Outlier Detection: Partitioning Methods

[Mining Massive Datasets]

Prof. [Carlos Castillo]

Topic 21

Universitat Pompeu Fabra Barcelona

Sources

Liu, F. T., Ting, K. M., & Zhou, Z. H. Isolation forest. ICDM 2008.

(1) Eryk Lewinson: Outlier detection with isolation forest (2018)

(2) Tobias Sterbak: Detecting network attacks with isolation forests (2018)

Partitioning-based method: isolation forest

Isolation forest method

- tree_build(X)
  - Pick a random dimension r of dataset X
  - Pick a random point p in $[\min_r(X), \max_r(X)]$
  - Divide the data into two pieces: $x_r < p$ and $x_r \geq p$
  - Recursively process each piece

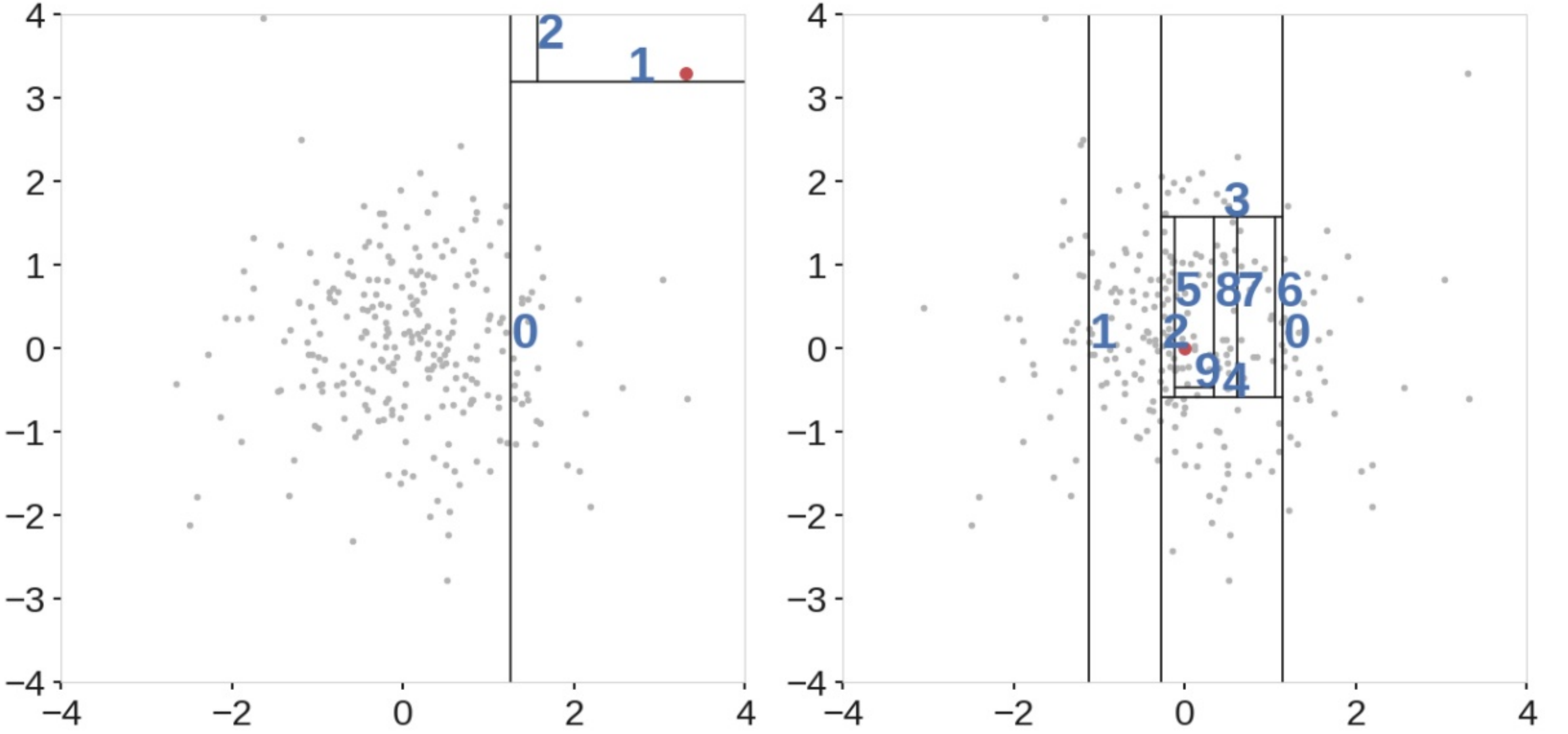
Stopping criteria for recursion

- Stop when a maximum depth has been reached

-or-

- Stop when each point is alone in one partition

Key: outliers lie at small depths



(a) Anomaly point (b) Nominal point

https://towardsdatascience.com/outlier-detection-with-extended-isolation-forest-1e248a3fe97b

