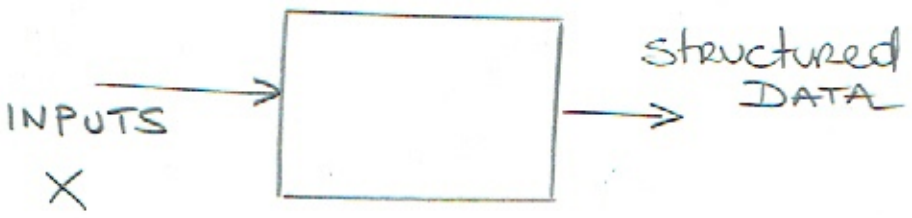
UNSUPERVISED LEARNING

- ANOMALY detection ①
- CLUSTERING



NO LABELS (OR JUST A few)
↳ SEMI-SUPERVISED learning.

ANOMALY Detection

Applications ↳ if IN REAL-TIME → QUICK AND DIRTY methods needed

* Credit card fraud → fraud
* Quality control → OK → NO fraud (- class)
  → NOT OK (- class)

- class >>> + class.

Put stress on a classifier!

↳ classifiers not ideal !

↳ ACCURACY not good KPI
⇒ USE ROC, AUC, F_1 ...

ANOMALY = OUTLIER

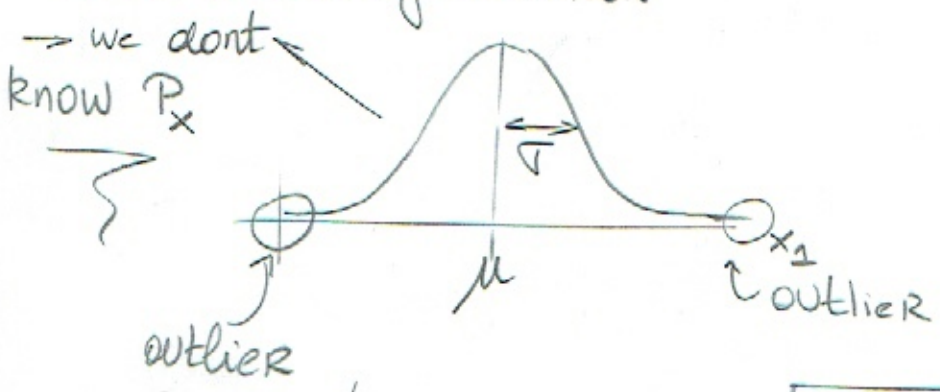
can change OVER time

↳ ANY deviation from NORMAL / EXPECTATION

POS NEG

ISSUE IN ANOMALY detection

→ we dont know P_x



① SIMPLE stat model

$P(x) < \varepsilon$ → ANOMALY (0.02)

Z-SCORE → $Z = \frac{x-\mu}{\sigma}$ → how many std. dev from mean is observation

(2) **Density based anomaly** → many variants (2)

↳ mass → # points / volume ↘ space

$D = \{x_1, x_2, \dots, x_d\}$

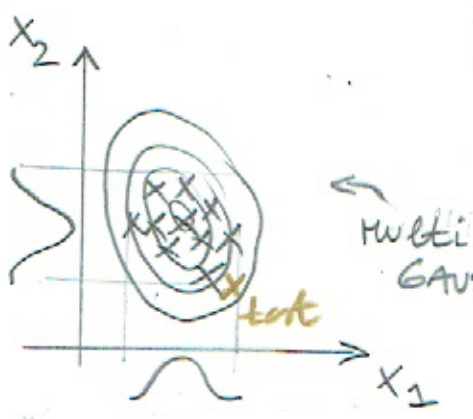
d features ↙ with unknown distribution

solution: try to estimate distribution through MLE parameters ↙

A) Assume Gaussian! $x_1^{(i)} \sim N(\mu_1, \sigma_1^2)$

$x_d^{(i)} \sim N(\mu_d, \sigma_d^2)$

↕ μ_i, σ_i unknown ↙ MLE



← multi-variate Gaussian

estimate μ_i, σ_i^2 so that the model best explains the observed data

B) Assume x_i are independent → $\prod_{j=1}^{d}$

(A) en (B) → $p(x)_{test} = \prod_{j=1}^{d} p(x_j; \mu_j, \sigma_j^2)$

So: 1) select features x_i that you think might be indicative of anomalous examples

