Outlier detection

Edouard Pauwels

M2-MAT SID

Outlier /anomaly detection problems

- Data cleaning
- Attack / intrusion detection (IT security)
- Fraud detection (banking, insurance).
- Medical diagnosis and monitoring of unusual symptoms
- Industrial monitoring, damage detection, predictive maintenance
- Image processing, video surveillance
- Text mining (news detection)
- Sensor networks, fault / attack
- etc. . .

What is an outlier ?

Often used interchangably with *anomaly*

Hawkins (1980) :

An observation that deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism.

Johnson (1992) :

An observation in a data set which appears to be inconsistent with the remainder of that set of data.

Barnett and Lewis (1994) :

An observation that appears to deviate markedly from other members of the sample in which it occurs

What is an outlier ?

Often used interchangably with *anomaly*

Hawkins (1980) :

An observation that deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism.

Johnson (1992) :

An observation in a data set which appears to be inconsistent with the remainder of that set of data.

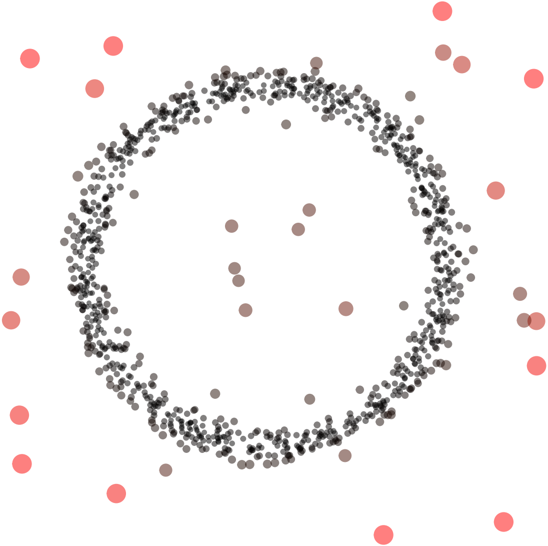
Barnett and Lewis (1994) :

An observation that appears to deviate markedly from other members of the sample in which it occurs

Main ideas :

- Need a reference distribution, sample.
- Outliers and non outliers are mixed in the same sample.
- The proportion of outliers is small

Illustration



Different paradigms

Input space : $\mathcal{X} = \mathbb{R}^p$, $n \in \mathbb{N}$, $S_n = (x_i)_{i=1}^n$, $x_i \in \mathcal{X}$, $i = 1, \dots, n$.

Different paradigms

Input space : $\mathcal{X} = \mathbb{R}^p$, $n \in \mathbb{N}$, $S_n = (x_i)_{i=1}^n$, $x_i \in \mathcal{X}$, $i = 1, \ldots, n$.

- **Supervised :**
  Binary classification with labels $(y_i)_{i=1}^n$ describing the status (anomaly or not).
  Unballanced classes.

Different paradigms

Input space : $\mathcal{X} = \mathbb{R}^p$, $n \in \mathbb{N}$, $S_n = (x_i)_{i=1}^n$, $x_i \in \mathcal{X}$, $i = 1, \ldots, n$.

- **Supervised :**
  Binary classification with labels $(y_i)_{i=1}^n$ describing the status (anomaly or not).
  Unballanced classes.
- **Semi-supervised :** knowledge only of the normal class
  - S_n consists only of points which are not anomalies.
  - PU learning : S_n has some point labeled as normal and the rest could be eigher normal or abnormal.

Different paradigms

Input space : $\mathcal{X} = \mathbb{R}^p$, $n \in \mathbb{N}$, $S_n = (x_i)_{i=1}^n$, $x_i \in \mathcal{X}$, $i = 1, \ldots, n$.

- **Supervised :**
  Binary classification with labels $(y_i)_{i=1}^n$ describing the status (anomaly or not).
  Unballanced classes.
- **Semi-supervised :** knowledge only of the normal class
  - S_n consists only of points which are not anomalies.
  - PU learning : S_n has some point labeled as normal and the rest could be eigher normal or abnormal.
- **Unsupervised :**
  S_n consists in a mixture of normal and a few abnormal examples, we wish to detect them automatically

We will consider only unsupervised anomaly detection

Input space : $\mathcal{X} = \mathbb{R}^p$, $n \in \mathbb{N}$, $S_n = (x_i)_{i=1}^n$, $x_i \in \mathcal{X}$, $i = 1, \ldots, n$.

- **Supervised :**
  Binary classification with labels $(y_i)_{i=1}^n$ describing the status (anomaly or not).
  Unballanced classes.
- **Semi-supervised :** knowledge only of the normal class
  - S_n consists only of points which are not anomalies.
  - PU learning : S_n has some point labeled as normal and the rest could be eigher normal or abnormal.
- **Unsupervised :**
  S_n consists in a mixture of normal and a few abnormal examples, we wish to detect them automatically

We will consider only unsupervised anomaly detection
We will still have labels : evaluate methods performances, only used for test purposes.

Outline

1. What is an outlier ?

2. Score based detection

3. Univartiate methods

4. Multivariate approaches

5. Practical

Score based approaches

Setting :

Training data : $S_n = (x_i)_{1 \leq i \leq n}$,

Goal : predict $y \in \{0, 1\}$ (anomaly or not).

Scoring : Compute a scoring function $s_n \colon \mathcal{X} \mapsto \mathbb{R}$. $s_n \colon x \mapsto h(x, x_1, \dots, x_n)$.

Ground truth : $(y_i)_{1 \leq i \leq n}$, 0 or 1 (outlier or not), not used for training.