Outlier score

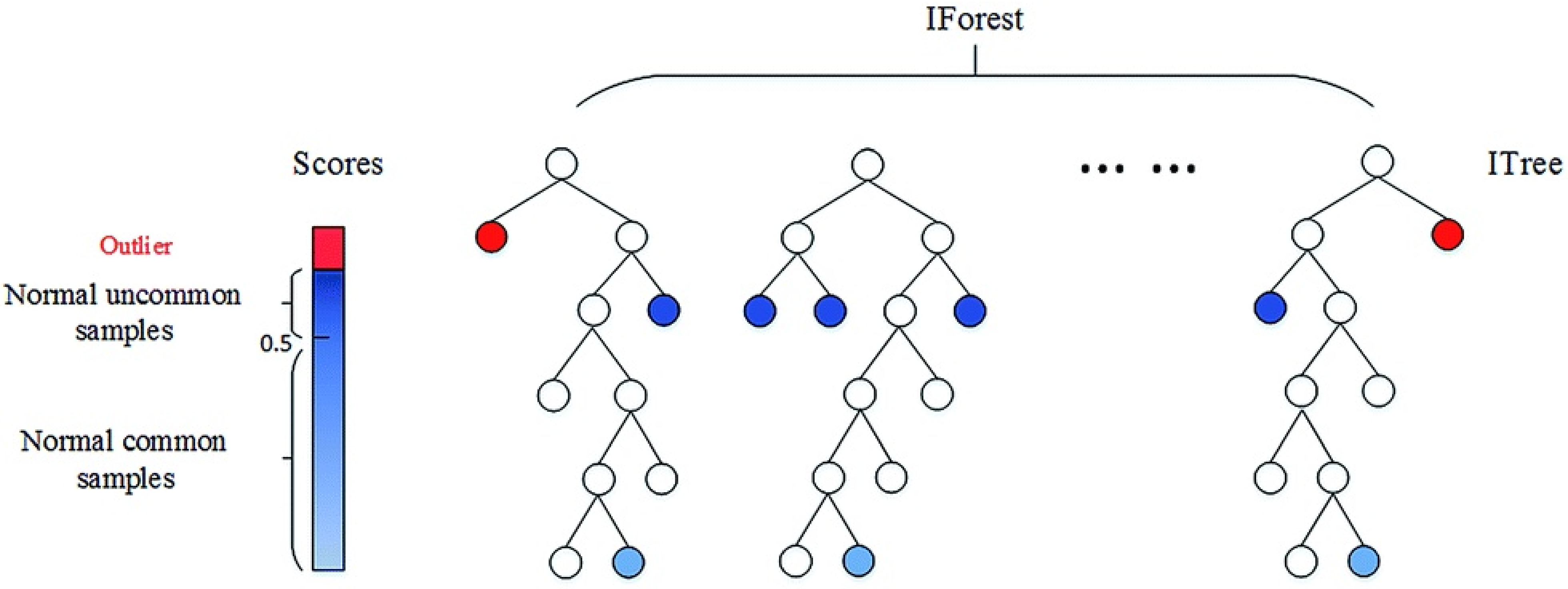
- Let c(n) be the average path length of an unsuccessful search in a binary tree of n items

$$c(n) = 2H(n-1) - (2(n-1)/n) \qquad H(n) = \sum_{k=1}^{n} \frac{1}{k}$$

- h(x) is the depth at which x is found in tree
- Score:

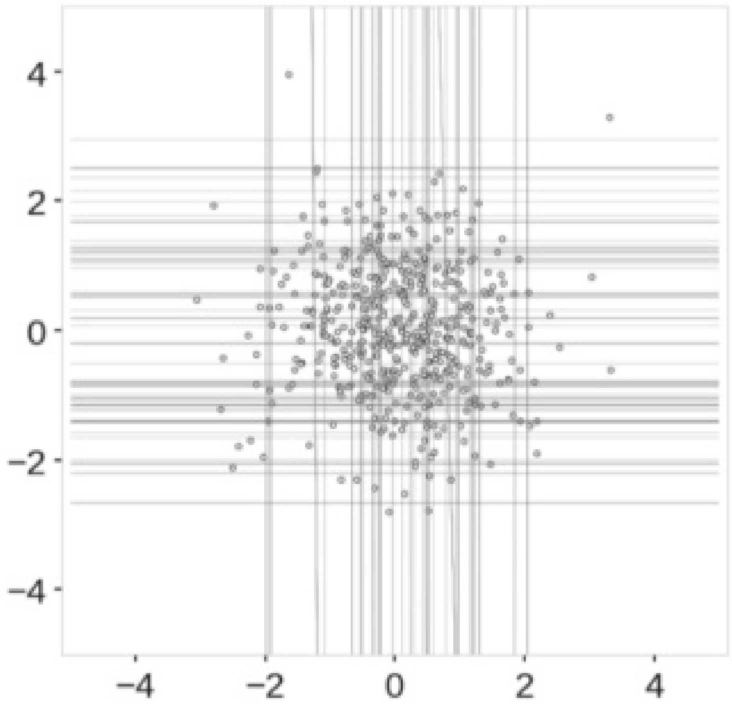
$$\text{outlier}(x, n) = 2^{-\frac{E(h(x))}{c(n)}}$$

