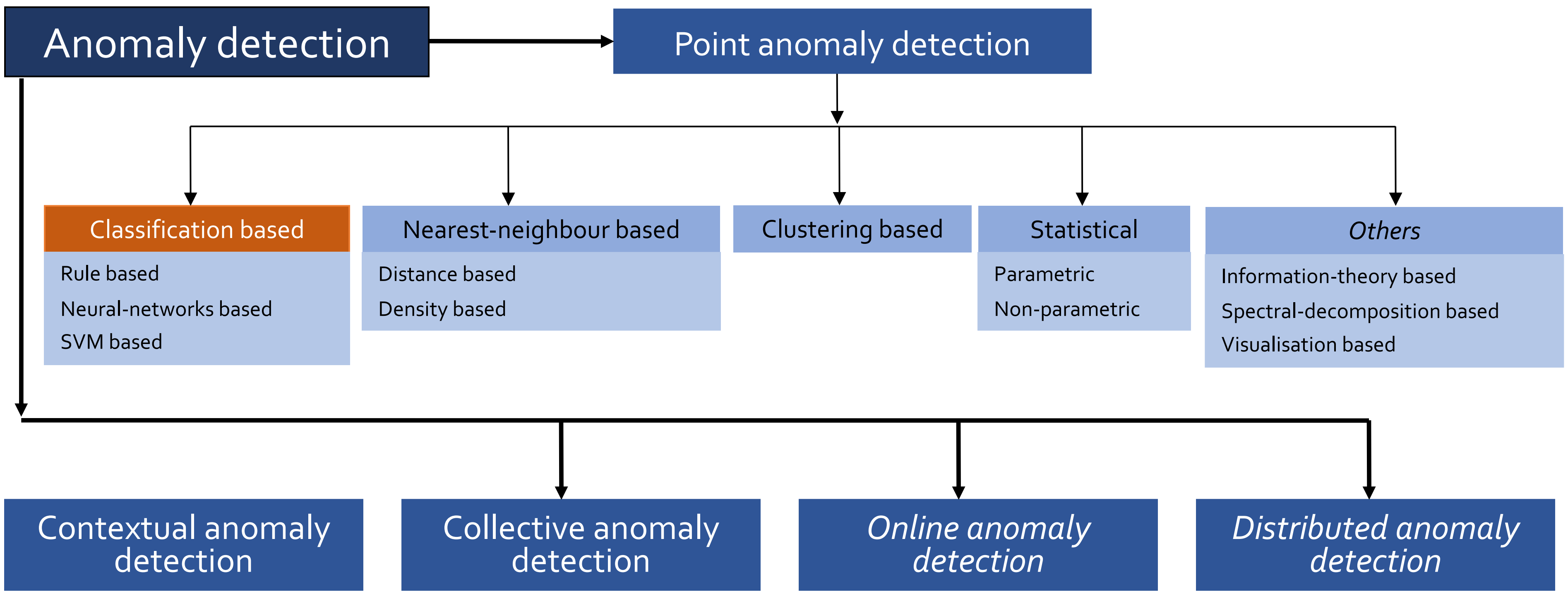
- In the remainder of this class and next week, we will look at the following **methods of anomaly detection** (following Chandola et al. (2008) – Anomaly Detection: A Survey; see reading materials):



Taxonomy of techniques

An overview of anomaly detection approaches

- In the remainder of this class and next week, we will look at the following **methods of anomaly detection** (following Chandola et al. (2008) – Anomaly Detection: A Survey; see reading materials):



Classification-based techniques

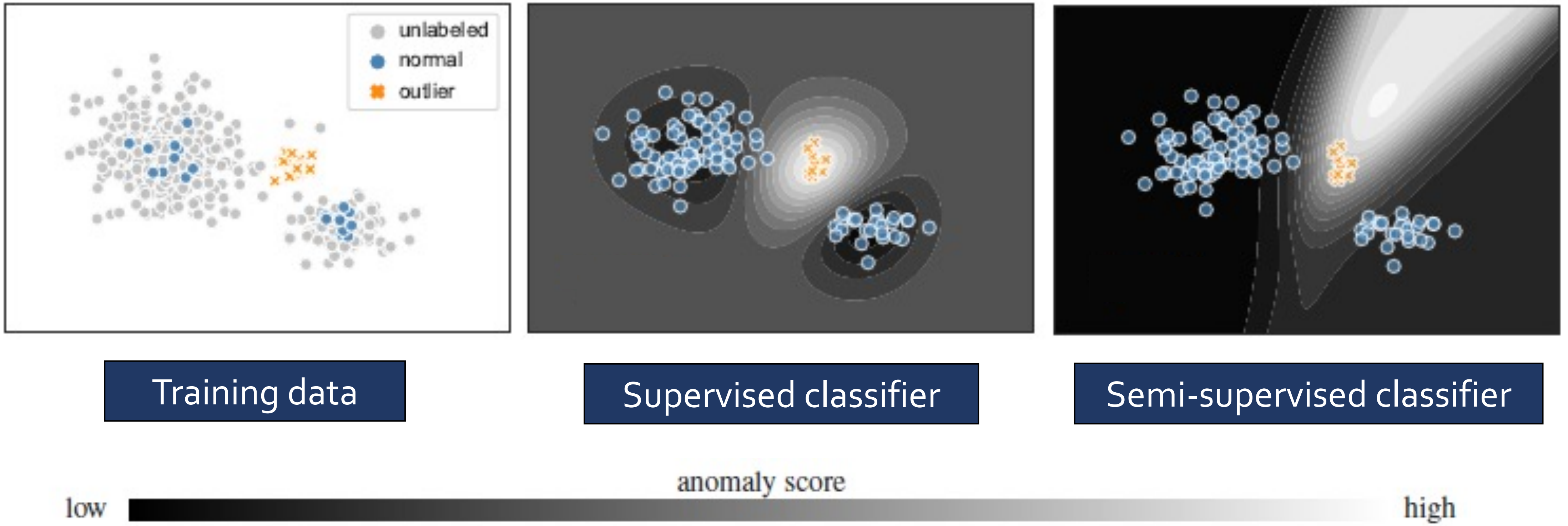
Main ideas when labelled data is available

- We want to build a classifier for normal and anomalous (rare) data based on **labelled training data** and then use it to classify new unseen events.
- Such classification models must be able to **handle skewed** (imbalanced) class distributions. We distinguish the following **two categories:**
  - **Supervised classification techniques**:
    - Require knowledge of the normal and anomaly class.
    - Build classifier to distinguish between normal and known anomalies.
  - **Semi-supervised classification techniques**:
    - Require knowledge of the normal class only.
    - Modify a classification model to learn the normal behaviour and then detect any deviations from normal behaviour as anomalous.

Classification-based techniques

A visual example

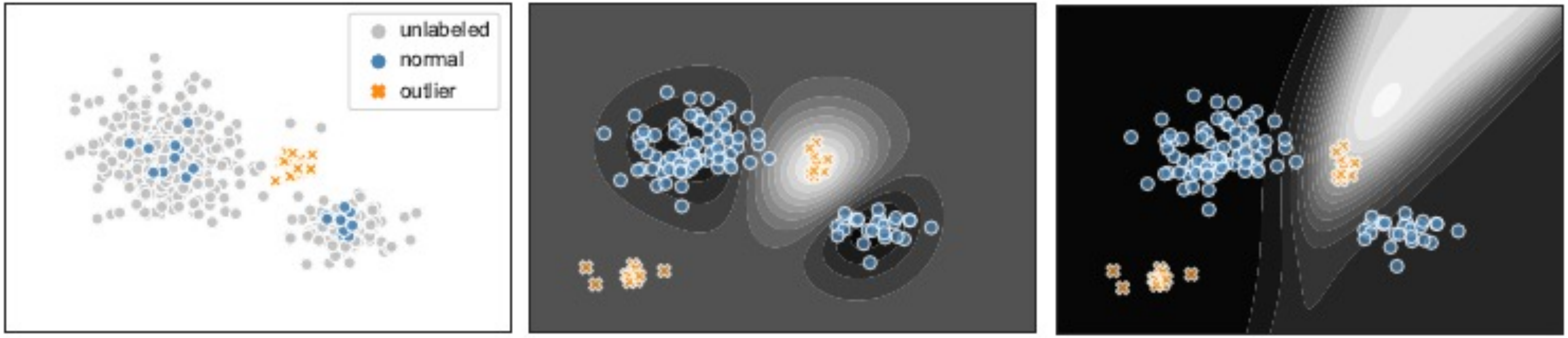
- The main idea of these two types is summarised in the following plots. To show how the models learn from some training data, we show the **anomaly score** as background colours and contour lines.



Classification-based techniques

A visual example

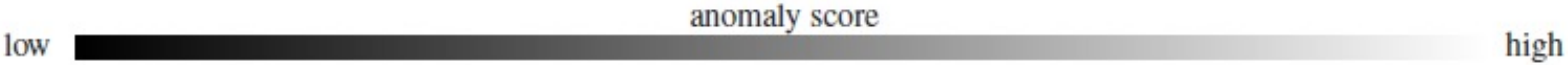
- The main idea of these two types is summarised in the following plots. To show how the models learn from some training data, we show the **anomaly score** as background colours and contour lines.



Training data

Supervised classifier

Semi-supervised classifier



Classification-based techniques

Pros and cons

ADVANTAGES

Supervised techniques:

- Models can be easily interpreted.
- High accuracy of detecting many kinds of known anomalies.

Semi-supervised techniques:

- Models can be easily interpreted.
- Normal behaviour can be accurately learned.

DISADVANTAGES

Supervised techniques:

- Require labels for both classes.
- Cannot detect unknown and emerging anomalies.

Semi-supervised techniques:

- Require labels for normal class.
- Possible high false positive rate - unseen (yet legitimate) data may be recognized as anomalies.

Classification-based techniques
Supervised anomaly detection

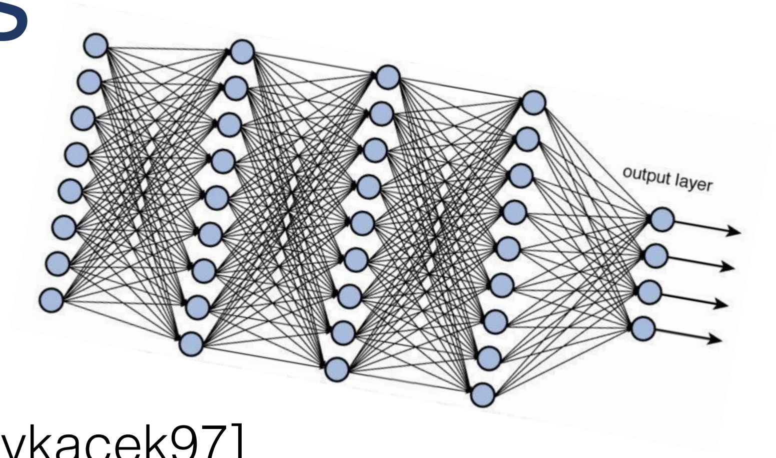
- Supervised approaches include the following:
  - Manipulating data records (oversampling / undersampling / generating artificial examples)
  - Rule-based techniques
  - Model-based techniques
    - Neural network-based approaches
    - Support vector machines (SVM) based approaches
    - Bayesian network-based approaches
  - Cost-sensitive classification techniques
  - Ensemble-based algorithms (SMOTEBoost, RareBoost, MetaCost)

Classification-based techniques
Supervised anomaly detection

- Supervised approaches include the following:
  - Manipulating data records (oversampling / undersampling / generating artificial examples)
  - Rule-based techniques
  - Model-based techniques
    - Neural network-based approaches
    - Support vector machines (SVM) based approaches
    - Bayesian network-based approaches
  - Cost-sensitive classification techniques
  - Ensemble-based algorithms (SMOTEBoost, RareBoost, MetaCost)

Classification-based techniques

Supervised: neural network-based examples



- **Multi-layer perceptron:**
  - Measuring the activation of output nodes [Augusteijn02]
  - Extending learning beyond decision boundaries:
    - Error bars as a measure of confidence for classification [Sykacek97]
    - Flexible hyperplanes for separating between classes [Vasconcelos95]

- **Auto-associative neural networks:**
  - Replicator neural networks [Hawkins02]
  - Hopfield networks [Jagota91, Crook01]

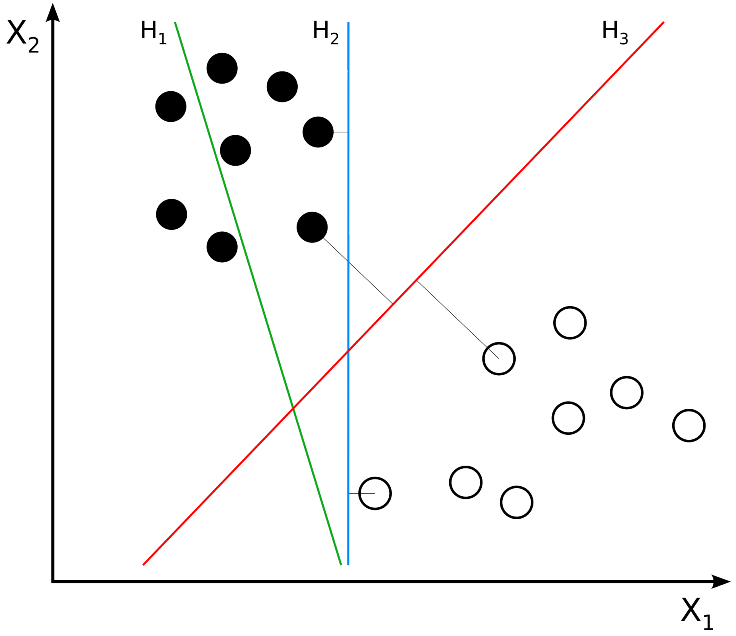
- **Radial basis functions:** reverse connections from output to central layer assigns neurons normal distribution. New instances that don't fit such distributions are anomalies [Albrecht00, Li02].