2) fit parameters μ_i, σ_i^2

3) calculate $p(x)$ for new point x_{test}

if $P(x_{test}) < \varepsilon$ → anomaly ↓ 0.02

example of density-based ANOMALY Detection Syst

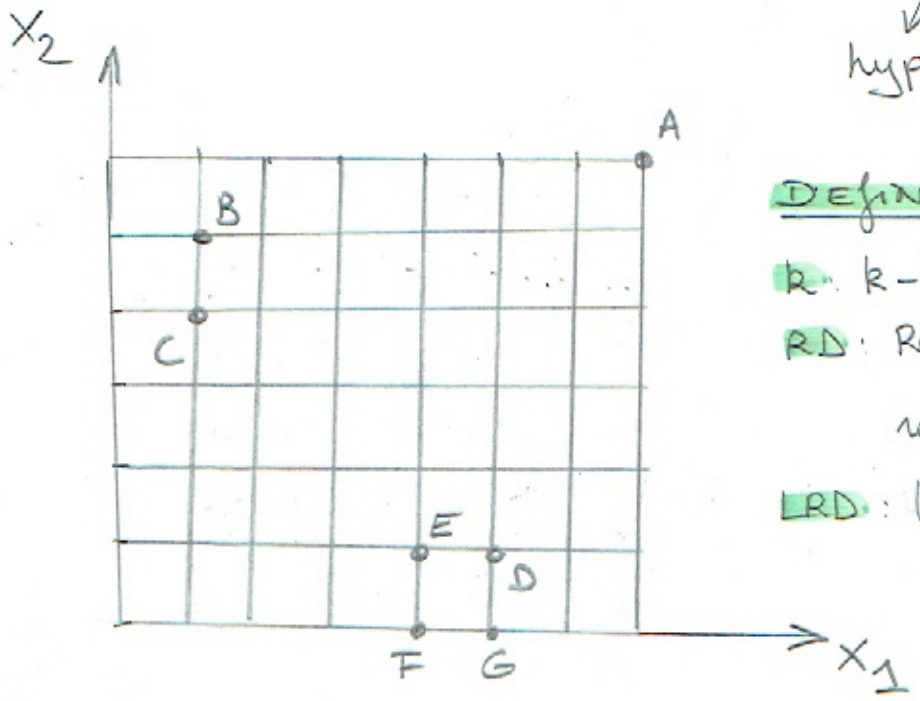
↳ LOCAL OUTLIER FACTOR : LOF (2000)

↳ compares local density of a point to the local density of its k neighbors

INTUITION : outliers are isolated

↳ low density AREA

hyper parameter !



DEFINITIONS

k : k-th neighbor of a point

RD : Reachability distance of a point A with respect to its k neighbors

LRD : local Reachability Density.

$$AV(RD_A) = \sum_k MAX(k\text{-}dist_R, dist(Ak))$$

↳ Average distance from pt A to its 3 neighbors !