Outlier scores in isolation forests
(each tree is built from a sub-sample of original data)

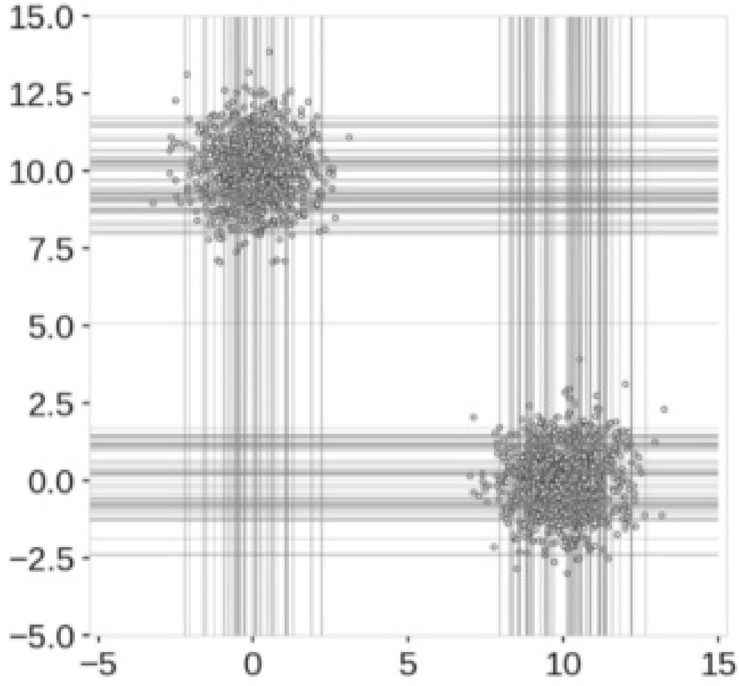


https://donghwa-kim.github.io/iforest.html

Example



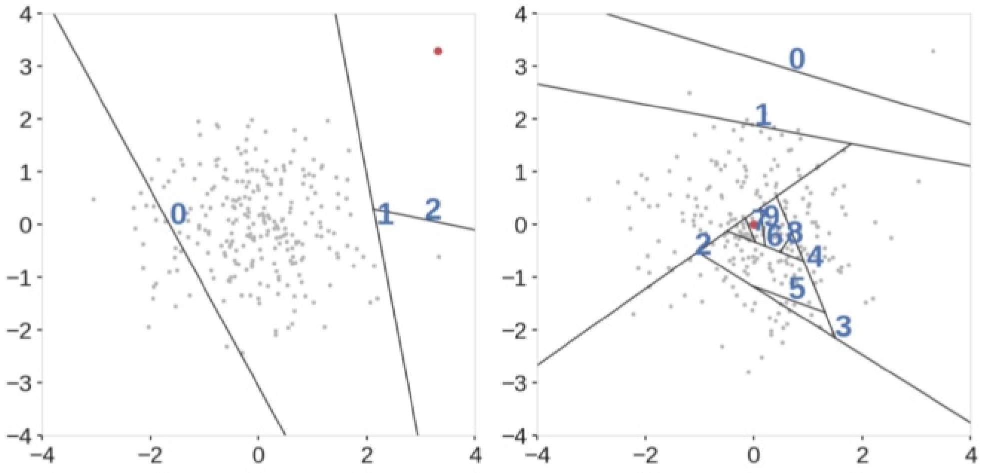
(a) Single blob



(b) Multiple Blobs

Extended Isolation Forest

- More freedom to partitioning by choosing a random slope and a random intercept



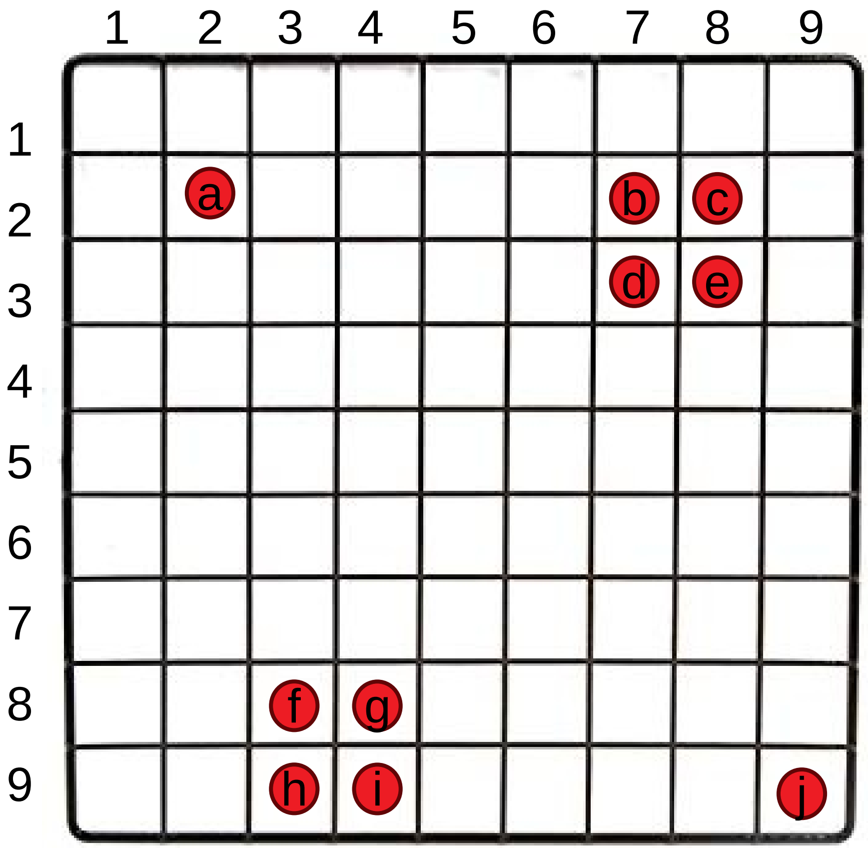
(a) Anomaly (b) Nominal

https://towardsdatascience.com/outlier-detection-with-extended-isolation-forest-1e248a3fe97b

Exercise

Answer in
Nearpod Draw-it

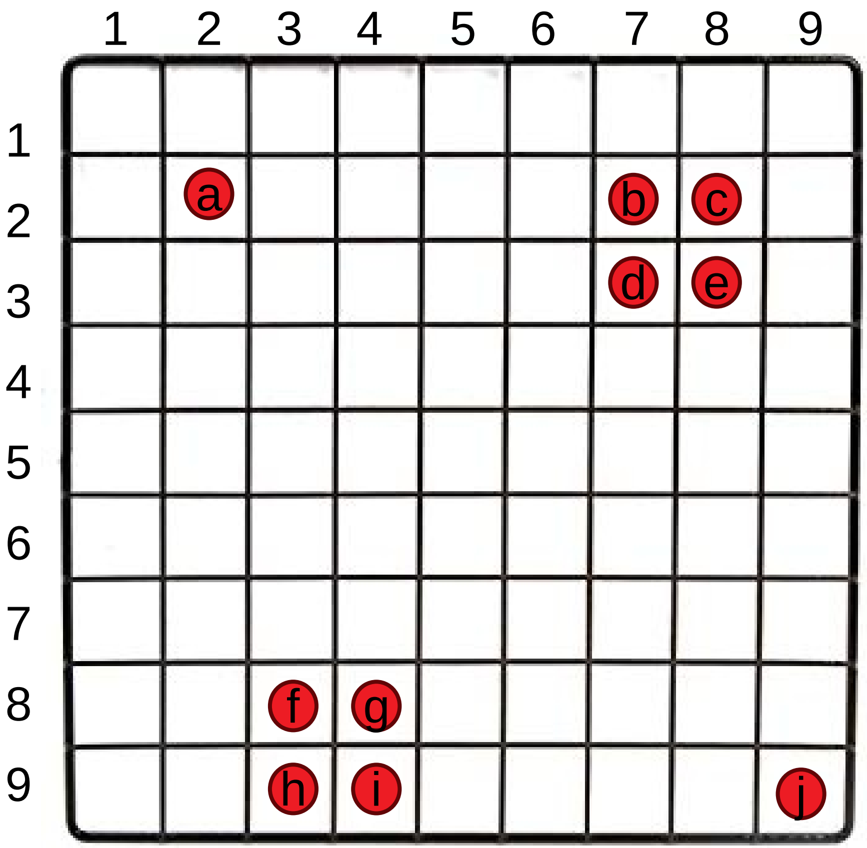
- Create one tree of the isolation forest by repeating 4 times:
  - Picking a sector containing >1 element
  - Picking a random dimension
  - Picking a random cut-off between min and max value along that dimension
- Draw the lines of your cuts
- Label each point with its depth h(x)



This is normally repeated several times, in the end:

$$\mathrm{outlier}(x,n) = 2^{-\frac{E(h(x))}{c(n)}}$$

In this case c(10) = 2xH(9) – (2x9/10) ≃ 3.857 ≃ **4**

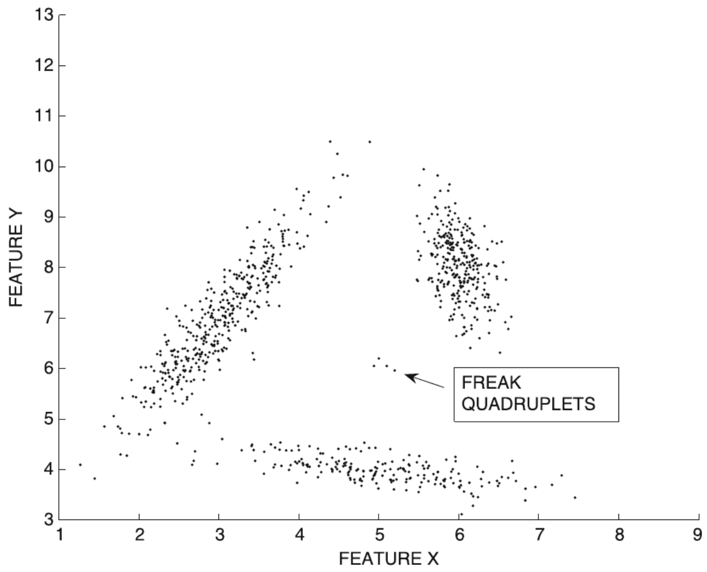

1 2 3 4 5 6 7 8 9
1 2 3 4 5 6 7 8 9
a
b
c
d
e
f
g
h
i
j

Additional materials

Distance-based methods

Instance-specific definition

- The distance-based outlier score of an object x is its distance to its kth nearest neighbor
- In this example of a small group of 4 outliers, we can set k > 3

