Machine Learning in High Dimension
IA317
Anomaly Detection

Thomas Bonald

2023 – 2024

TELECOM Paris
IP PARIS

Context

Anomaly

Deviation from the **expected** behavior

Various sources of anomaly:

- Errors
- Frauds
- Failures
- Attacks
- Specific events

How to detect **anomalies / outliers** without **supervision**?

Outline

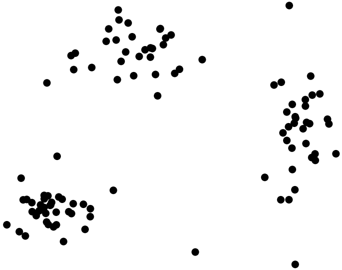
- **Algorithms**

Isolation metrics	Nearest neighbors
Isolation tables	Locally sensitive hashing
Isolation forests	Ensemble methods

- **Metrics**

Isolation metrics

Idea: Isolated samples are likely outliers

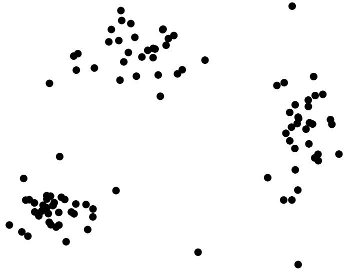


Local outlier factor

Definition

$$\mathrm{LOF}(i) = \frac{\text{Local density of } k \text{ NN of } i}{\text{Local density of } i}$$

$$\mathrm{LOF} >> 1 \;\Leftrightarrow\; \text{potential outlier}$$

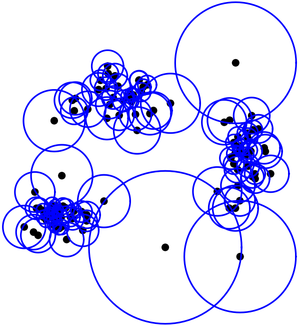


Ball radius

The **local density** around sample i can be estimated as:

$$\boxed{\frac{1}{r(i)}}$$

where $r(i)$ is the distance to the k-th nearest neighbor (ball radius)



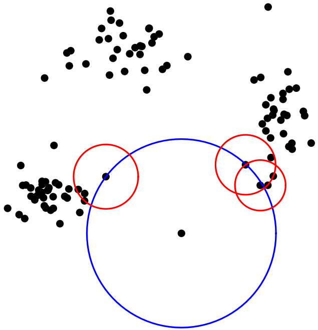
Example: Balls formed by the nearest neighbors ($k = 3$)

Simple local outlier factor

Let $N(i)$ be the k nearest neighbors of sample i (excluding itself)

Definition

$$\text{Simple-LOF}(i) = \frac{\text{Local density of } k \text{ NN of } i}{\text{Local density of } i} = \frac{1}{k} \sum_{j \in N(i)} \frac{r(i)}{r(j)}$$



Example: MNIST

$X \in \{0, \ldots, 255\}^{n \times d}$
$n = 10,000$ samples
$d = 28 \times 28 = 784$

Simple-LOF ($k = 10$, cosine similarity)

Random samples

Example: MNIST

$X \in \{0, \dots, 255\}^{n \times d}$
$n = 10,000$ samples
$d = 28 \times 28 = 784$

Simple-LOF ($k = 10$, cosine similarity)

Random samples

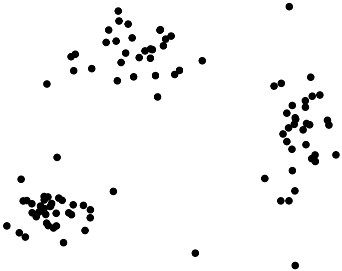
Outliers

Another measure of local density

The **local density** around sample i can be estimated as:

$$\boxed{\frac{1}{R(i)}}$$

where $R(i)$ is the **average reachibility distance** to the k nearest neighbors

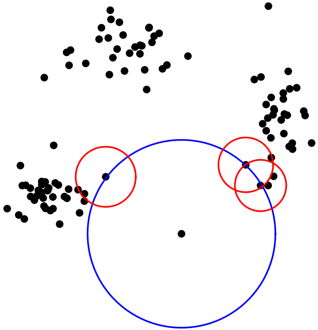


Local outlier factor

Let $N(i)$ be the k nearest neighbors of sample i (excluding itself)