↳ $LRD_A = [AV(RD_A)]^{-1}$ ← to go from distance to density !

k=3

$$RD_A = MAX(3rd\text{-}dist\ B, distance(AB)) + MAX(3rd\text{-}dist\ C, distance(AC)) + MAX(3rd\text{-}dist\ D, distance(AD))$$

5.65 (3rd-dist B) · 6.08 (distance AB) · 5 (3rd-dist C) · 6.32 (distance AC) · 1.4 (3rd-dist D) · 5.4 (distance AD)

↖ B, C, D don't see A as their kth-NN

$$\Rightarrow AV(RD_A) = \frac{1}{3}(6.08 + 6.32 + 5.4) = 5.9$$

→ $LRD_A = 0.168$

$RD_B = [\max(\text{3rd dist } A, \text{dist}(AB)) +$ (6.32, 6.08)
$\max(\text{3rd dist } C, \text{dist}(BC)) +$ (5, 1)
$\max(\text{3rd dist } D, \text{dist}(BD))]$ (1-4, 5.65)

$\rightarrow Av(RD_B) = \frac{1}{3}(6.32 + 5 + 5.65) = 5.65$

$LRD_B = 0.176$

$RD_C = [\max(\text{3rd dist } A, \text{dist}(AC)) +$ (6.32, 6.32)
$\max(\text{3rd dist } B, \text{dist}(BC)) +$ (5.65, 1)
$\max(\text{3rd dist } D, \text{dist}(CD))]$ (1.4, 5)

$\rightarrow Av(RD_C) = \frac{1}{3}(6.32 + 5.65 + 5)$

$= 5.65 \rightarrow LRD_C = 0.176$

$RD_D = [\max(3 \text{ dist } A, \text{dist}(AD)) +$ (6.32, 5.4)
$\max(3 \text{ dist } B, \text{dist}(BD)) +$ (5.65, 5.65)
$\max(3 \text{ dist } C, \text{dist}(CD))]$ (5, 5)

$\rightarrow AV(RD_D) = \frac{1}{3}(6.32 + 5.65 + 5)$

$= 5.65 \rightarrow LRD_D = 0.176$

$LOF_A = \frac{LRD_B + LRD_C + LRD_D}{3} \Big/ LRD_A = \frac{0.176}{0.168} = 1.05$ → outlier?