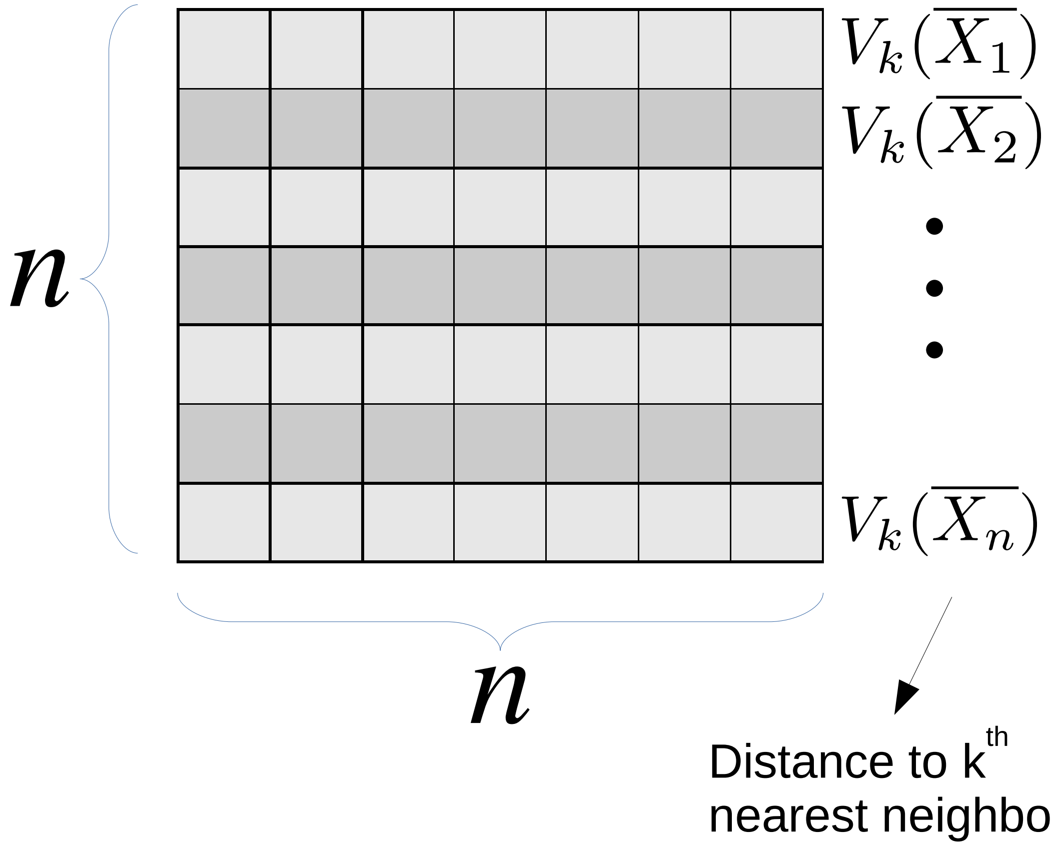


Problem: computational cost

- The distance-based outlier score of an object x is its distance to its k^{th} nearest neighbor
- In principle this requires $O(n^2)$ computations!
  - Index structure:
    useful only for cases of low data dimensionality
  - Pruning tricks:
    useful when only top-r outliers are needed