Definition

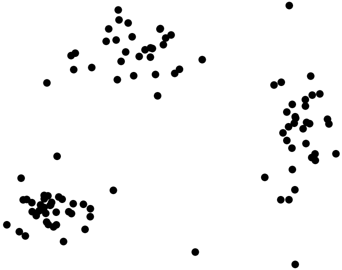
$$\mathrm{LOF}(i) = \frac{\text{Local density of } k \text{ NN of } i}{\text{Local density of } i} = \frac{1}{k}\sum_{j \in N(i)} \frac{R(i)}{R(j)}$$



Reachibility distance

The **reachibility distance** of i from j is defined by:

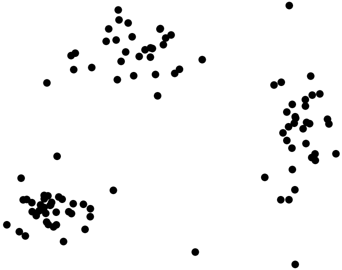
$$\boxed{r(i,j) = \max(d(i,j), r(j))}$$



Reachibility distance

The **reachibility distance** of i from j is defined by:

$$\boxed{r(i,j) = \max(d(i,j), r(j))}$$



Note: Each k-NN of j has the reachibility distance $r(j)$ from j

Warning: The reachibility distance is **not** symmetric!

Example: MNIST

$X \in \{0, \ldots, 255\}^{n \times d}$
$n = 10,000$ samples
$d = 28 \times 28 = 784$

Outliers ($k = 10$, cosine similarity)

Simple LOF

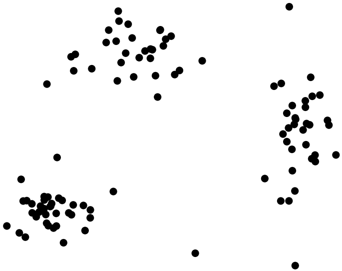
LOF

Outline

- **Algorithms**
  Isolation metrics — Nearest neighbors
  Isolation tables — Locally sensitive hashing
  Isolation forests — Ensemble methods
- **Metrics**

Isolation tables

Idea: Isolated samples tend to have **specific signatures** by locally sensitive hashing



Hashing isolation score

Data $x_1, \ldots, x_n \in \mathbb{R}^d$

Given some **locally sensitive hash** functions $h_1, \ldots, h_L$ chosen uniformly at random, we build L hash tables:

$$H_1 : s \to \text{set of samples } i \text{ such that } h_1(x_i) = s$$
$$H_2 : s \to \text{set of samples } i \text{ such that } h_2(x_i) = s$$
$$\vdots$$
$$H_L : s \to \text{set of samples } i \text{ such that } h_L(x_i) = s$$

Definition

$$\mathrm{HIS}(i) = \frac{1}{L} \sum_{l=1}^{L} \frac{1}{|\underbrace{\{j : h_l(x_j) = h_l(x_i)\}}_{\text{bucket of sample } i}|} \quad \in [0, 1]$$

Example: MNIST

$X \in \{0, \ldots, 255\}^{n \times d}$
$n = 10,000$ samples
$d = 28 \times 28 = 784$

Outliers

Simple LOF

HIS (random projection)

Outline

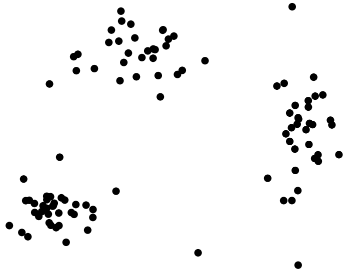
- **Algorithms**

Isolation metrics	Nearest neighbors
Isolation tables	Locally sensitive hashing
Isolation forests	Ensemble methods