- **Oscillatory networks:** relaxation time of oscillatory neural nets is used as a criterion for novelty detection when a new instance is presented [Ho98, Borisyuk00].

Classification-based techniques

Supervised: support vector machine (SVM) examples

- A quick reminder: SVM are a type of supervised classifier that aims to separate data points (n-dimensional vectors) with an (n-1)-dimensional hyperplane that maximises the margins (distances) to the classes.



- Pioneered by the following two studies:
  - Mukkamala02: normal and anomalous data records are labelled and SVMs used for standard classification
  - Steinwart05: normal data records belong to high-density data regions, while anomalies belong to low-density ones; use SVM to classify data density levels

Classification-based techniques

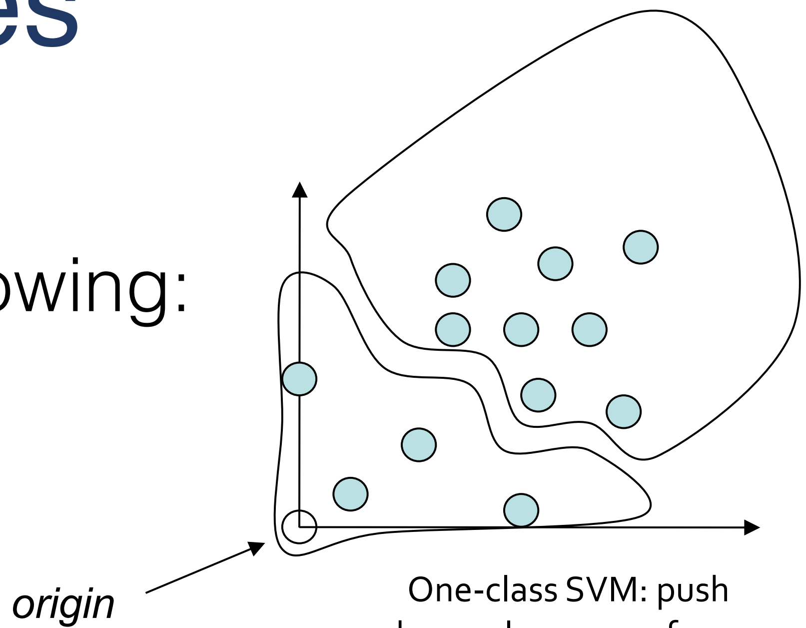
Semi-supervised anomaly detection

- Semi-supervised approaches include the following:
  - Rule-based techniques
  - Model-based techniques
    - Neural network-based approaches
    - SVM-based approaches
    - Markov model-based approaches (covered next semester)

Classification-based techniques

Semi-supervised anomaly detection

- Semi-supervised approaches include the following:
  - Rule-based techniques
  - Model-based techniques
    - Neural network-based approaches
    - SVM-based approaches
    - Markov model-based approaches (covered next semester)



One-class SVM: push hyperplane away from origin as much as possible

For SVMs, anomaly detection is converted into a one-class classification problem: the idea is to separate the entire set of training data from the origin, i.e., to find a small region where most of the data lies. Points in this region are then labelled as the normal class. Everything else is an anomaly [Scholkopf99].

Classification-based techniques

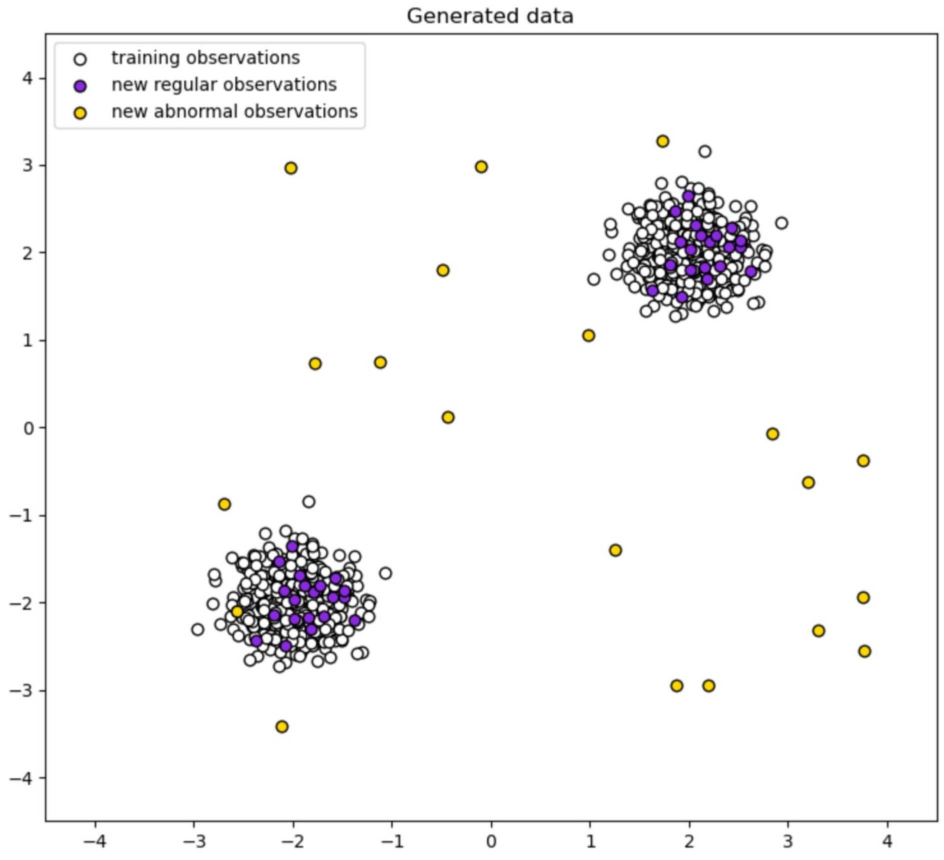
An SVM example using scikit-learn

- We generate some training data, normal observations and anomalous points:

```
# Generate training data points.
# Two clusteres are centred at −2 and +2.
X = 0.3 * rng.randn(500, 2)
X_train = np.r_[X + 2, X - 2]
```

```
# Generate some regular novel observations.
# These follow the same distribution as the training data.
X = 0.3 * rng.randn(20, 2)
X_test = np.r_[X + 2, X - 2]
```

```
# Generate some abnormal novel observations.
X_anomalies = rng.uniform(low=-4, high=4, size=(20, 2))
```



See 8_AD_SVM.ipynb
for the full code for this example.

Classification-based techniques

An SVM example using scikit-learn

- When training a SVM for anomaly detection, we can vary the **kernel** used to determine the shape of our hyperplane and the kernel's parameters. Different kernels result in different anomaly counts.
- In our example, we will focus on the **non-linear RBF** (radial-basis function) **kernel** that you have seen before.

```python
# SVM hyperparameters
nu = 0.1
gamma = 0.1

# Create classifier instance and fit the model.
clf = OneClassSVM(kernel="rbf", gamma=gamma, nu=nu)
clf.fit(X_train)

# Predict on the three datasets.
y_pred_train = clf.predict(X_train)
y_pred_test = clf.predict(X_test)
y_pred_anomalies = clf.predict(X_anomalies)
```

Focus specifically on the one-class SVM approach.

Classification-based techniques

An SVM example using scikit-learn

The output will be 1 if the data is part of the normal class and -1 if the data instance is an anomaly.

```
print(y_pred_train[-20:])
```

```
[ 1  1  1  1  1  1  1 -1 -1  1  1  1  1  1  1 -1  1  1  1  1]
```

We can check how many classification errors were performed in the following way:

```
n_error_train = len(y_pred_train[y_pred_train == -1])
n_error_test = len(y_pred_test[y_pred_test == -1])
n_error_anomalies = len(y_pred_anomalies[y_pred_anomalies == 1])

print("Misclassifications in training dataset:", n_error_train)
print("Misclassifications in normal test dataset:", n_error_test)
print("Misclassifications in anomolous test dataset:", n_error_anomalies)
```

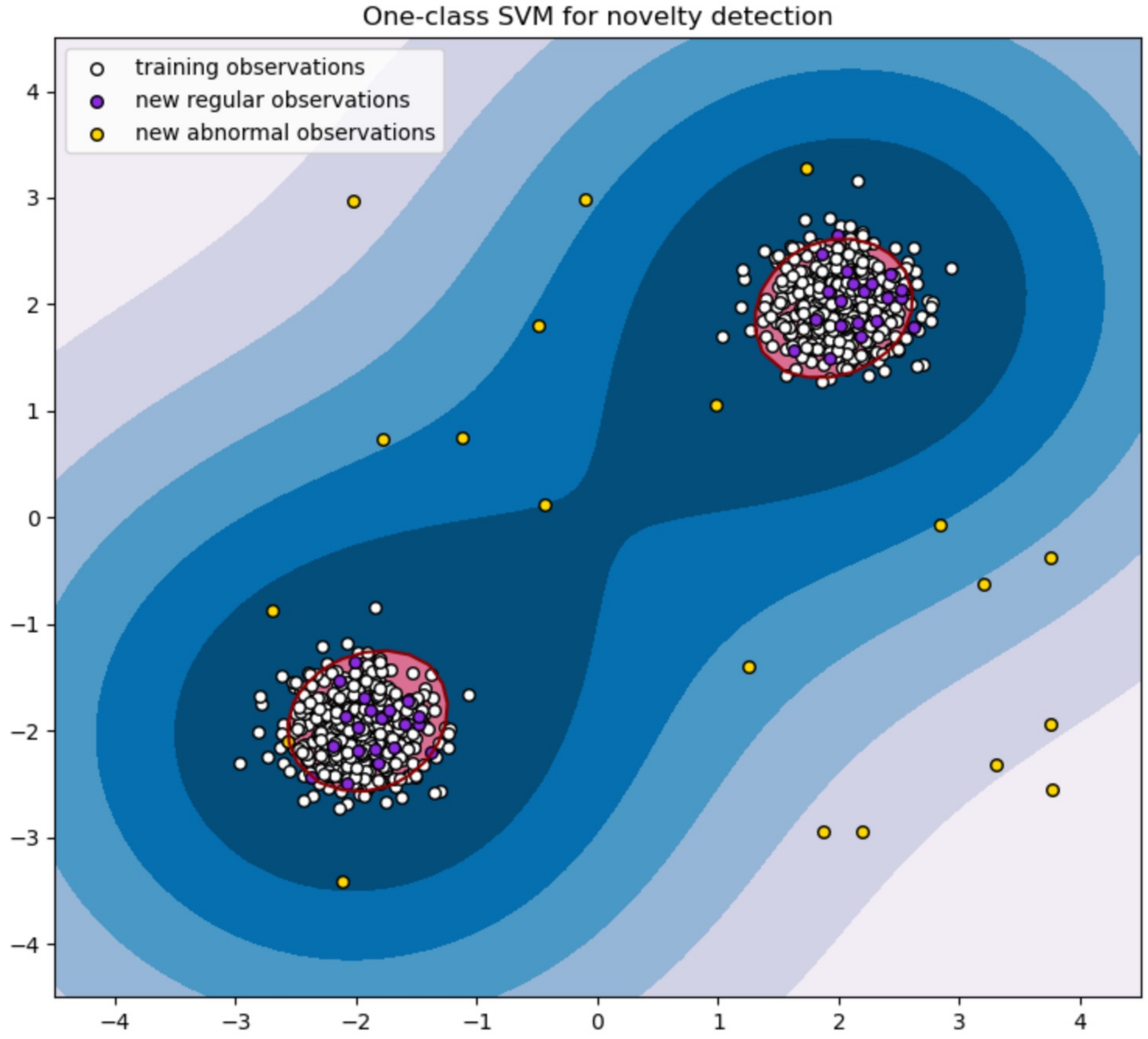
```
Misclassifications in training dataset: 100
Misclassifications in normal test dataset: 2
Misclassifications in anomolous test dataset: 1
```

Classification-based techniques

An SVM example using scikit-learn

- Plot contours of our **decision function** (distance of points from the hyperplane) to show the optimised model.

Because the one-class SVM is sensitive to outliers in the training data, it is best suited for novelty detection where the training set is not contaminated by outliers.