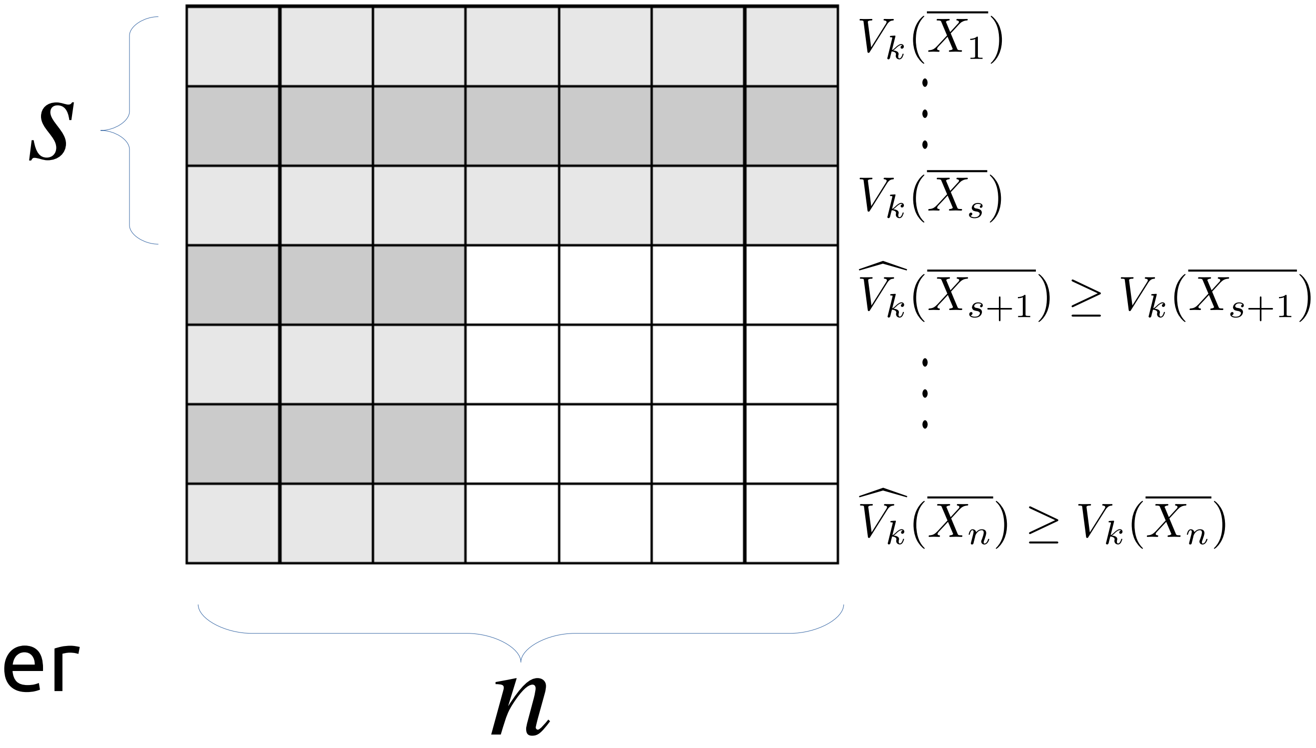
Problem: computational cost

- The distance-based outlier score of an item x is its distance to its kth nearest neighbor
- In principle this requires:
  - O(n^2) computations for evaluating the n x n distance matrix
  - O(n^2) computations for finding the r smallest values on each row



Pruning method: sampling

- Evaluate s x n distances
- For points *1…s* we are OK
- For points *(s+1)…n* we know only upper bounds

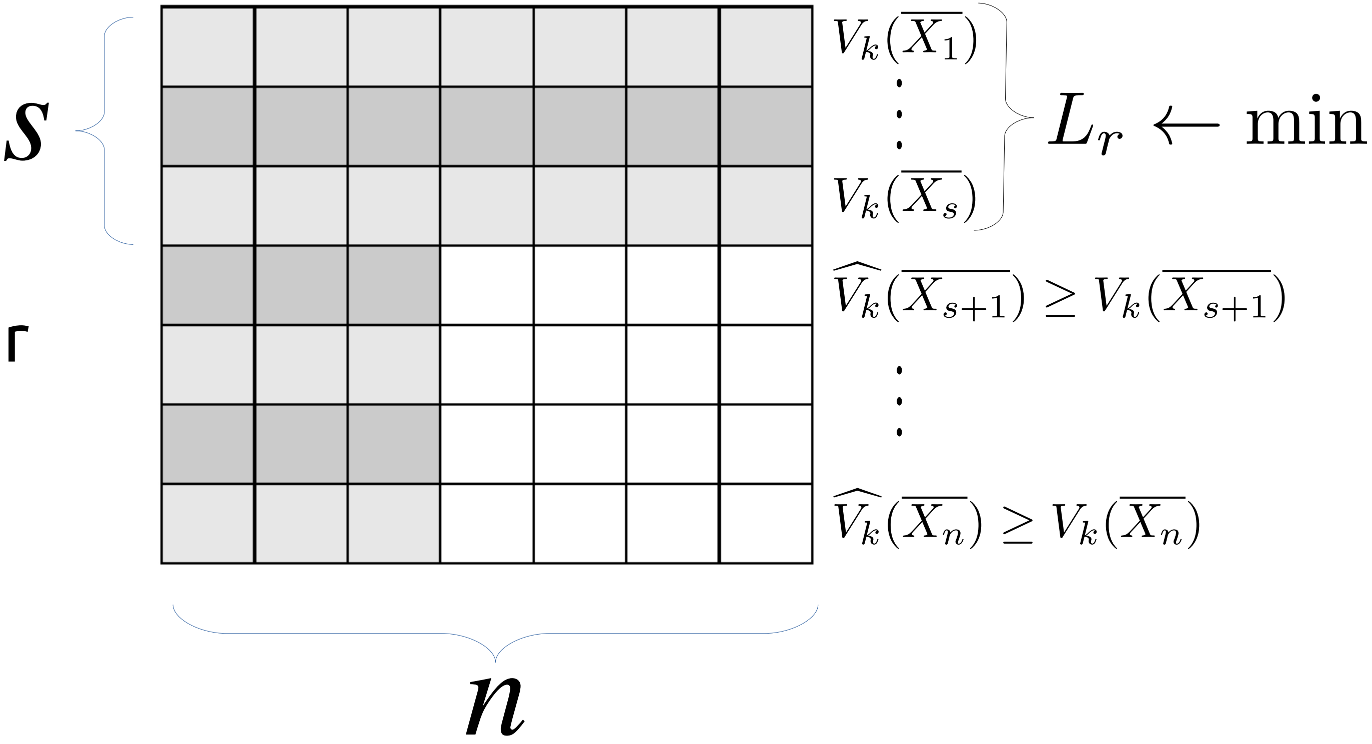


Pruning method: sampling (cont.)