Setting :

Training data : $S_n = (x_i)_{1 \leq i \leq n}$,

Goal : predict $y \in \{0, 1\}$ (anomaly or not).

Scoring : Compute a scoring function $s_n \colon \mathcal{X} \mapsto \mathbb{R}$. $s_n \colon x \mapsto h(x, x_1, \dots, x_n)$.

Ground truth : $(y_i)_{1 \leq i \leq n}$, 0 or 1 (outlier or not), not used for training.

Evaluation : Need an annotated sample of outliers.

In sample outlier detection : compare $s(x_i)$ and y_i, $i = 1, \dots, n$.

Out of sample intrusion / change detection : compare score and class on unseen data $s_n(\tilde{x})$, $\tilde{y}$.

Score based approaches

Setting :

Training data : $S_n = (x_i)_{1 \le i \le n}$,

Goal : predict $y \in \{0, 1\}$ (anomaly or not).

Scoring : Compute a scoring function $s_n : \mathcal{X} \mapsto \mathbb{R}$. $s_n : x \mapsto h(x, x_1, \dots, x_n)$.

Ground truth : $(y_i)_{1 \le i \le n}$, 0 or 1 (outlier or not), not used for training.

Evaluation : Need an annotated sample of outliers.

In sample outlier detection : compare $s(x_i)$ and y_i, $i = 1, \dots, n$.

Out of sample intrusion / change detection : compare score and class on unseen data $s_n(\tilde{x})$, $\tilde{y}$.

We will focus on *in sample* detection : fix a threshold $\bar{s} \in \mathbb{R}$ and predict for $i = 1, \dots, n$,

Anomaly if $s_n(x_i) \ge \bar{s}$.

Normal otherwise.

Compare prediction and ground truth $(y_i)_{1 \le i \le n}$

Score based approaches

Setting :

Training data : $S_n = (x_i)_{1 \leq i \leq n}$,

Goal : predict $y \in \{0, 1\}$ (anomaly or not).

Scoring : Compute a scoring function $s_n \colon \mathcal{X} \mapsto \mathbb{R}$. $s_n \colon x \mapsto h(x, x_1, \ldots, x_n)$.

Ground truth : $(y_i)_{1 \leq i \leq n}$, 0 or 1 (outlier or not), not used for training.

Evaluation : Need an annotated sample of outliers.

In sample outlier detection : compare $s(x_i)$ and y_i, $i = 1, \ldots, n$.

Out of sample intrusion / change detection : compare score and class on unseen data $s_n(\tilde{x})$, $\tilde{y}$.

We will focus on *in sample* detection : fix a threshold $\bar{s} \in \mathbb{R}$ and predict for $i = 1, \ldots, n$,

Anomaly if $s_n(x_i) \geq \bar{s}$.

Normal otherwise.

Compare prediction and ground truth $(y_i)_{1 \leq i \leq n}$

Remark : All the methods which we will see can be used for out of sample anomaly detection. The evaluation is then close to what is done in supervised learning settings.

Evaluation metrics : precision and recall

		Reality		
		Abnormal	Normal	Total
Prediction	Abnormal	TP	FP	$TP + FP$
	Normal	FN	TN	$FN + TN$
	Total	$TP + FN$	$FP + TN$	n

Evaluation metrics : precision and recall

		Reality		
		Abnormal	Normal	Total
Prediction	Abnormal	TP	FP	$TP + FP$
	Normal	FN	TN	$FN + TN$
	Total	$TP + FN$	$FP + TN$	n

Precision $\frac{TP}{TP+FP} = \frac{TP}{|\text{predicted anomalies}|}$.

Recall $\frac{TP}{TP+FN} = \frac{TP}{|\text{real anomalies}|}$.

F1 score $\text{F1} = 2 \times \frac{Pr \times Rec}{Pr + Rec}$

In fact : Prediction($\bar{s}$) depends on chosen threshold,

Evaluation metrics : precision and recall

In fact : Prediction($\bar{s}$) depends on chosen threshold, $TP(\bar{s})$, $FP(\bar{s})$, $FN(\bar{s})$, $TN(\bar{s})$

Evaluation metrics : precision and recall

In fact : Prediction($\bar{s}$) depends on chosen threshold, $TP(\bar{s})$, $FP(\bar{s})$, $FN(\bar{s})$, $TN(\bar{s})$

		Reality: Abnormal	Reality: Normal	Total
Prediction($\bar{s}$)	Abnormal	$TP(\bar{s})$	$FP(\bar{s})$	$TP(\bar{s}) + FP(\bar{s})$
	Normal	$FN(\bar{s})$	$TN(\bar{s})$	$FN(\bar{s}) + TN(\bar{s})$
	Total	$TP(\bar{s}) + FN(\bar{s})$	$FP(\bar{s}) + TN(\bar{s})$	n

In fact : Prediction($\bar{s}$) depends on chosen threshold, $TP(\bar{s})$, $FP(\bar{s})$, $FN(\bar{s})$, $TN(\bar{s})$

		Reality: Abnormal	Reality: Normal	Total
Prediction($\bar{s}$)	Abnormal	$TP(\bar{s})$	$FP(\bar{s})$	$TP(\bar{s}) + FP(\bar{s})$
	Normal	$FN(\bar{s})$	$TN(\bar{s})$	$FN(\bar{s}) + TN(\bar{s})$
	Total	$TP(\bar{s}) + FN(\bar{s})$	$FP(\bar{s}) + TN(\bar{s})$	n

Precision($\bar{s}$) : $\frac{TP(\bar{s})}{TP(\bar{s})+FP(\bar{s})} = \frac{TP(\bar{s})}{|\text{predicted anomalies}(\bar{s})|}$.