Classification-based techniques

Unsupervised anomaly detection: isolation forests

- The methods discussed so far, are only applicable to those instances where we have **labelled training data**. This is, however, not always given.
- In these cases, **unsupervised anomaly detection** approaches are needed. We will focus on a popular one, so-called **isolation forests**.

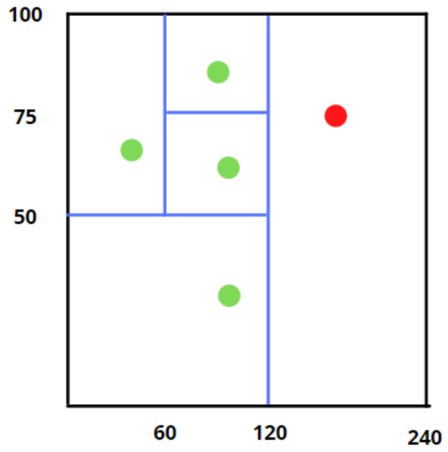
Isolation forests are an ensemble of "isolation trees" that "isolate" observations by recursive random partitioning, which can be represented by a tree structure.

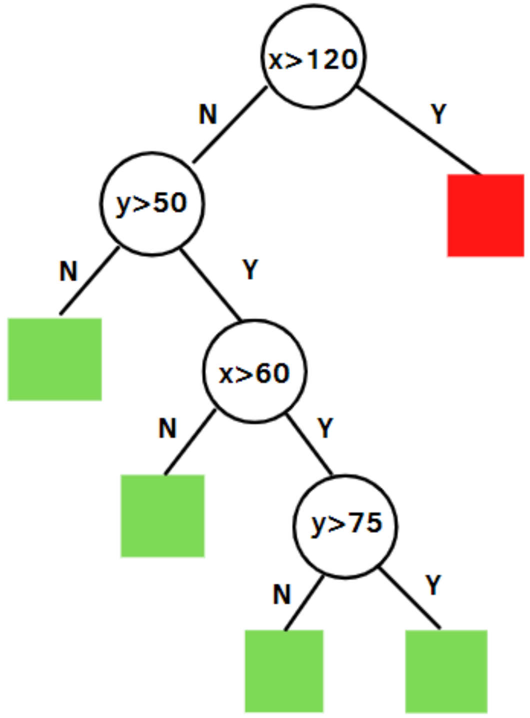
The number of partitions required to isolate a sample is lower for outliers and higher for inliers.

Classification-based techniques

Isolation forests step-by-step

- **Step 1**: A random subsample of data is selected for **binary tree construction.**
- **Step 2:** Branches are constructed by selecting a random feature (from a set of N features) first. Branching is then obtained by applying a (random) threshold (value between feature minimum and maximum).
- **Step 3**: If the value of a data point is less than the selected threshold, it is assigned to the left branch otherwise to the right. Each decision node is split into left and right branches.
- **Step 4**: Steps 2 & 3 are continued recursively until each data point is completely isolated or a maximum depth reached.
- **Step 5**: The above steps are repeated to construct a forest of random binary trees.





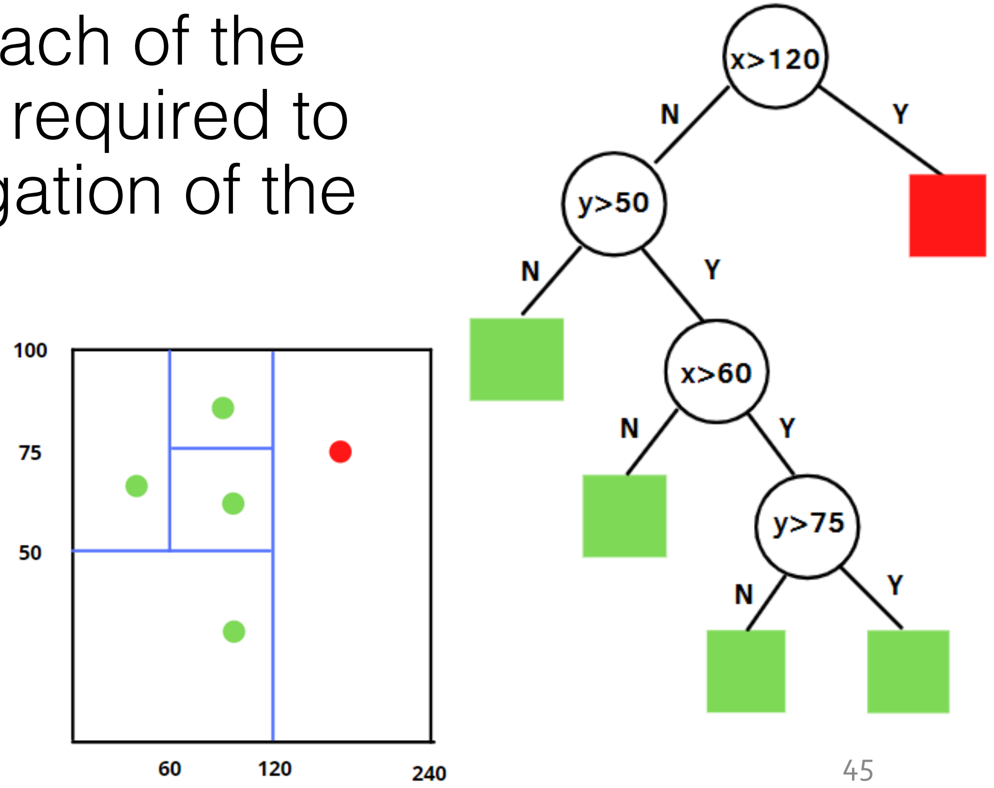
Classification-based techniques

Isolation forests step-by-step

- **Step 6**: After an ensemble of trees (isolation forest) is created, model training is complete. Then, during scoring, a new data point is traversed through all the trees which were trained previously.

- **Step 7**: An anomaly score is assigned to each of the data points based on the depth of the tree required to arrive at that point. This score is an aggregation of the depth obtained from each of the trees.

A final anomaly score of -1 is assigned to anomalies and 1 to normal points based on the percentage of anomalies present in the data (contamination parameter).

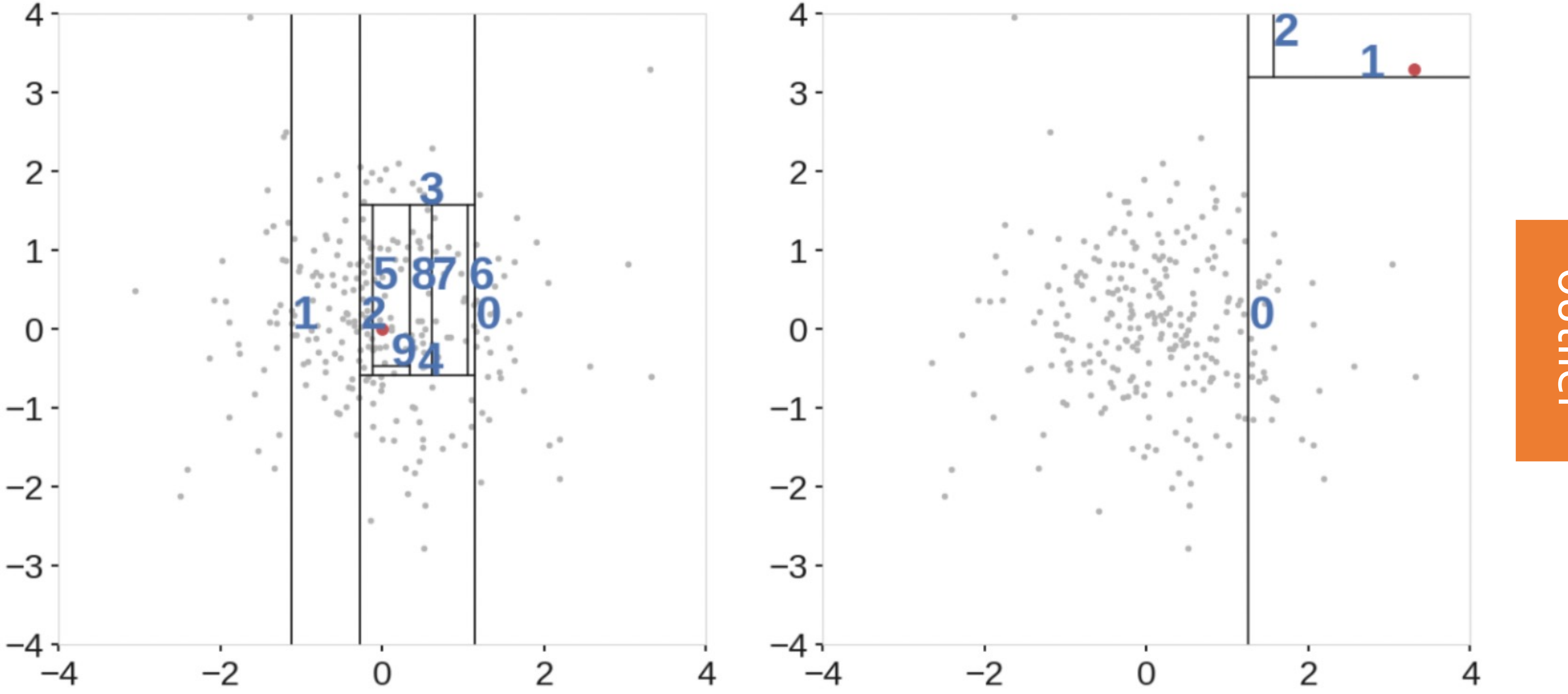


Classification-based techniques

Isolation forest example

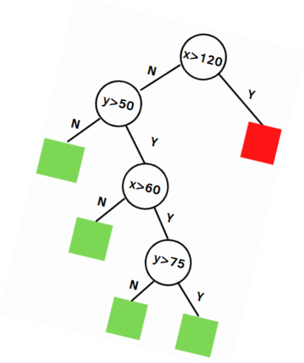
- In practice, a single binary tree might look as follows for a normal data point (inlier) and an anomalous point (outlier):

inlier



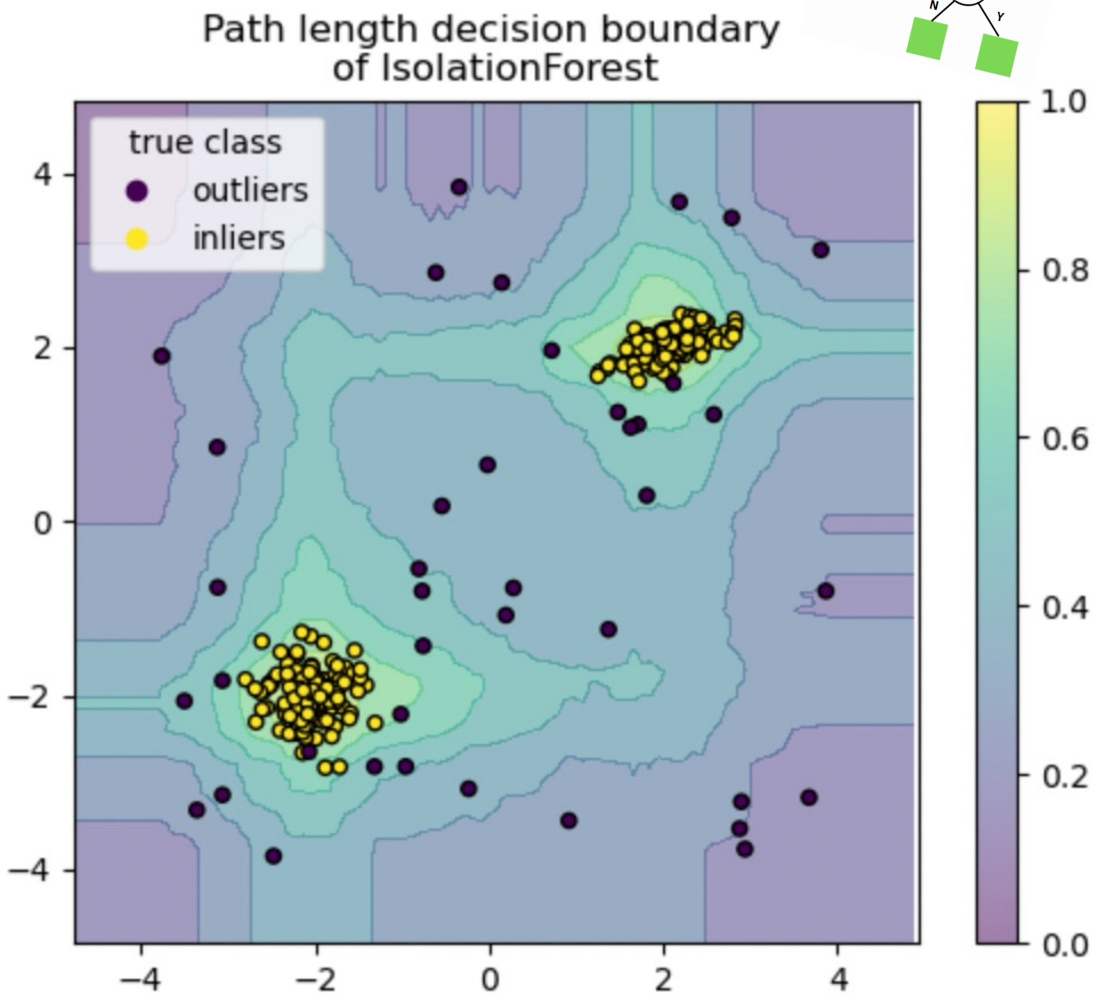
outlier

Classification-based techniques
Path lengths in isolation forests



- As recursive partitioning can be represented by a tree structure, the number of splittings needed to isolate a sample is equivalent to the **path length** from the root node to the terminating node.