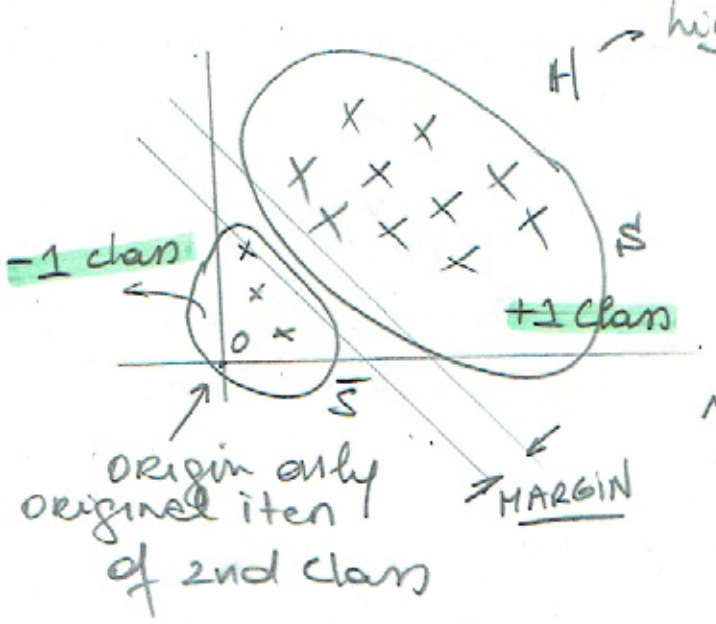
density A is smaller than its neighbors

(3) One Class SVM → UNSUPERVISED

Schölkopf 1999

$x_1, x_2, \dots, x_m$ → TRAINING examples belonging to 1 class

$\phi : X \to H$ VIA KERNEL



H → high DIM feature space

GOAL: Separate transformed pt from ORIGIN

choose { # features, KERNEL parameters → 'relaxation' parameters

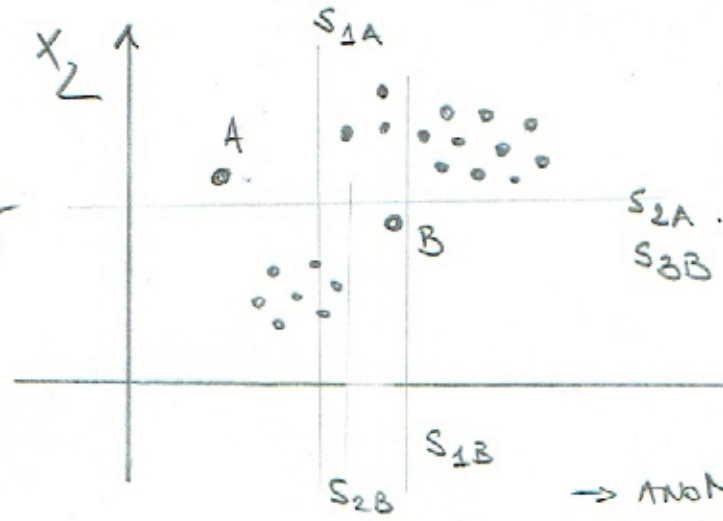
FIND subspace S of H such that

$\to P(x_{test} \notin S) < \nu$ ← PARAMETER replaces C → REST are outliers

$$f(x) = \begin{cases} +1 & \text{if } x \in S \\ -1 & \text{if } x \in \bar{S} \end{cases}$$ WITH $\bar{S}$ the complement of S

(4) Isolation Forest

FAST ACCURATE



A AND B ARE ANOMALOUS!

↓ takes 2 splits to isolate

takes 3 splits to isolate

→ ANOMALOUS pts have shorter paths to the tree

→ need fewer splits to be isolated

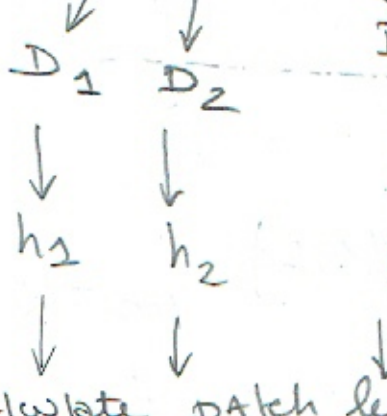
Renenber Random Forests

→ limit depth (6)
IN iForests

size D_i = size D

① split until you have 1 pt ~~per leaf~~

② split on $k < d$ features



← sampling with replacement

$$\bar{h} = \frac{1}{M} \sum_{i=1}^{M} h_i$$

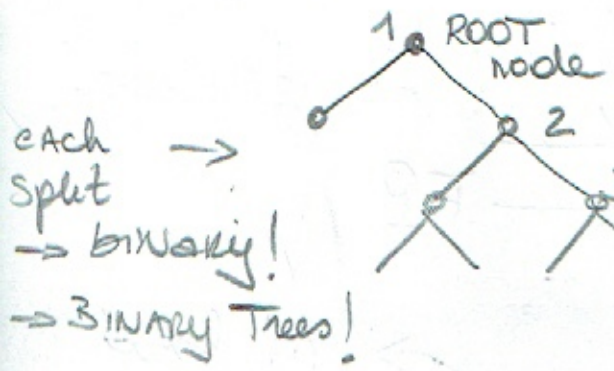
calculate path lenght for each point in each Tree

↳ # Nodes

→ average path length for each pt is a measure for its anomaly score (shorter path means higher anomaly score)

↳ 1 feature = 1 dimension

$X = \{x_1, \dots, x_M\}$ M data pts



each → Split → binary! → Binary Trees!

↳ n-1

INT Node → 2 children

ext Node → no kids

↳ fully grown trees have n external nodes.

↳ memory requirement ~ n

↳ limit depth of tree because anomalies are isolated much earlier compared to normal points

reduce → | SWAMPING → FALSE Positive
| MASKING → FALSE NEGative

iForests do not need to calculate distances
→ lowees comp costs
→ Avoid curse!

iForests work well with Large DATAsets!