Recall($\bar{s}$) : $\frac{TP(\bar{s})}{TP(\bar{s})+FN(\bar{s})} = \frac{TP(\bar{s})}{|\text{real anomalies}(\bar{s})|}$ $(= \mathbf{TPR}(\bar{s}))$.

FPR($\bar{s}$) : $\frac{FP(\bar{s})}{FP(\bar{s})+TN(\bar{s})} = \frac{FP(\bar{s})}{|\text{real normal}(\bar{s})|}$.

Evaluation metrics : F1 score

Score dependant evaluation :

- Choose $\bar{s}$.
- Compute F1-score $2 \times \frac{Pr(\bar{s}) \times Rec(\bar{s})}{Pr(\bar{s}) + Rec(\bar{s})}$.

Evaluation metrics : F1 score

Score dependant evaluation :

- Choose $\bar{s}$.
- Compute F1-score $2 \times \frac{Pr(\bar{s}) \times Rec(\bar{s})}{Pr(\bar{s}) + Rec(\bar{s})}$.

Question :

Evaluation metrics : F1 score

Score dependant evaluation :

- Choose $\bar{s}$.
- Compute F1-score $2 \times \frac{Pr(\bar{s}) \times Rec(\bar{s})}{Pr(\bar{s}) + Rec(\bar{s})}$.

Question : How to choose $\bar{s}$?

Score dependant evaluation :

- Choose $\bar{s}$.
- Compute F1-score $2 \times \frac{Pr(\bar{s}) \times Rec(\bar{s})}{Pr(\bar{s}) + Rec(\bar{s})}$.

Question : How to choose $\bar{s}$?

- No universal rule, depends on the regime considered.

Evaluation metrics : F1 score

Score dependant evaluation :

- Choose $\bar{s}$.
- Compute F1-score $2 \times \frac{Pr(\bar{s}) \times Rec(\bar{s})}{Pr(\bar{s}) + Rec(\bar{s})}$.

Question : How to choose $\bar{s}$?

- No universal rule, depends on the regime considered.
- Ex : known proportion of outliers.
  - You know that 10% of the data are outliers.
  - You computed $s(x_i)$, $i = 1, \dots, n$.

Score dependant evaluation :

- Choose $\bar{s}$.
- Compute F1-score $2 \times \frac{Pr(\bar{s}) \times Rec(\bar{s})}{Pr(\bar{s}) + Rec(\bar{s})}$.

Question : How to choose $\bar{s}$?

- No universal rule, depends on the regime considered.
- Ex : known proportion of outliers.
  - You know that 10% of the data are outliers.
  - You computed $s(x_i)$, $i = 1, \dots, n$.
- Ex : known outlier free dataset.
  - You know that $\tilde{x}_1, \dots, \tilde{x}_m$ which are not outliers.
  - You computed $s(\tilde{x}_i)$, $i = 1, \dots, m$.

Evaluation metrics : F1 score

Score dependant evaluation :

- Choose $\bar{s}$.
- Compute F1-score $2 \times \frac{Pr(\bar{s}) \times Rec(\bar{s})}{Pr(\bar{s}) + Rec(\bar{s})}$.

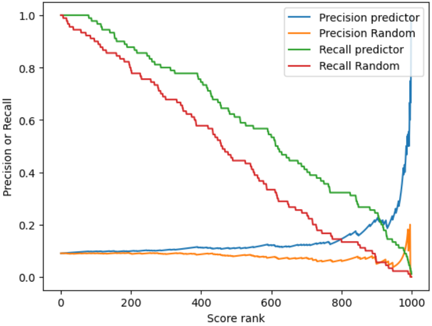
Question : How to choose $\bar{s}$?

- No universal rule, depends on the regime considered.
- Ex : known proportion of outliers.
  - You know that 10% of the data are outliers.
  - You computed $s(x_i)$, $i = 1, \ldots, n$.
- Ex : known outlier free dataset.
  - You know that $\tilde{x}_1, \ldots, \tilde{x}_m$ which are not outliers.
  - You computed $s(\tilde{x}_i)$, $i = 1, \ldots, m$.
- Ex : semi-supervised approach.
  - Choose $\bar{s}$ which has the largest F1-score.
  - Evaluate using cross validation.

Score independant evaluation : sort examples by score (degree of outlyingness)

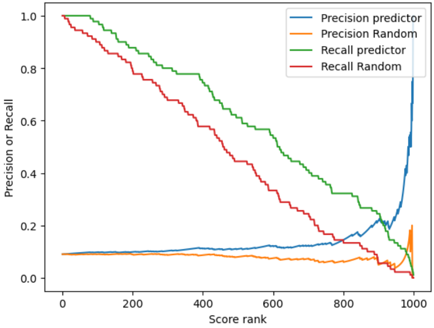
Score independant evaluation : sort examples by score (degree of outlyingness)

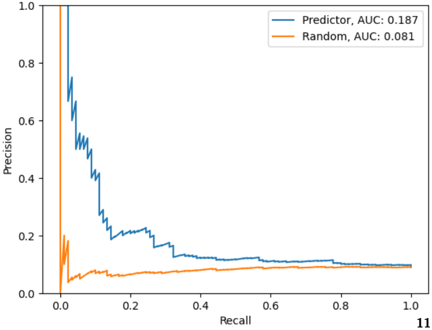
- Plot precision and recall as function of $\bar{s}$.



Score independant evaluation : sort examples by score (degree of outlyingness)

- Plot precision and recall as function of $\bar{s}$.
- Precision recall curve : Precision($\bar{s}$) as a function of Recall($\bar{s}$) for varying $\bar{s}$.