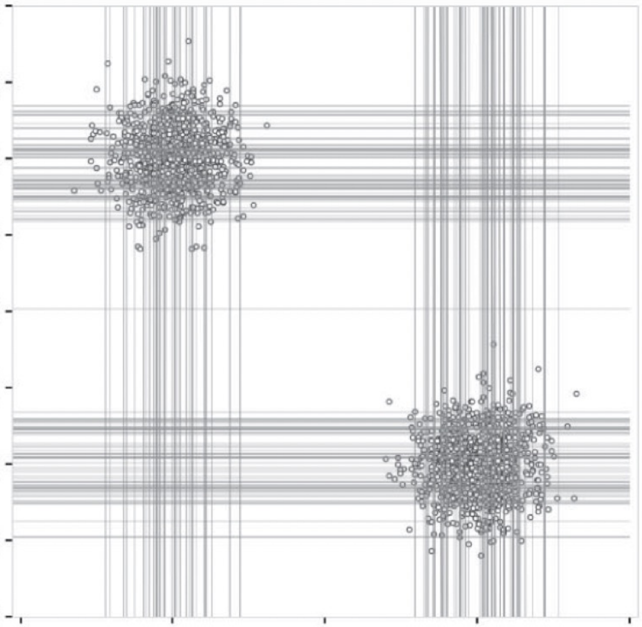
- The path length, averaged over a forest of random trees, **measures normality**.



> Random partitioning produces shorter paths for anomalies. Thus, when an isolation forest collectively produces shorter path lengths for some samples, they are likely to be anomalies.

Classification-based techniques

Isolation forest advantages and limitations



- Isolation forests are **computationally efficient** and have been proven to be very effective when performing **unsupervised anomaly detection**.

- However, there are a **few shortcomings**:
  - The final anomaly score depends on the **contamination parameter**, provided during training the model. This means we need an idea of what percentage of the data is anomalous beforehand to get a better prediction.
  - The approach **suffers a bias** due to the way the branching takes place (see images on right).

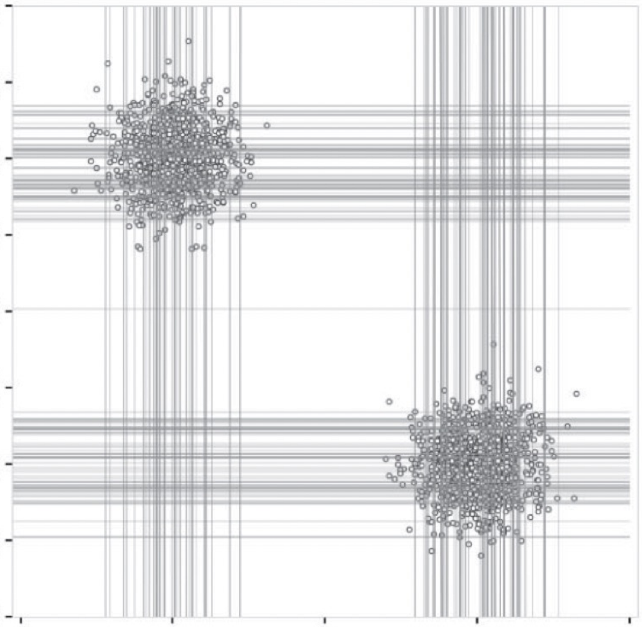
Classification-based techniques

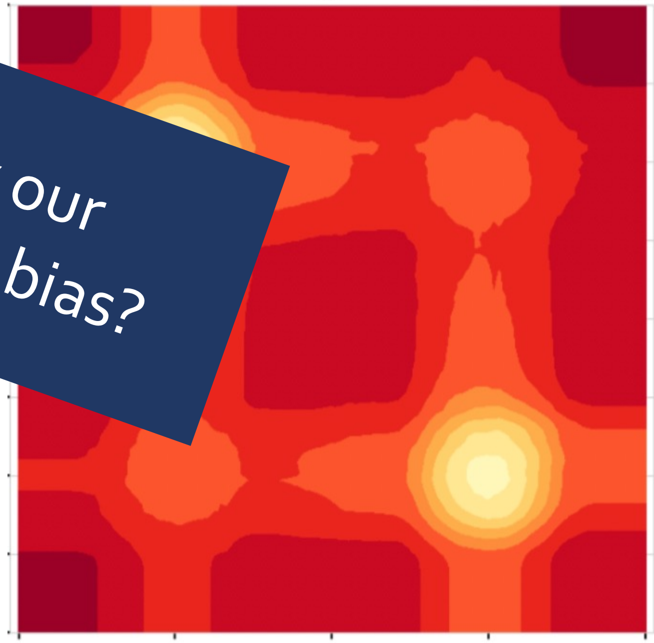
Isolation forest advantages and limitations

- Isolation forests ar**e computationally efficient** and have been pr... to be very effective when performing **unsu**... **maly detection**.

- However, there are a...
  - The final anomaly score depe... **contamination parameter**, provided ... training the model. This means we need an i... of what percentage of the data is anomalous beforehand to get a better prediction.
  - The approach **suffers a bias** due to the way the branching takes place (see images on right).

Any ideas how we could modify our decision boundaries to remove this bias?



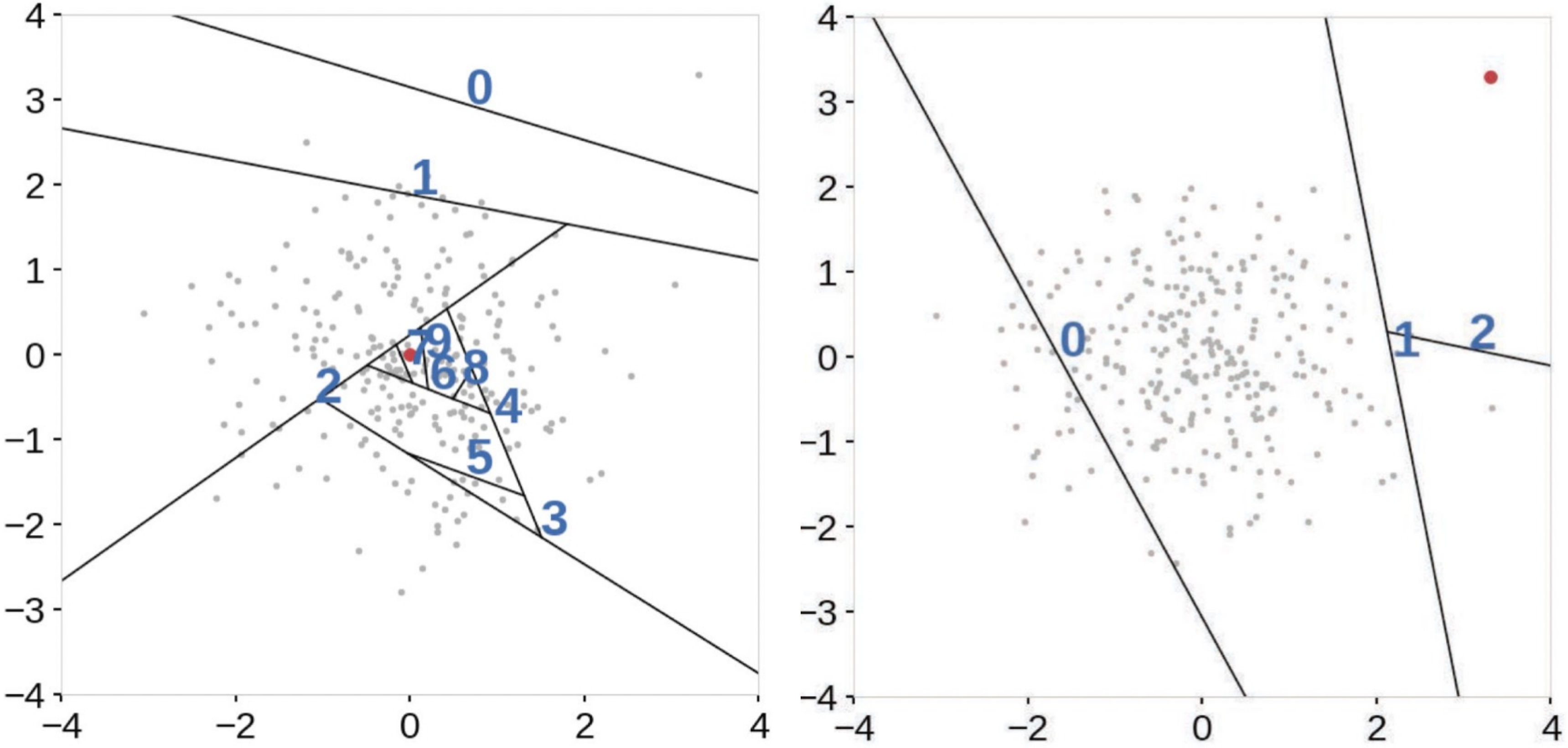


Classification-based techniques

Extended isolation forest

- We can remove this bias by allowing for sloped decision boundaries. See Hariri et al. (2021) on “Extended Isolation Forests” on Canvas.

inlier

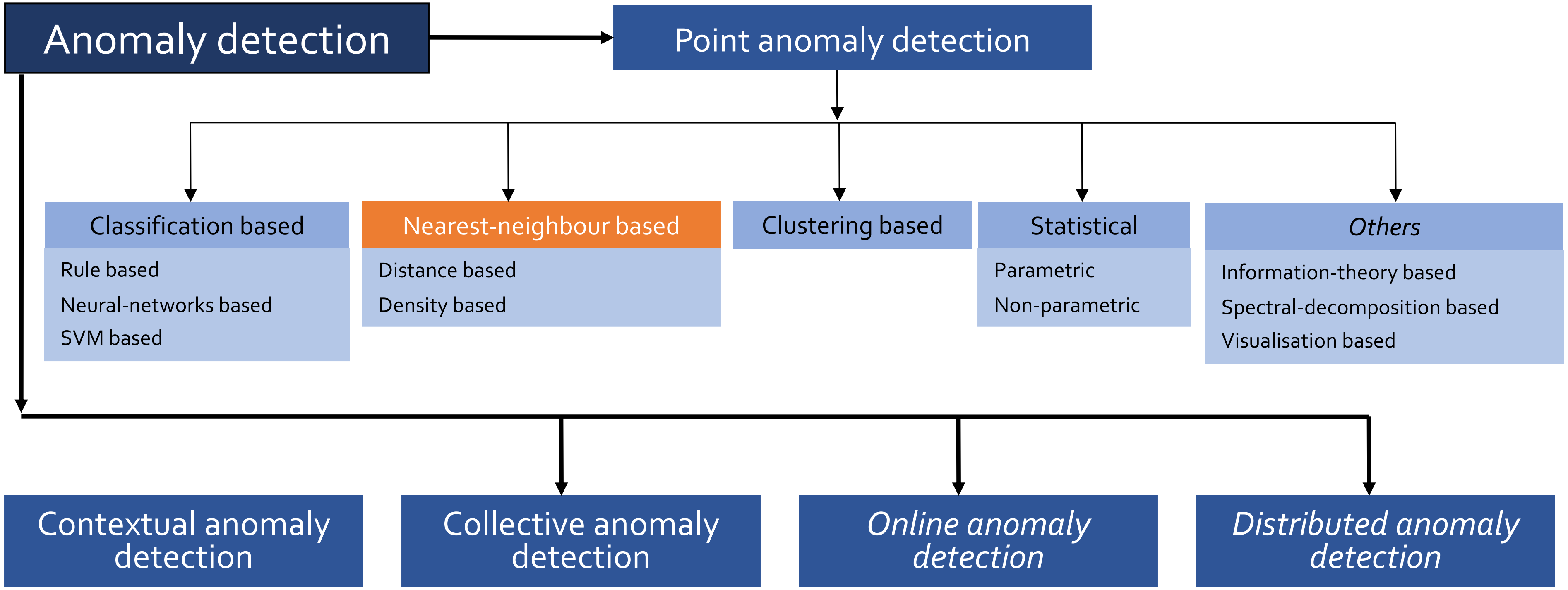


outlier

Taxonomy of techniques

An overview of anomaly detection approaches

- In the remainder of this class and next week, we will look at the following **methods of anomaly detection** (following Chandola et al. (2008) – Anomaly Detection: A Survey; see reading materials):



Nearest neighbour-based techniques

Two types of approaches

- We assumed that outliers are objects that are **located far away** from other objects, while normal data instances have close neighbours. Thus, it is natural to assume that we can detect outliers by **determining an anomaly score** that is based on **their neighbouring data points**.