From points $1 \ldots s$ we already know the r "winners"

($r \leq s$ nodes with the larger distance to their kth nearest neighbor)

Any point having $V_k < L_s$ cannot be among the top r outliers

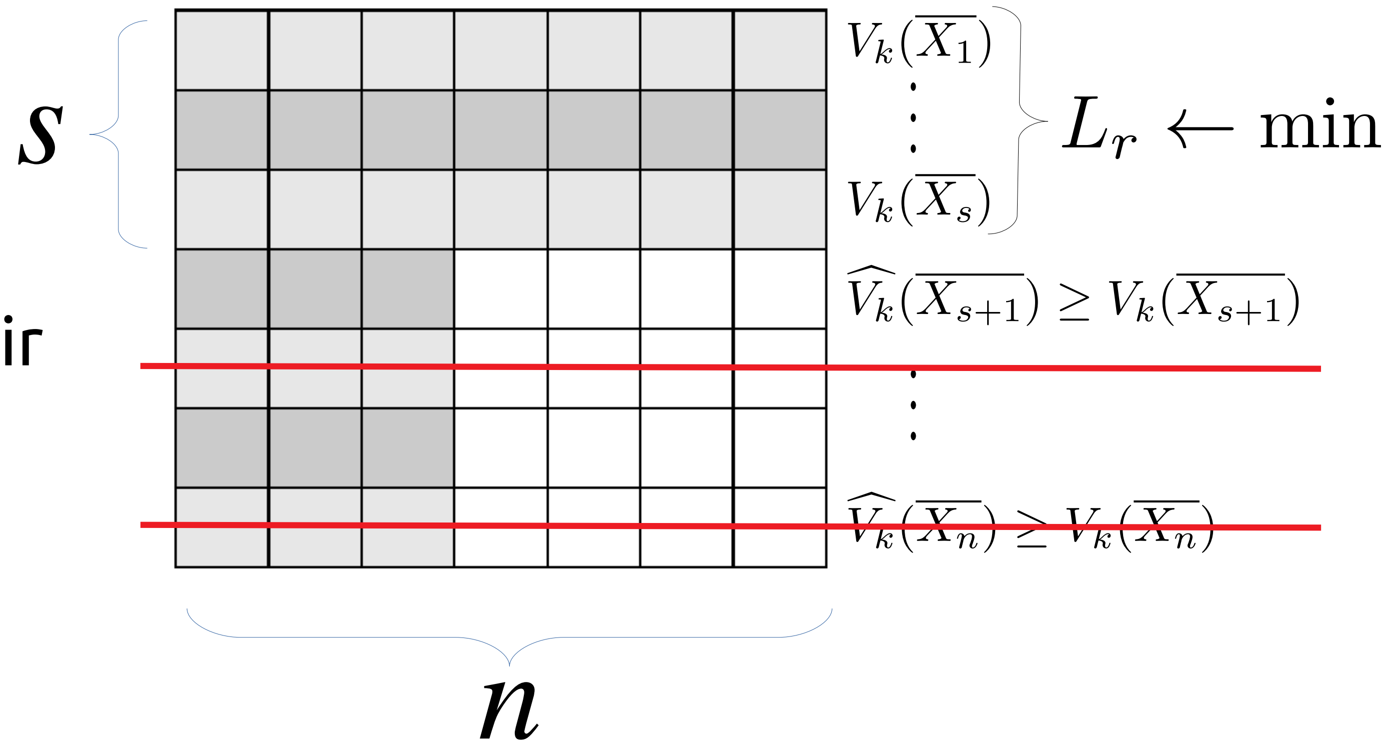


Pruning method: sampling (cont.)