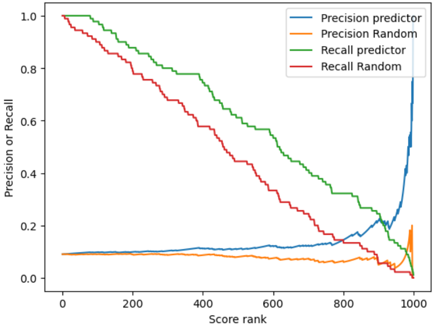


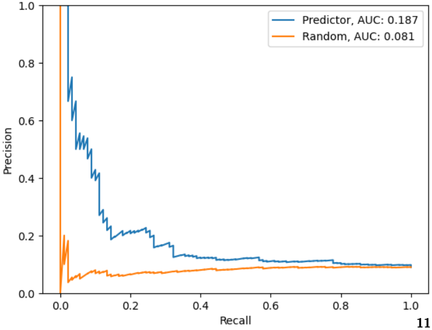


Evaluation metrics : PR curves

Score independant evaluation : sort examples by score (degree of outlyingness)

- Plot precision and recall as function of $\bar{s}$.
- Precision recall curve : Precision($\bar{s}$) as a function of Recall($\bar{s}$) for varying $\bar{s}$.
- Compute AUPR (Area Under the PR curve).





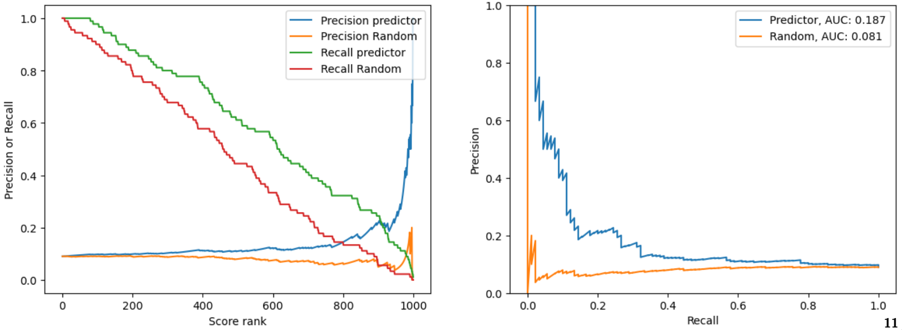
Evaluation metrics : PR curves

Score independant evaluation : sort examples by score (degree of outlyingness)

- Plot precision and recall as function of $\bar{s}$.
- Precision recall curve : Precision($\bar{s}$) as a function of Recall($\bar{s}$) for varying $\bar{s}$.
- Compute AUPR (Area Under the PR curve).

Comments :

- Compare methods ability to order by degree of outlyingness.
- Allows to compare methods without having to select $\bar{s}$
- More general but less taylored to certain regimes.
- **In any case :** $\bar{s}$ will be needed in practice.



Hyperparameters : number of neighbors, polynomial degree . . .

Scoring : Compute a scoring function

$$s_n : \mathcal{X} \mapsto \mathbb{R}$$
$$s_n : x \mapsto h(x, x_1, \ldots, x_n, params).$$

Hyperparameters : number of neighbors, polynomial degree . . .

Scoring : Compute a scoring function

$$s_n : \mathcal{X} \mapsto \mathbb{R}$$
$$s_n : x \mapsto h(x, x_1, \ldots, x_n, params).$$

Question :

Hyperparameters : number of neighbors, polynomial degree . . .

Scoring : Compute a scoring function

$$s_n : \mathcal{X} \mapsto \mathbb{R}$$
$$s_n : x \mapsto h(x, x_1, \ldots, x_n, params).$$

Question : How to tune *params* ?

Hyperparameters : number of neighbors, polynomial degree . . .

Scoring : Compute a scoring function

$$s_n : \mathcal{X} \mapsto \mathbb{R}$$
$$s_n : x \mapsto h(x, x_1, \ldots, x_n, params).$$

Question : How to tune *params* ?

- Can be score dependent or independent (F1 score or AUPR).
- Cannot use data twice : for hyper parameter tuning and for model evaluation.
- Unsupervised detection : less prone to overfitting (does not use labels for training).

Hyperparameters : number of neighbors, polynomial degree . . .

Scoring : Compute a scoring function

$$s_n : \mathcal{X} \mapsto \mathbb{R}$$
$$s_n : x \mapsto h(x, x_1, \ldots, x_n, params).$$

Question : How to tune *params* ?

- Can be score dependent or independent (F1 score or AUPR).
- Cannot use data twice : for hyper parameter tuning and for model evaluation.
- Unsupervised detection : less prone to overfitting (does not use labels for training).

Tools from supervised learning : cross validation, validation set.

Practical session

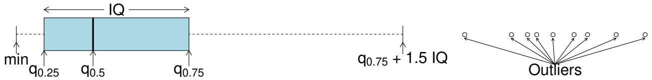
TP_PR_ROC

Outline

What could we do in the univariate case ?

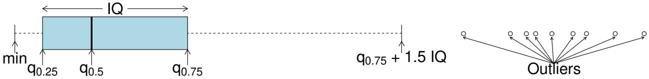
What could we do in the univariate case ?

Interquartile range :



What could we do in the univariate case ?