- **Metrics**

Isolation forests

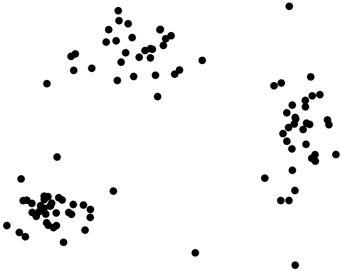
Idea: Isolated samples are **easily separable** by random splits



Extra Random Tree

Algorithm

Recursively split each set of samples at random (random feature, random threshold) until each sample is **isolated** or has the **same value** as all other samples



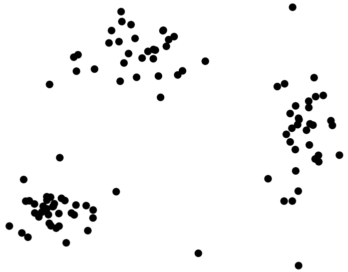
Isolation forest

Parameters: Number of trees N, sampling size $s < n$

Training

For $t = 1, \dots, N$,

- sample s data points without replacement
- build an Extra Random Tree using these s samples



Isolation forests

Evaluation

Evaluate the **average depth** of each sample i over the N trees:

$$D(i) = \frac{1}{N}\sum_{t=1}^{N} \text{depth of } i \text{ in tree } t$$

$$\text{Low depth} \quad \Leftrightarrow \quad \text{potential outlier}$$

Anomaly score

Definition

For each sample i,

$$\boxed{S(i) = 2^{-\frac{D(i)}{D}}} \quad \in [0, 1]$$

where D is the average depth of unsuccessful searches in a **random binary tree** with s values

$S(i)$ close to 1 $\Leftrightarrow$ potential outlier

Example: MNIST

$X \in \{0, \ldots, 255\}^{n \times d}$
$n = 10,000$ samples
$d = 28 \times 28 = 784$

Outliers

Simple LOF

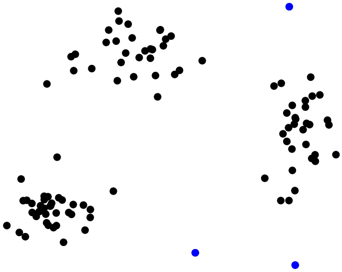
Isolation forest

Outline

- **Algorithms**
  Isolation metrics — Nearest neighbors
  Isolation tables — Locally sensitive hashing
  Isolation forests — Ensemble methods
- **Metrics**

Metric

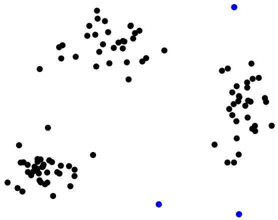
Idea: Use **annotated data** to assess the quality of anomaly detection



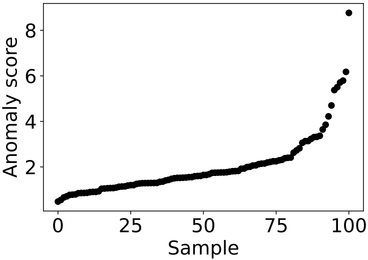
Metric

Issue: The 3 algorithms (isolation metrics / tables / forests) provide an **anomaly score** S, not a binary decision

Samples



Scores



Binary classification

Let $y_1, \ldots, y_n \in \{0, 1\}$ be the **true labels** ($1 =$ anomaly)
Let $\hat{y}_1 = 1_{\{S_1 \geq t\}}, \ldots, \hat{y}_n = 1_{\{S_n \geq t\}} \in \{0, 1\}$ be the **predicted labels** at threshold t

Binary classification

Let $y_1, \ldots, y_n \in \{0, 1\}$ be the **true labels** ($1 =$ anomaly)
Let $\hat{y}_1 = 1_{\{S_1 \geq t\}}, \ldots, \hat{y}_n = 1_{\{S_n \geq t\}} \in \{0, 1\}$ be the **predicted labels** at threshold t

True positive rate (recall)

$$\boxed{\frac{\# \text{ True positive}}{\# \text{ Positive}} = \frac{\sum_i 1_{\{\hat{y}_i = 1, y_i = 1\}}}{\sum_i 1_{\{y_i = 1\}}}}$$