- Nearest neighbour-based techniques generally involve **two steps**:
  - **Compute neighbourhood** for each data instance.
  - **Analyse the neighbourhood** to determine whether data is normal or not.

Distance-based methods: anomalies are data points most distant from other points.	Density-based methods: anomalies are data points in low-density regions.

Nearest neighbour-based techniques

Pros and cons

ADVANTAGES

- Can be used in supervised or semi-supervised settings.
- Easy to implement and interpret as closeness measures are relatively simple (easier than determining dataset statistics).
- They provide a quantitative measure of the degree to which an object is an outlier.
- Versatile and able to deal with the presence of multiple clusters.

DISADVANTAGES

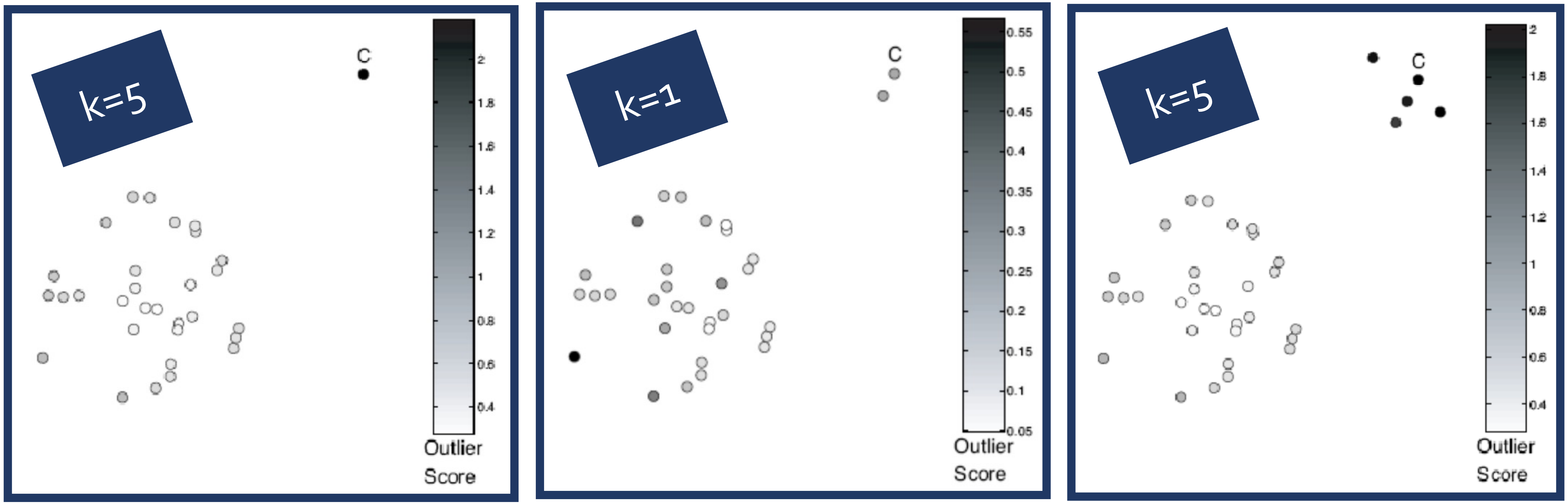
- They are computationally expensive for high-dimensional data.
- They are sensitive to the choice of relevant parameters and the distance/density measure used.
- They suffer from curse of dimensionality: closeness becomes less meaningful in higher dimensions.
- Do not work well if the normal point have too few neighbours.

Nearest neighbour-based techniques

Distance-based approaches: k-nearest neighbour

- **For distance-based methods:** A point, P, in a dataset is an outlier if at least a fraction, p, of data points lies greater than a distance, d, from point, P.

- An example is the k^{th} **nearest neighbour algorithm**, illustrated below.

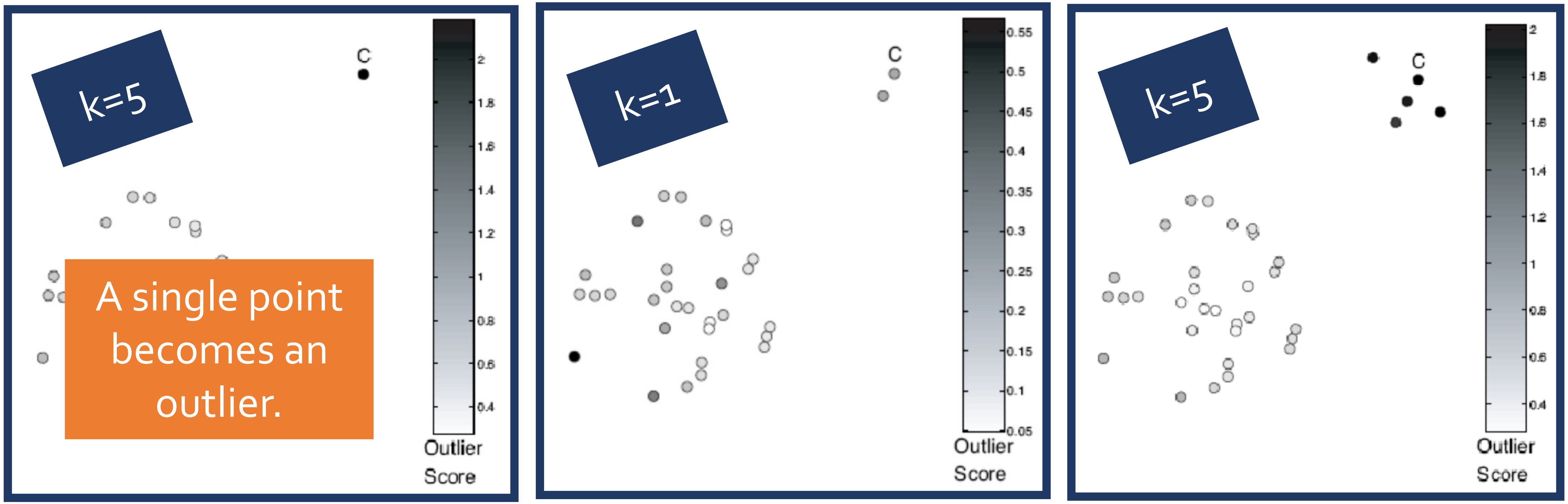


Nearest neighbour-based techniques

Distance-based approaches: k-nearest neighbour

- **For distance-based methods:** A point, P, in a dataset is an outlier if at least a fraction, p, of data points lies greater than a distance, d, from point, P.

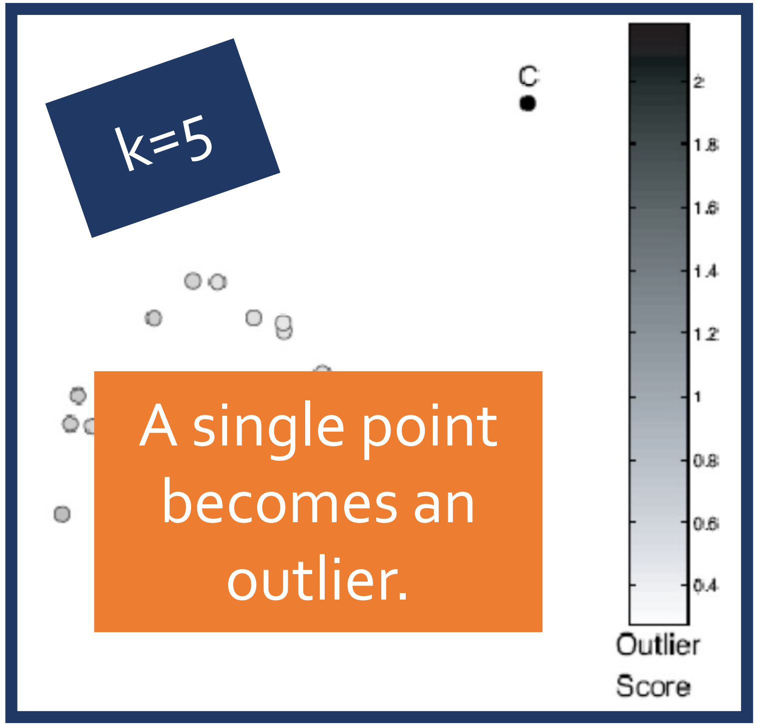
- An example is the **kth nearest neighbour algorithm**, illustrated below.

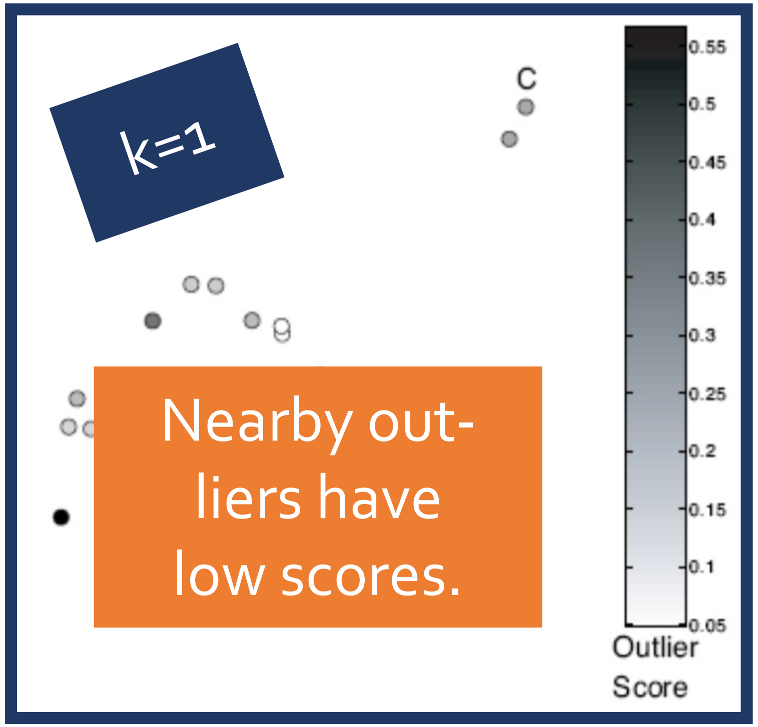


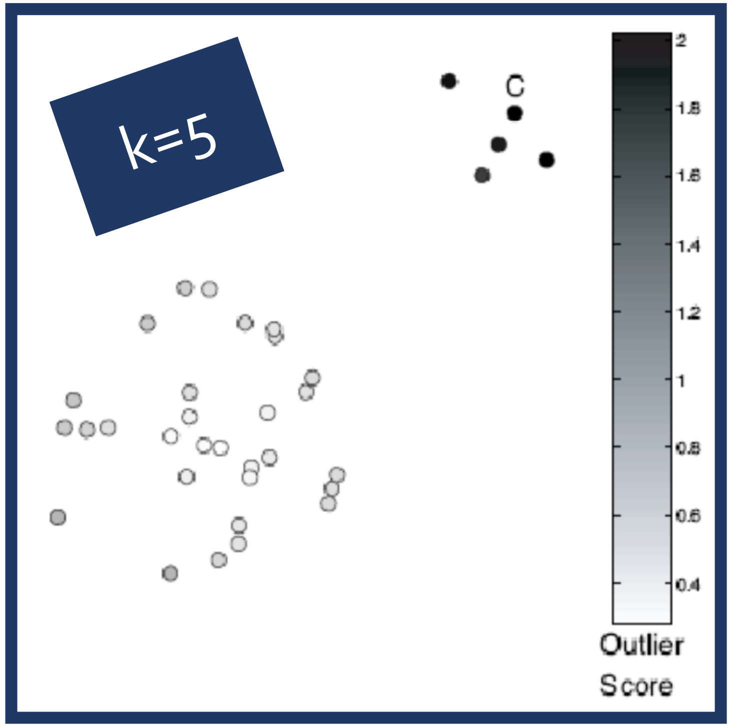
Nearest neighbour-based techniques
Distance-based approaches: k-nearest neighbour

- **For distance-based methods:** A point, P, in a dataset is an outlier if at least a fraction, p, of data points lies greater than a distance, d, from point, P.

- An example is the **kth nearest neighbour algorithm**, illustrated below.