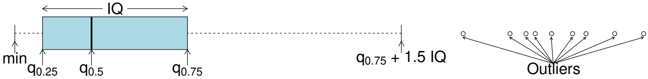
Interquartile range :



Z-score :

$$s_n : t \mapsto \frac{\left|t - \bar{x}\right|}{\sigma_x}$$

Interquartile range :



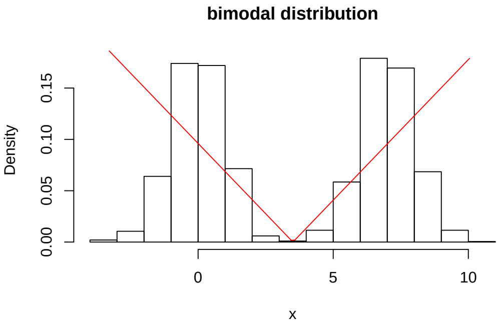
Z-score :

$$s_n : t \mapsto \frac{|t - \bar{x}|}{\sigma_x}$$

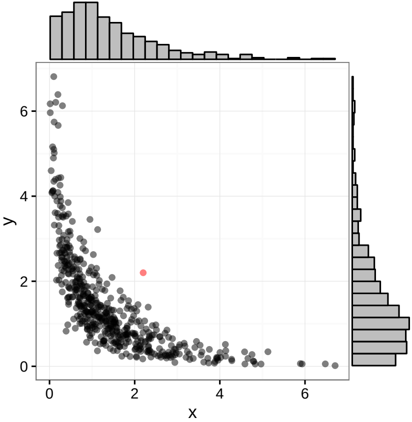
Shortcomings and limitations ?

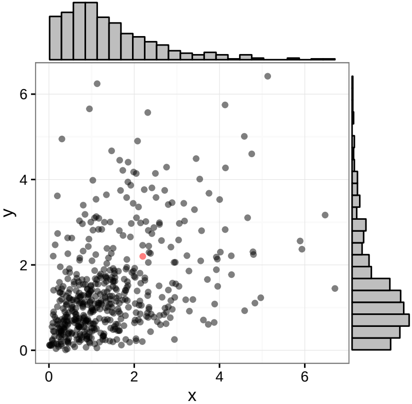
A case for more advanced methods

A bimodal distribution, Z-score in red.



Bi-variate $\neq 2 \times$ univariate



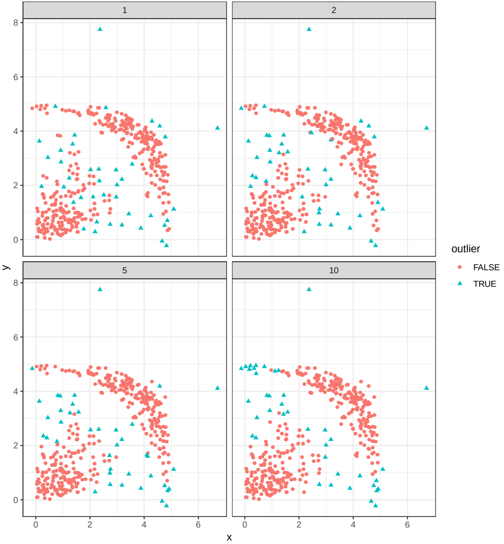


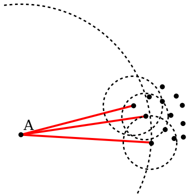
Distance to the k-th neighbor :

$$s_n\colon x \mapsto k\mathrm{dist}(x) := \mathrm{dist}(x, x_l)$$

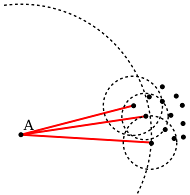
where x_l is the k-th neirest neighbor of x in $S_n = (x_i)_{i=1}^n$.

Variation of k





$N_k(x)$	k nearest neighbors of x
$\mathrm{REACH}_k(x, y) = \max\{k\mathrm{dist}(y), \mathrm{dist}(x, y)\}$	reachability
$\mathrm{LRD}_k(x) = \left(\frac{1}{k} \sum_{y \in N_k(x)} \mathrm{REACH}_k(x, y)\right)^{-1}$	Local Reachability Density
$\mathrm{LOF}_k(x) = \frac{1}{k} \sum_{y \in N_k(x)} \frac{\mathrm{LRD}_k(y)}{\mathrm{LRD}_k(x)}$	Local Outlier Factor



$$N_k(x) \qquad k \text{ nearest neighbors of } x$$

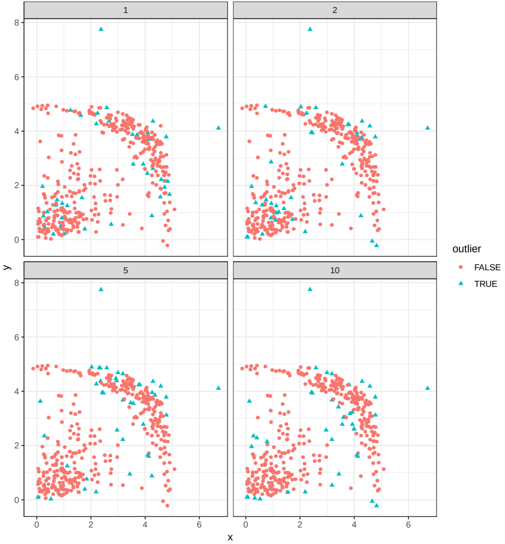
$$\mathrm{REACH}_k(x, y) = \max\{k\mathrm{dist}(y), \mathrm{dist}(x, y)\} \qquad \text{reachability}$$

$$\mathrm{LRD}_k(x) = \left(\frac{1}{k} \sum_{y \in N_k(x)} \mathrm{REACH}_k(x, y)\right)^{-1} \qquad \text{Local Reachability Density}$$

$$\mathrm{LOF}_k(x) = \frac{1}{k} \sum_{y \in N_k(x)} \frac{\mathrm{LRD}_k(y)}{\mathrm{LRD}_k(x)} \qquad \text{Local Outlier Factor}$$

$\mathrm{LOF} > 1$ implies smaller density as neighbors. The LOF is used as a score s_n.