Binary classification

Let $y_1, \ldots, y_n \in \{0, 1\}$ be the **true labels** ($1 =$ anomaly)
Let $\hat{y}_1 = 1_{\{S_1 \geq t\}}, \ldots, \hat{y}_n = 1_{\{S_n \geq t\}} \in \{0, 1\}$ be the **predicted labels** at threshold t

True positive rate (recall)

$$\boxed{\frac{\#\text{ True positive}}{\#\text{ Positive}} = \frac{\sum_i 1_{\{\hat{y}_i = 1, y_i = 1\}}}{\sum_i 1_{\{y_i = 1\}}}}$$

False positive rate (fall out)

$$\boxed{\frac{\#\text{ False positive}}{\#\text{ Negative}} = \frac{\sum_i 1_{\{\hat{y}_i = 1, y_i = 0\}}}{\sum_i 1_{\{y_i = 0\}}}}$$

A probabilistic view

Let $y \in \{0, 1\}$ be the **true label** of a sample ($1 =$ anomaly)
Let S be its score, and $\hat{y} = 1_{\{S \geq t\}}$ the predicted label

A probabilistic view

Let $y \in \{0, 1\}$ be the **true label** of a sample ($1 =$ anomaly)
Let S be its score, and $\hat{y} = 1_{\{S \geq t\}}$ the predicted label

True positive rate (recall)

$$\boxed{R(t) = \mathrm{P}(S \geq t |\ y = 1)}$$

A probabilistic view

Let $y \in \{0, 1\}$ be the **true label** of a sample (1 = anomaly)
Let S be its score, and $\hat{y} = 1_{\{S \geq t\}}$ the predicted label

True positive rate (recall)

$$\boxed{R(t) = \mathrm{P}(S \geq t |\ y = 1)}$$

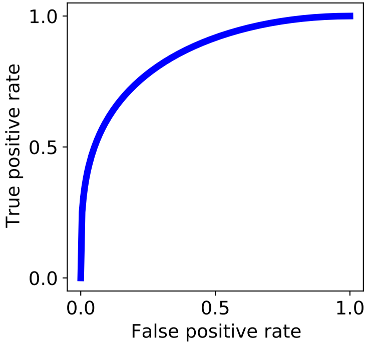
False positive rate (fall out)

$$\boxed{F(t) = \mathrm{P}(S \geq t |\ y = 0)}$$

The ROC[1] curve

Definition

Plot of **Recall** against **Fall out** when the threshold t goes from $-\infty$ to $+\infty$



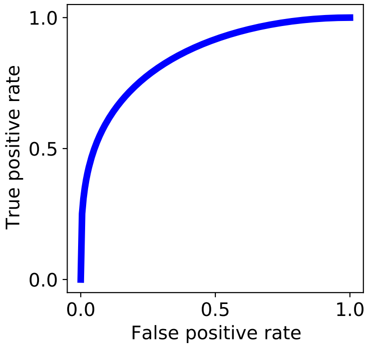
[1]Receiver Operating Characteristic

The ROC AUC score

Definition

The ROC AUC (Area Under Curve) score is:

$$\mathsf{AUC} = \int_0^1 \mathsf{ROC}(u)\mathrm{d}u \in [0,1]$$



Probabilistic interpretation of the ROC AUC score

Proposition

The ROC AUC score is the probability that the score is consistent for a **random pair** of samples i, j with distinct labels:

$$\mathrm{AUC} = P(S_i > S_j | y_i = 1, y_j = 0)$$

Probabilistic interpretation of the ROC AUC score

Proposition

The ROC AUC score is the probability that the score is consistent for a **random pair** of samples i, j with distinct labels:

$$\text{AUC} = P(S_i > S_j | y_i = 1, y_j = 0)$$

Notes:

- We have $\text{AUC} = \frac{1}{2}$ for a **random** prediction
- Alternative metric = **Mean Average Precision**
  $\rightarrow$ AUC of Precision against Recall

Summary

Anomaly detection

Algorithms

- Isolation metrics Nearest neighbors
- Isolation tables Locally sensitive hashing
- Isolation forests Ensemble method

The ROC AUC score, a **quality metric** for annotated data