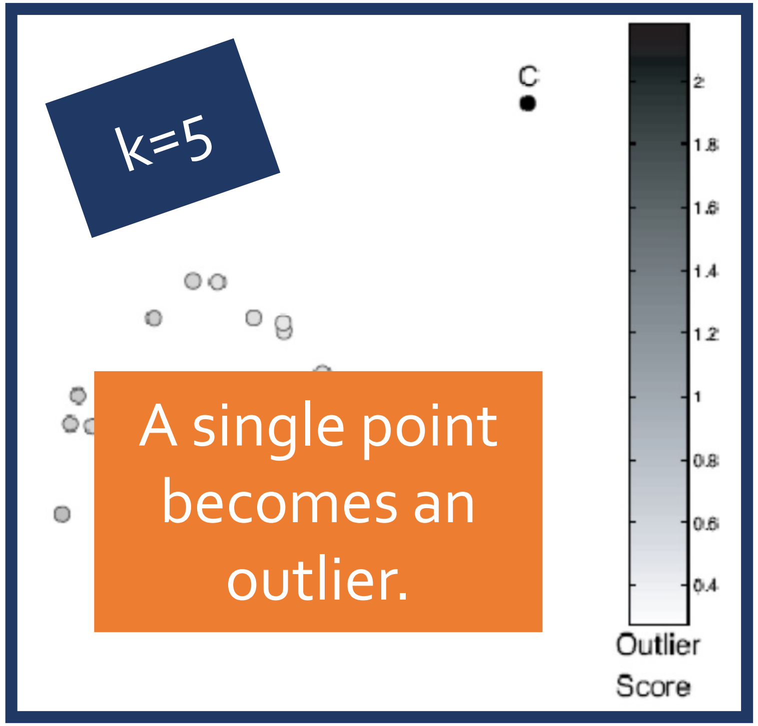


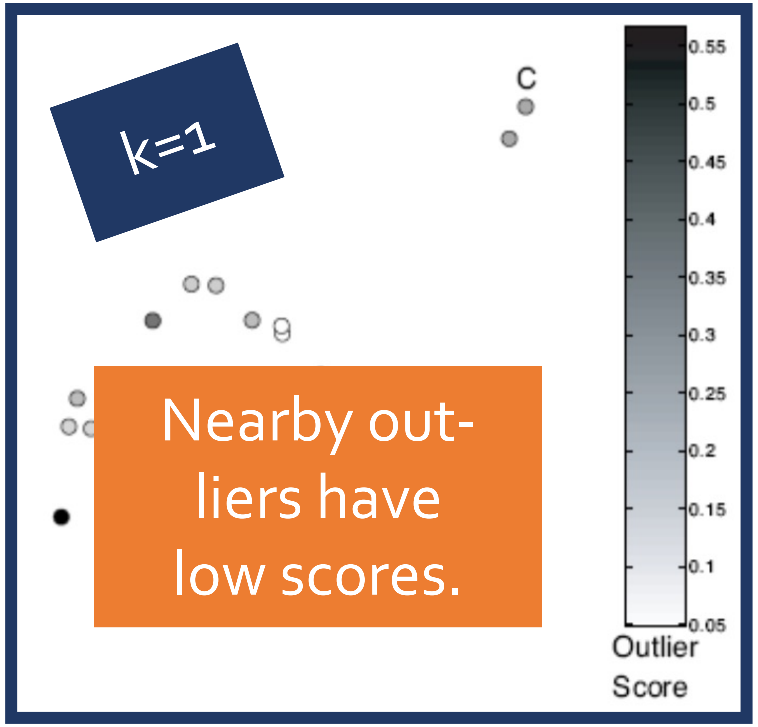
Nearest neighbour-based techniques

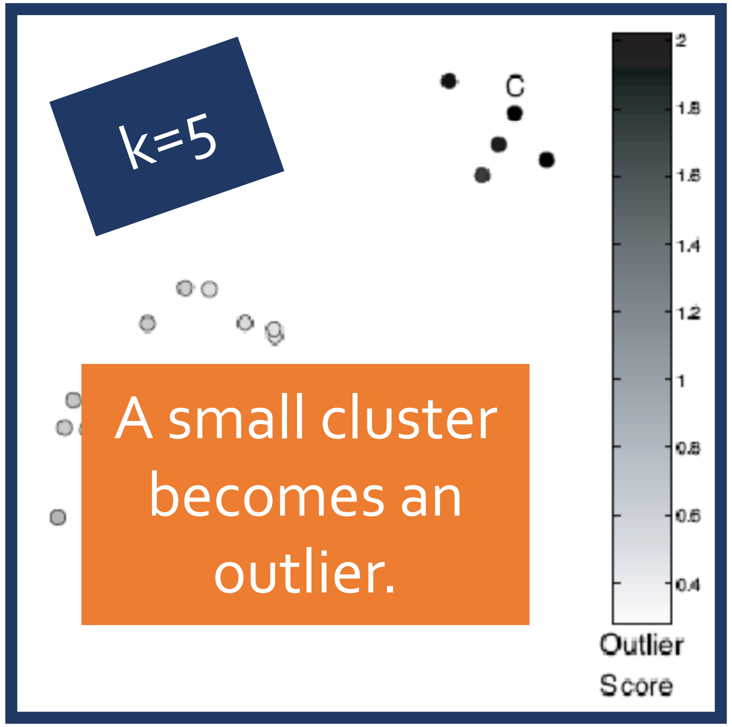
Distance-based approaches: k-nearest neighbour

- **For distance-based methods:** A point, P, in a dataset is an outlier if at least a fraction, p, of data points lies greater than a distance, d, from point, P.

- An example is the k^{th} **nearest neighbour algorithm**, illustrated below.







Nearest neighbour-based techniques

Distance-based approaches: k-nearest neighbour

- **For distance-based methods:** A point, P, in a dataset is an outlier if at least a fraction, p, of data points lies greater than a distance, d, from point, P.

- An example is th **earest neighbour algorithm**, illustrated below.

This illustrates one of our disadvantages: the anomaly score is very sensitive to the choice of k.

Nearest neighbour-based techniques

Density-based approaches

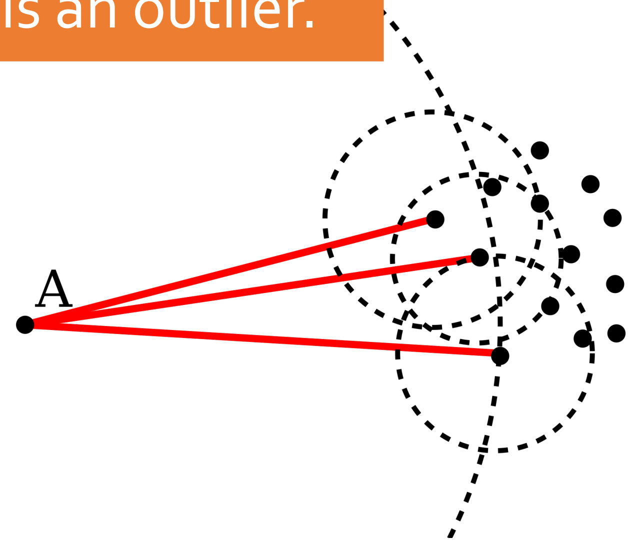
- Here, we want to compute the **local densities** of particular regions and declare instances in low-density regions as potential anomalies.
- Density-based approaches include:
    - Local outlier factor (LOF)
    - Connectivity outlier factor (COF)
    - Multi-granularity deviation factor (MDEF)

Nearest neighbour-based techniques
Density-based approaches

- Here, we want to compute the **local densities** of particular regions and declare instances in low-density regions as potential anomalies.

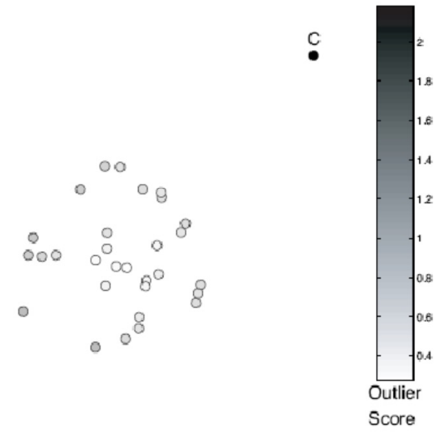
- Density-based approaches include:
  - Local outlier factor (LOF)
  - Connectivity outlier factor (COF)
  - Multi-granularity deviation factor (MDEF)

Point A has a much lower density than the others. It is an outlier.



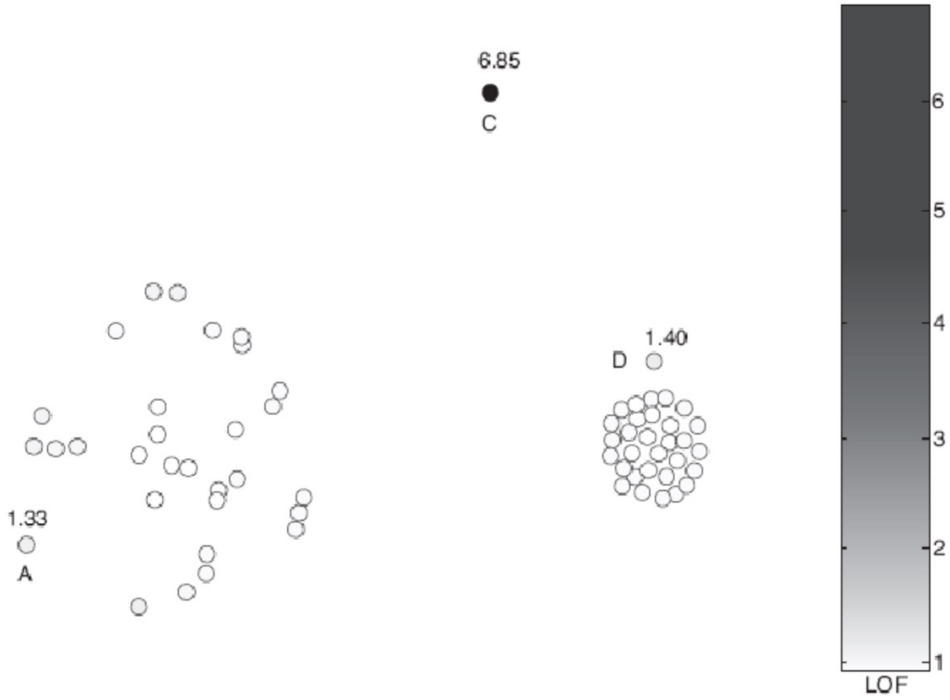
LOF incorporates the concept of local density. Locality is defined through the k nearest neighbours and their distance is used to estimate the density. We then compare the local densities of a point to the average local density of its k nearest neighbours.

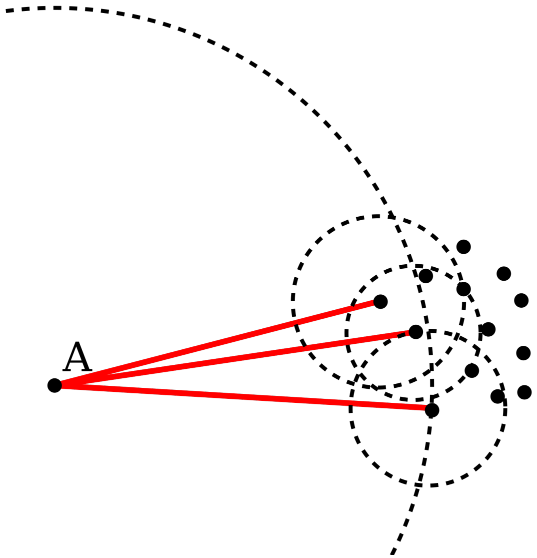
Nearest neighbour-based techniques
Density-based approaches: LOF



- With this in mind, we return to our **earlier examples** for the distance-based k-nearest neighbour approach.
- Applying a **density-based LOF method**, we obtain

The density-based LOF method is more successful at handling situations with various densities and identifying the outlier point C.





Nearest neighbour-based techniques

Density-based approaches: outlier detection with LOF in scikit-learn

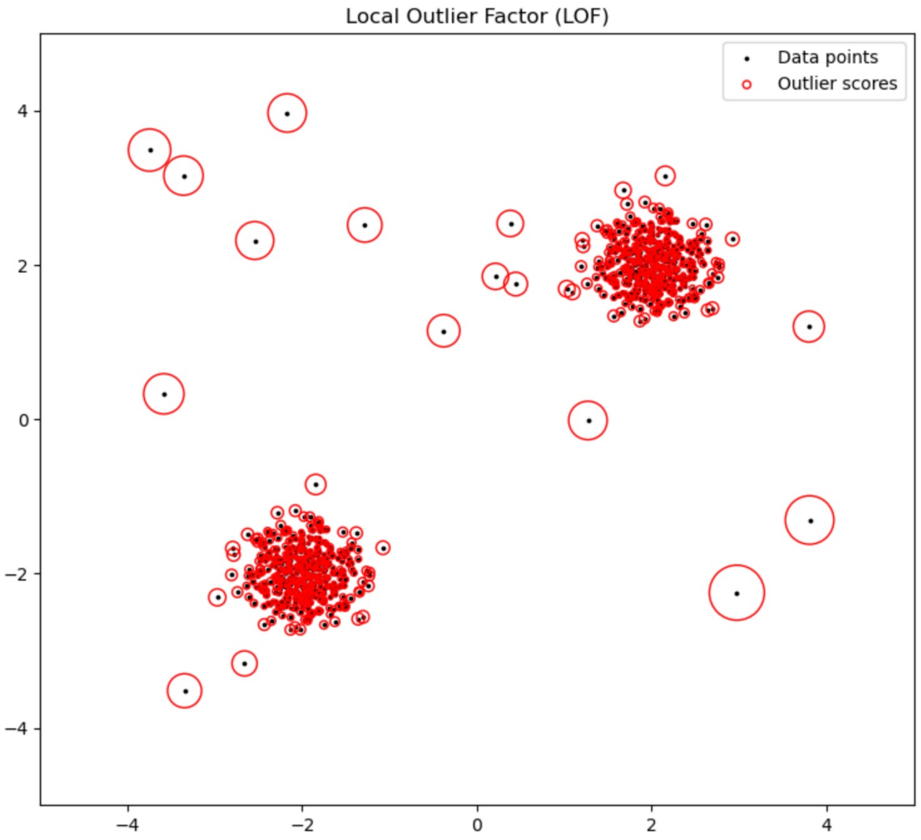
- To test this in Python, we first generate some normal and anomalous data, and then fit the LocalOutlierFactor() function:

```python
# Generate training data points.
# Two clusteres are centred at -2 and +2.
X = 0.3 * rng.randn(500, 2)
X_inliers = np.r_[X + 2, X - 2]
```

```python
# Generate some outlier observations.
X_outliers = rng.uniform(low=-4, high=4, size=(20, 2))
```

```python
# Fit the model for outlier detection with default parameters.
clf = LocalOutlierFactor(n_neighbors=20, contamination="auto")

# Use fit_predict to compute the predicted labels of the training samples.
y_pred = clf.fit_predict(X)
X_scores = clf.negative_outlier_factor_
```



See 8_AD_LOF.ipynb
for the full code for this example.