Variation of k



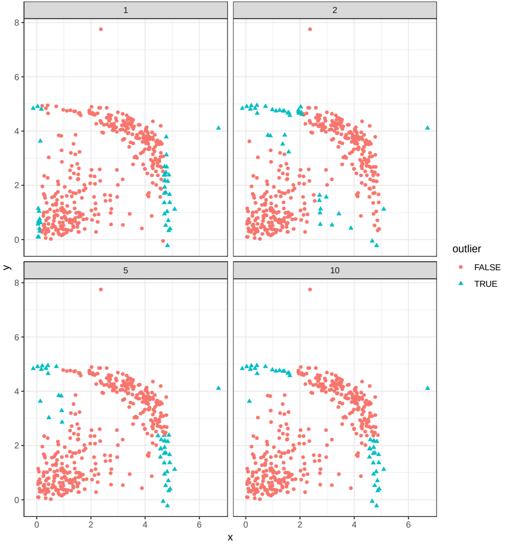
K-means detection

k clusters

- Perform clustering using
- $s_n : x \mapsto \mathrm{dist}(x, c)$, where c is the centroid the closest to x.

K-means detection

Variation of the number of clusters



Likelihood based outlier detection

Family of parametrized models density functions p_θ :

- Maximum likelihood : $\hat{\theta} \in \arg\max_\theta \sum_{i=1}^n \log(p_\theta(x_i))$.
- Score using likelihood : $s_n \colon x \mapsto p_{\hat{\theta}}(x)$.

Likelihood based outlier detection

Family of parametrized models density functions p_θ :

- Maximum likelihood : $\hat{\theta} \in \arg\max_\theta \sum_{i=1}^n \log(p_\theta(x_i))$.
- Score using likelihood : $s_n : x \mapsto p_{\hat{\theta}}(x)$.

Mahalanobis distance and Gaussian model :

Likelihood based outlier detection

Family of parametrized models density functions p_θ :

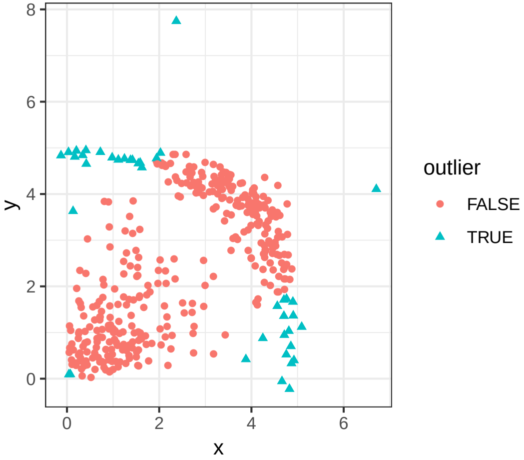
- Maximum likelihood : $\hat{\theta} \in \arg\max_\theta \sum_{i=1}^n \log(p_\theta(x_i))$.
- Score using likelihood : $s_n \colon x \mapsto p_{\hat{\theta}}(x)$.

Mahalanobis distance and Gaussian model :

$$s_n \colon x \mapsto \exp\left(-(x - \bar{x}_n)\,\Sigma_n^{-1}\,(x - \bar{x}_n)\right)$$

where $\bar{x}_n$ is the empirical mean and Σ_n is the emprirical covariance matrix.

No tuning parameter



Gaussian mixture model

Density of the form

$$p_\theta : x \mapsto \sum_{i=1}^{K} \tau_i p(x|\mu_i, \Sigma_i)$$

where $\tau_i > 0$, $\sum_i \tau_i = 1$,
$p(x|\mu_\Sigma)$ is the density of the multivariate Gaussian with mean μ and covariance Σ.

Gaussian mixture model

Density of the form

$$p_\theta : x \mapsto \sum_{i=1}^{K} \tau_i p(x|\mu_i, \Sigma_i)$$

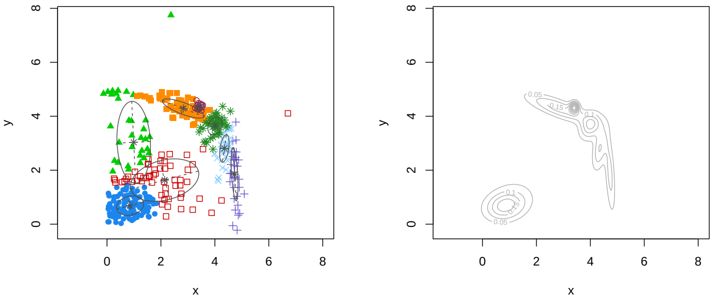
where $\tau_i > 0$, $\sum_i \tau_i = 1$,
$p(x|\mu_\Sigma)$ is the density of the multivariate Gaussian with mean μ and covariance Σ.
Maximum likelihood : using EM algorithm ($\sim$ extension of k-means).

Gaussian mixture model

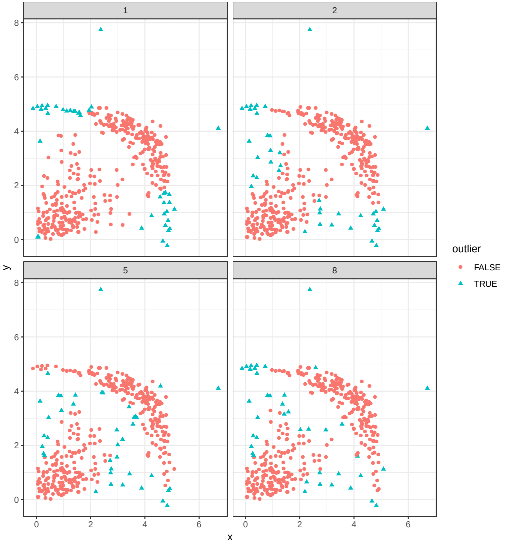
Density of the form

$$p_\theta : x \mapsto \sum_{i=1}^{K} \tau_i p(x|\mu_i, \Sigma_i)$$

where $\tau_i > 0$, $\sum_i \tau_i = 1$,
$p(x|\mu_\Sigma)$ is the density of the multivariate Gaussian with mean μ and covariance Σ.
Maximum likelihood : using EM algorithm ($\sim$ extension of k-means).



Variation of the number of clusters



Density based

Gaussian kernel with bandwidth σ

$$k(x, y) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{\|y-x\|^2}{\sigma^2}}$$

Kernel density estimator :

$$p_\sigma : x \mapsto \frac{1}{n} \sum_{i=1}^{n} k(x, x_i)$$

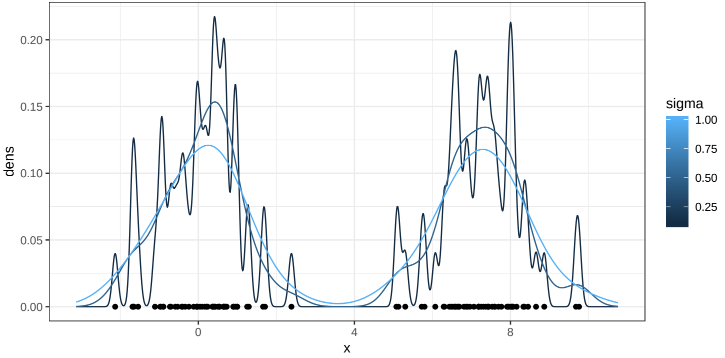
Density based

Gaussian kernel with bandwidth σ

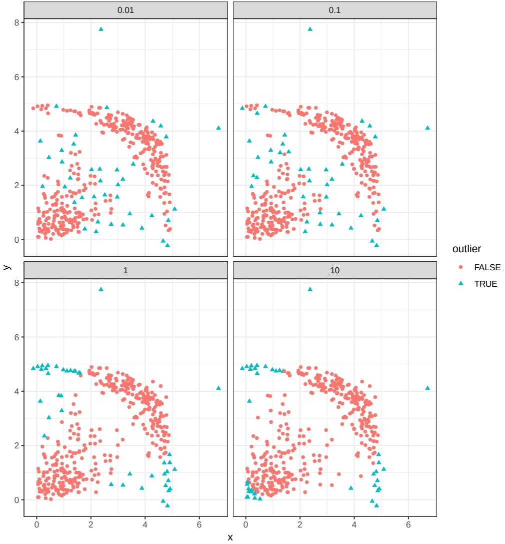
$$k(x, y) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{\|y-x\|^2}{\sigma^2}}$$

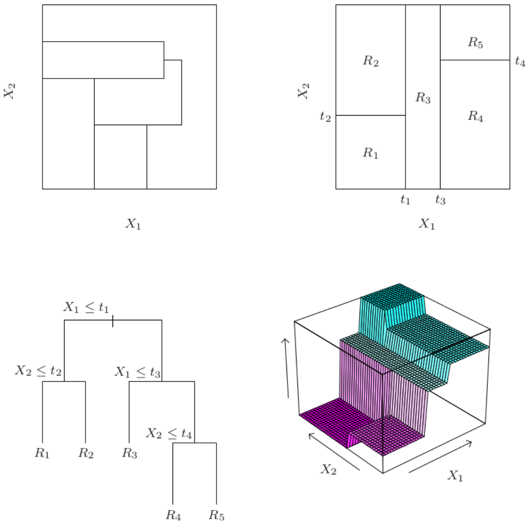
Kernel density estimator :

$$p_\sigma : x \mapsto \frac{1}{n} \sum_{i=1}^{n} k(x, x_i)$$



Variation of the bandwidth





X2
X1
R5
t4
R2
R3
R4
t2
R1
t1
t3
X2
X1
X1 ≤ t1
X2 ≤ t2
X1 ≤ t3
X2 ≤ t4
R1
R2
R3
R4
R5
X2
X1

Isolation forest

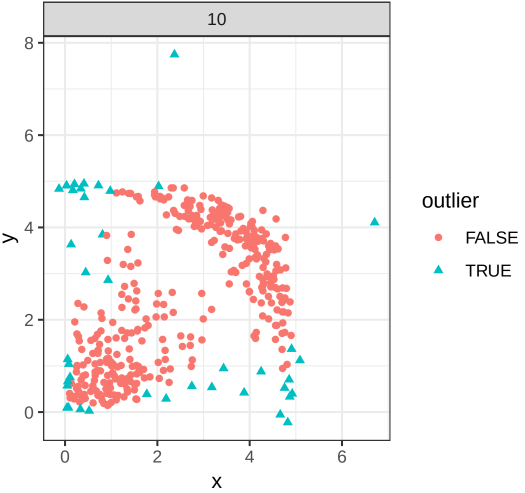
- A tree induces a partition of the space
- A tree is grown randomly by induction.
- Given a rectangle which contains more than 1 point, we split it in two by chosing one variable and one threshold randomly.
- Stop when points are alone in their rectangle.