Nearest neighbour-based techniques

Density-based approaches: novelty detection with LOF

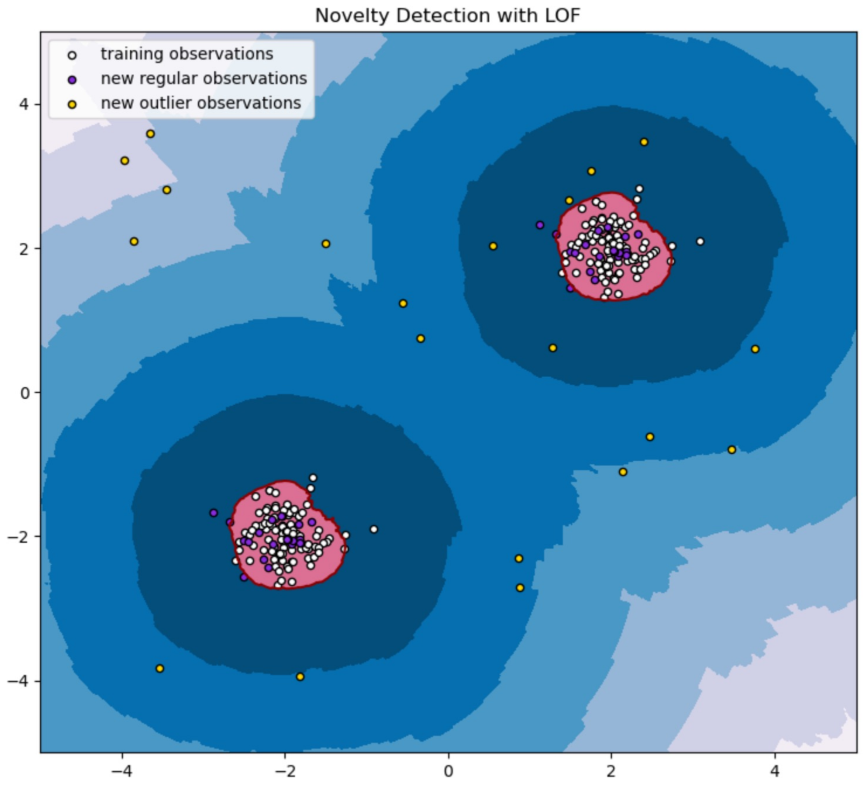
- To test this in Python, we generate some training data, normal test data and several outliers, and then fit the LocalOutlierFactor() function:

```
# Generate normal (not abnormal) training observations.
X = 0.3 * np.random.randn(100, 2)
X_train = np.r_[X + 2, X - 2]

# Generate new normal (not abnormal) observations.
X = 0.3 * np.random.randn(20, 2)
X_test = np.r_[X + 2, X - 2]

# Generate some new outlier observations.
X_outliers = np.random.uniform(low=-4, high=4, size=(20, 2))
```

```
# Instantiate the classifier and fit the model for novelty detection.
clf = LocalOutlierFactor(n_neighbors=20, novelty=True, contamination="auto")
clf.fit(X_train)
```

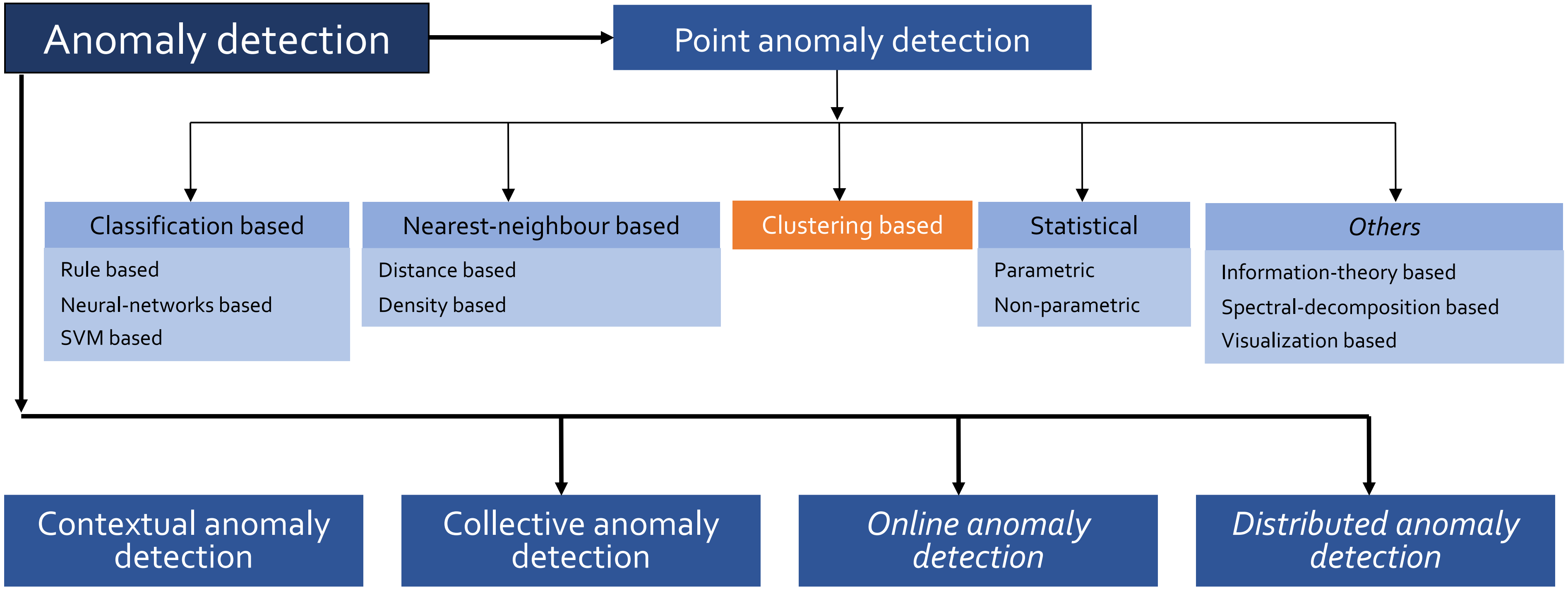


See 8_AD_LOF.ipynb
for the full code for this example.

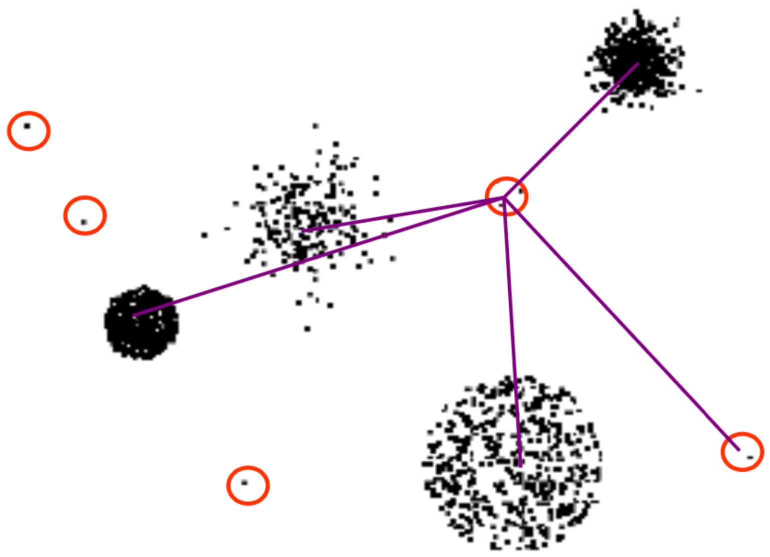
Taxonomy of techniques

An overview of anomaly detection approaches

- In the remainder of this class and next week, we will look at the following **methods of anomaly detection** (following Chandola et al. (2008) – Anomaly Detection: A Survey; see reading materials):



Clustering-based techniques
General concepts



- The key assumption is that **normal data instances belong to large and dense clusters.** Anomalies do not belong to any clusters.

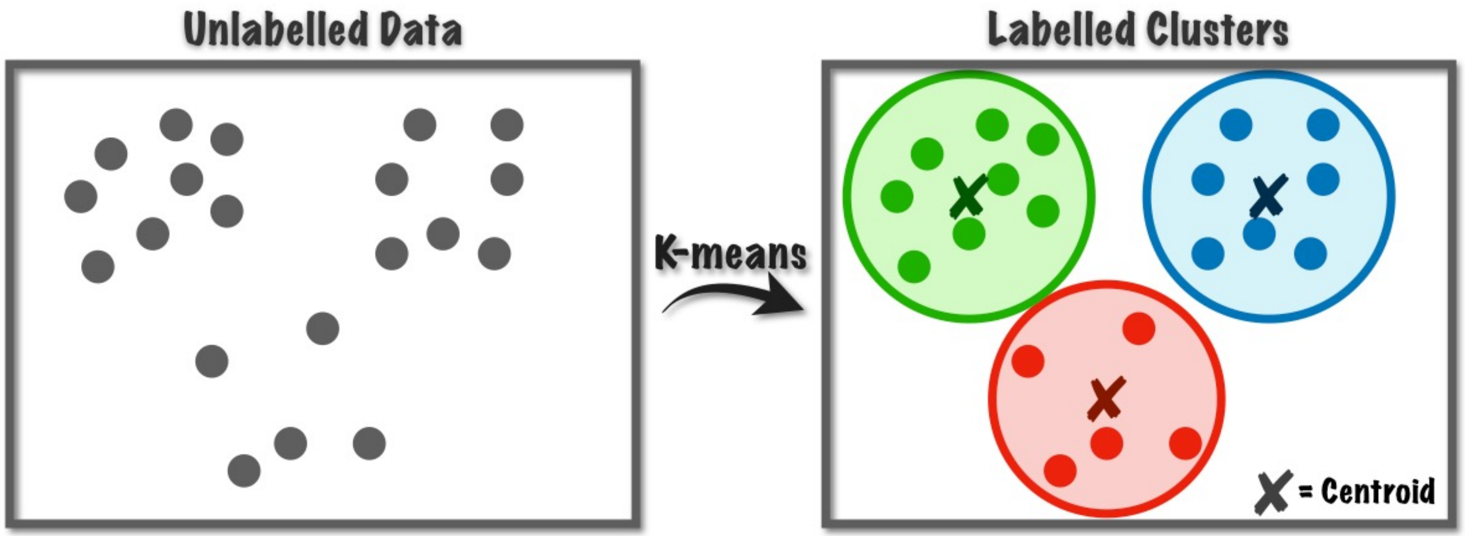
- The **general approach** follows these steps:
  - Cluster data into a **finite number of clusters with different densities**.
  - **Analyse each data instance** with respect to their closest clusters.
  - **Anomalous instances** are then those data instances that
    - do not fit into any cluster (residuals from clustering).
    - are located in small clusters.
    - are located in low-density clusters.
    - are far from other points within the same cluster.

If candidate outliers are far from all other cluster points, they are true outliers.

Clustering-based techniques
General concepts

- To assess if candidate outliers are true outliers, we have to determine the degree to which a data **object belongs to a cluster**.

- A baseline can be obtained through proto-type approaches like **k-means clustering**, where we evaluate distances of points to the cluster centre.



- For **variable densities**, we can use the **relative distance** as a measure.