Isolation forest

- A tree induces a partition of the space
- A tree is grown randomly by induction.
- Given a rectangle which contains more than 1 point, we split it in two by chosing one variable and one threshold randomly.
- Stop when points are alone in their rectangle.

A tree provides a notion of depth which can be used as a score to measure abnormality. An isolation forest consists of several such trees, s_n is the average depth across trees.

No parameter (number of trees in the forest)



One class SVM

Main idea, find a ball of minimal radius which encloses all the points :

$$\min_{r \in \mathbb{R}, c \in \mathbb{R}^p} \quad r^2$$
$$\text{s.t.} \quad \|x_i - c\|^2 \leq r^2, \; i = 1 \ldots, n.$$

One class SVM

Main idea, find a ball of minimal radius which encloses all the points :

$$\min_{r\in\mathbb{R},c\in\mathbb{R}^p} \quad r^2$$
$$\text{s.t.} \quad \|x_i - c\|^2 \leq r^2,\ i = 1\ldots,n.$$

Too restrictive, add slack, $\nu > 0$

$$\min_{r\in\mathbb{R},c\in\mathbb{R}^p} \quad r^2 + \frac{1}{n\nu}\sum_{i=1}^{n}\xi_i$$
$$\text{s.t.} \quad \|x_i - c\|^2 \leq r^2 + \xi_i,\ i = 1\ldots,n.$$

One class SVM

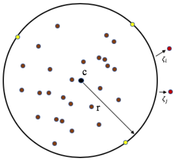
Main idea, find a ball of minimal radius which encloses all the points :

$$\min_{r\in\mathbb{R}, c\in\mathbb{R}^p} \quad r^2$$
$$\text{s.t.} \quad \|x_i - c\|^2 \leq r^2,\ i = 1\ldots, n.$$

Too restrictive, add slack, $\nu > 0$

$$\min_{r\in\mathbb{R}, c\in\mathbb{R}^p} \quad r^2 + \frac{1}{n\nu}\sum_{i=1}^{n}\xi_i$$
$$\text{s.t.} \quad \|x_i - c\|^2 \leq r^2 + \xi_i,\ i = 1\ldots, n.$$

s_n is roughly the distance to the center.



One class SVM

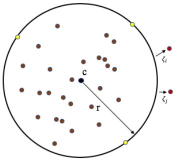
Main idea, find a ball of minimal radius which encloses all the points :

$$\min_{r\in\mathbb{R}, c\in\mathbb{R}^p} \quad r^2$$
$$\text{s.t.} \quad \|x_i - c\|^2 \leq r^2,\ i = 1\dots, n.$$

Too restrictive, add slack, $\nu > 0$

$$\min_{r\in\mathbb{R}, c\in\mathbb{R}^p} \quad r^2 + \frac{1}{n\nu}\sum_{i=1}^{n}\xi_i$$
$$\text{s.t.} \quad \|x_i - c\|^2 \leq r^2 + \xi_i,\ i = 1\dots, n.$$

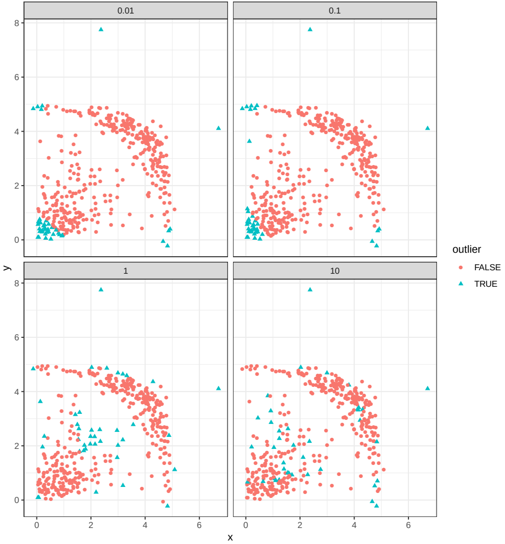
s_n is roughly the distance to the center.



Kernel trick : ϕ: $x \mapsto X \in \mathbb{R}^P$ sends x to a high (infinite) dimensional feature space.
Implicitely : $x_i \to \phi(x_i)$, $i = 1, \dots, n$.
Positive definite kernel (*ex :* Gaussian) implicitely encodes ϕ.

One class SVM

Gaussian kernel with varying bandwidth



Take away

Outliers correspond roughly to boundary of the cloud or low density regions.

Take away

Outliers correspond roughly to boundary of the cloud or low density regions.

- Many methods
- Some algorithms are very complex, some are very simple
- The output of some methods is random.
- Many parameters.

Take away

Outliers correspond roughly to boundary of the cloud or low density regions.

- Many methods
- Some algorithms are very complex, some are very simple
- The output of some methods is random.
- Many parameters.

Caveats :

- Many ways to choose (or not) the score threshold.
- Tune hyperparameters, similar as supervised learning.

Take away

Outliers correspond roughly to boundary of the cloud or low density regions.

- Many methods
- Some algorithms are very complex, some are very simple
- The output of some methods is random.
- Many parameters.

Caveats :

- Many ways to choose (or not) the score threshold.
- Tune hyperparameters, similar as supervised learning.

Exercise : For each method that we have seen describe

- How many parameters ?
- Is the computed score random ?
  - if you run the algorithm twice, do you get the same result ?
- Do anomaly correspond to large or small values of the score ?

Outline

1. What is an outlier ?
2. Score based detection
3. Univartiate methods
4. Multivariate approaches
5. Practical