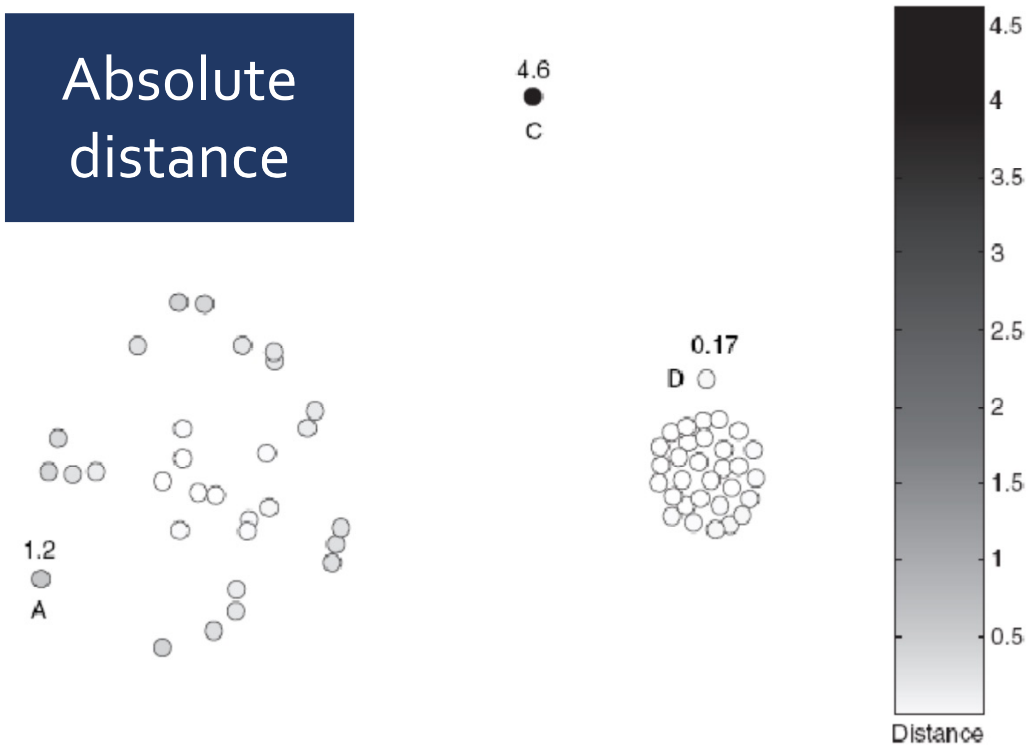
- Similar approaches exist for density- and connec-tivity-based clustering.

Clustering-based techniques

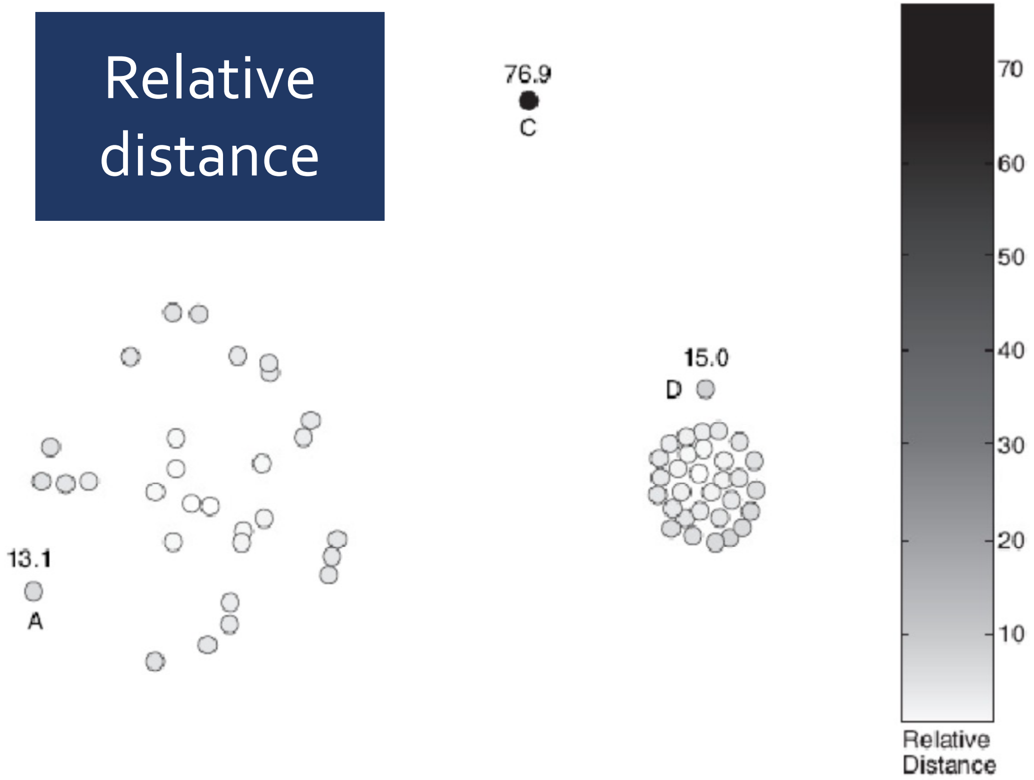
Examples

- Compare impact of absolute vs relative distance to nearest centroids:

Absolute distance



Relative distance



Clustering-based techniques
Pros and cons

ADVANTAGES

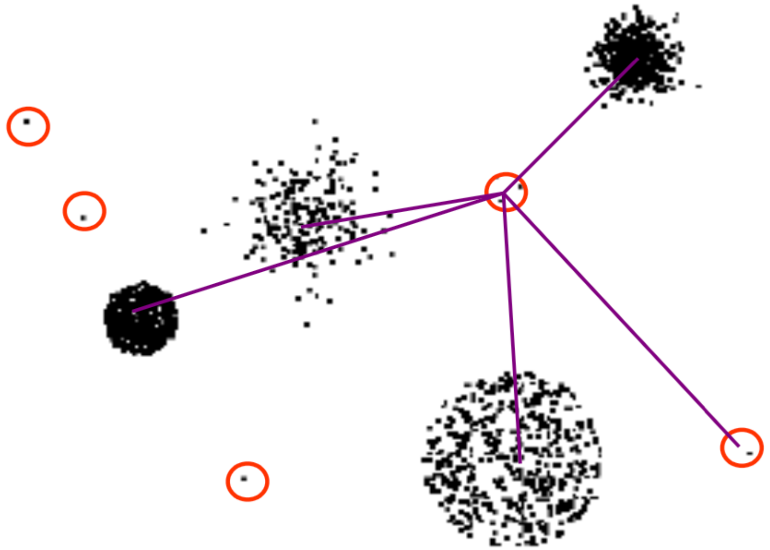
- Extend concept of outliers from single points to groups of objects.
- No labels needed to perform unsupervised anomaly detection.
- Methods are relatively easy to implement and interpret.
- There are a lot of existing clustering algorithms out there that can be readily exchanged.

DISADVANTAGES

- Sensitive to cluster number chosen.
- The presence of outliers affects the initial formation of clusters.
- In absence of natural clustering in a dataset, the technique may fail.
- Distance-based methods suffer from the curse of dimensionality.
- They are often computationally expensive (but using tree-based approaches can alleviate this).

Clustering-based techniques

Overcoming the issue of outliers



- To mitigate the problem that outliers affects the nature of our initial clusters, there are two common approaches.

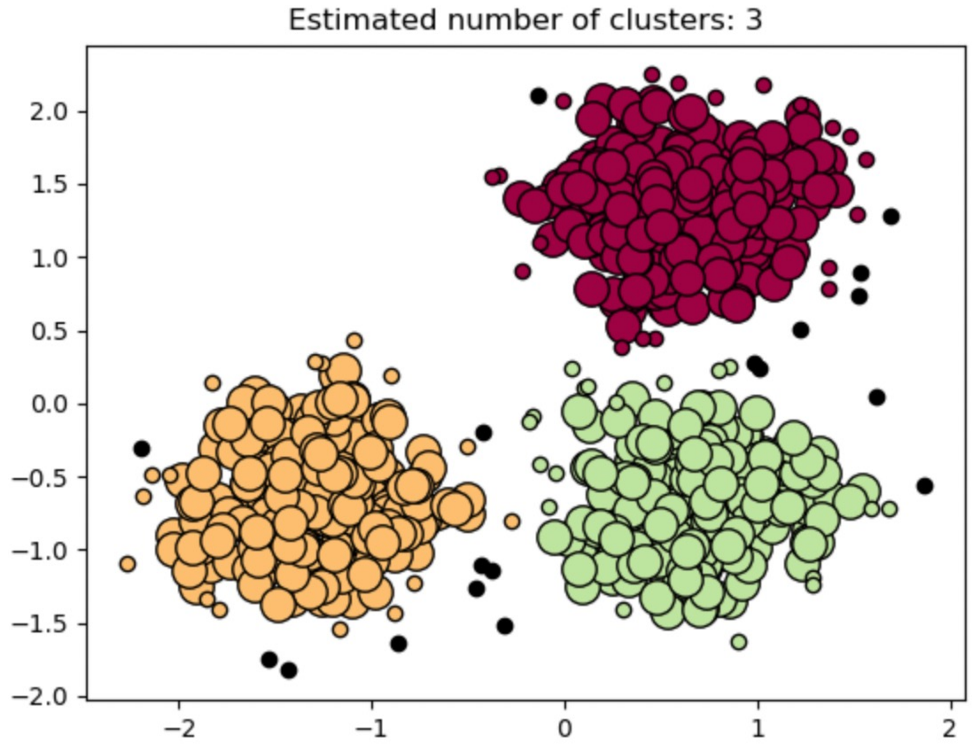
- We can eliminate certain objects from our dataset to improve the objective function. To do so, we would:
    - Form an initial set of clusters.
    - Remove those objects that most improve the objective function.
    - Repeat these steps until a desired outcome is obtained.

- We can discard small clusters located far away from other clusters. To do so, we require an understanding of “small” and “far”.

Clustering-based techniques

An example: DBSCAN

- A popular **density-based clustering technique** that has been successfully applied to a variety of problems is DBSCAN (density-based spatial clustering of applications with noise).
- It clusters data points based on continuous **regions of high point density** & determines ideal number of clusters. Outliers remain without a cluster and are easily spotted.



In contrast to k-means, not all points are assigned to a cluster, and we are not required to declare the number of clusters.

We need to set (1) the minimum number of data points required to make a cluster and (2) the allowed distance between two points to assign them to the same cluster.

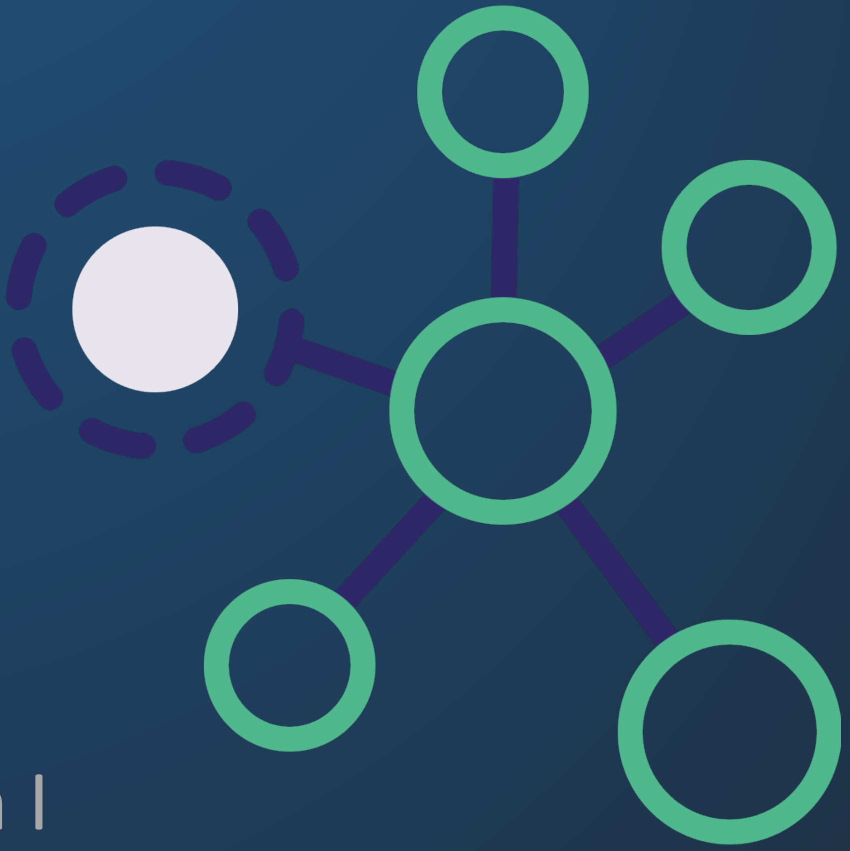
Introduction

Key questions

Applications

Techniques for anomaly detection I

Summary



Summary

- Anomaly detection can **detect critical information** in data.
- Anomaly detection **applies to various domains**. Anomalies and outliers are often the piece of information of greatest interest.
- The nature of the anomaly detection problem is dependent on the application domain. We, therefore, need different techniques to solve a particular problem formulation.
- We have introduced classification based, nearest-neighbour based and clustering based methods.