From points *1…s* we already know the r “winners”

($r \leq s$ nodes with the larger distance to their kth nearest neighbor)

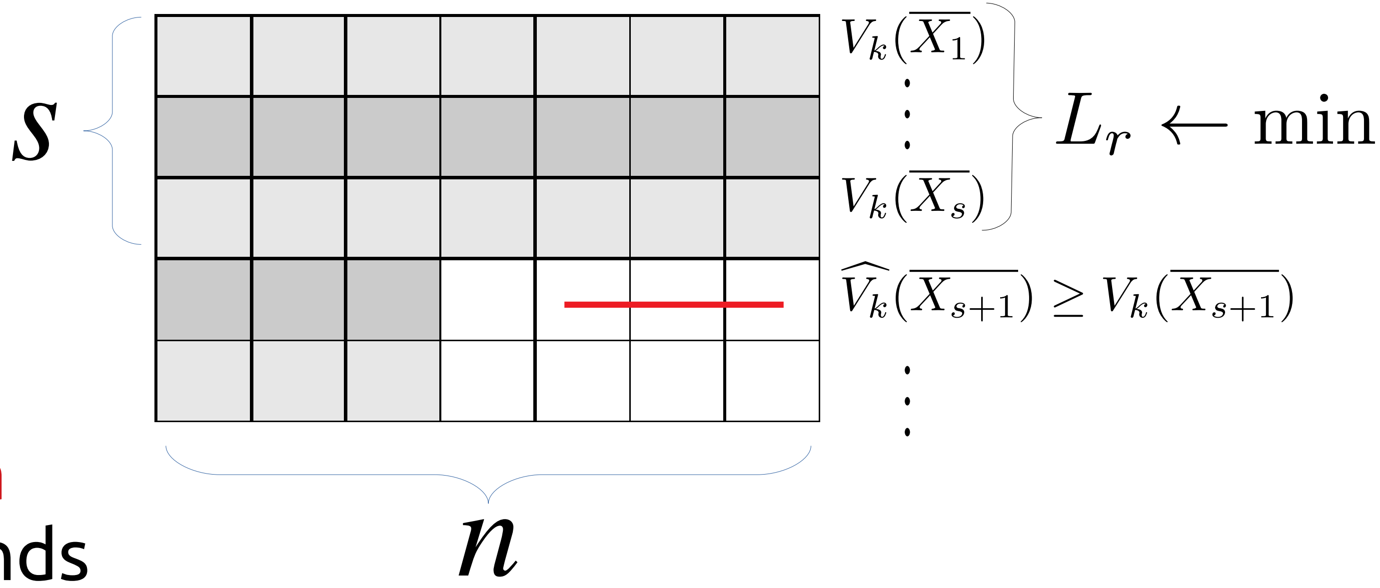
Any point having $V_k < L_s$ cannot be among the top r outliers



Pruning method: sampling (cont.)

Remove points having $\widehat{V_k} \leq L_r$

Update L_r keeping r largest values, and stop computing for a row if one already finds k nearest neighbors in that row that are all below distance L_r



Local outlier factor

Local Outlier Factor (LOF)

- Let $V_k(\overline{X})$ be the distance of $\overline{X}$ to its k-nearest neighbor
- Reachability distance

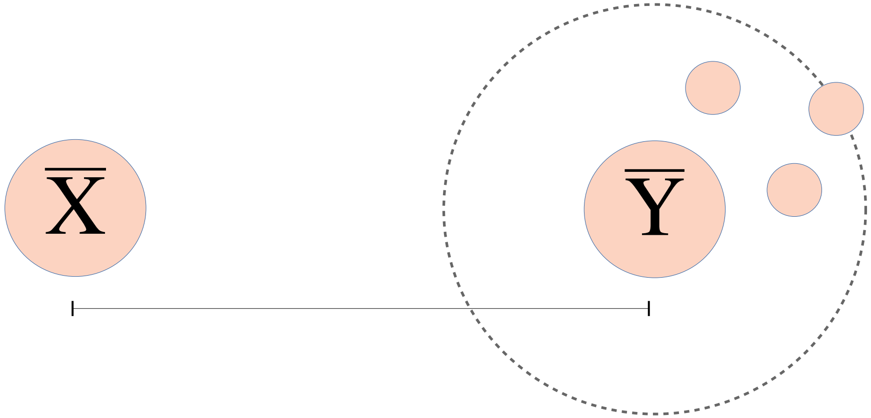
$$R_k(\overline{X}, \overline{Y}) = \max\{\mathrm{Dist}(\overline{X}, \overline{Y}), V_k(\overline{Y})\}$$

Local Outlier Factor (LOF) (cont.)

- $V_k(\overline{X})$: distance of $\overline{X}$ to its k-nearest neighbor
- Reachability distance

$$R_k(\overline{X}, \overline{Y}) = \max\{\text{Dist}(\overline{X}, \overline{Y}), V_k(\overline{Y})\}$$

- Not symmetric
- Equal to simple distance for long distances
- Smoothed by $V_k(\overline{X})$ for short distances



Local Outlier Factor (LOF) (cont.)

- Reachability distance

$$R_k(\overline{X}, \overline{Y}) = \max\{\mathrm{Dist}(\overline{X}, \overline{Y}), V_k(\overline{Y})\}$$

- Average reachability distance

$$AR_k(\overline{X}) = \underset{\overline{Y} \in L_k(\overline{X})}{E} \left[R_k(\overline{X}, \overline{Y})\right]$$

$L_k(\overline{X})$ is the set of points within distance $V_k(\overline{X})$ of $\overline{X}$ (might be more than k due to ties)

Local Outlier Factor (LOF) (cont.)

$$R_k(\overline{X}, \overline{Y}) = \max\{\text{Dist}(\overline{X}, \overline{Y}), V_k(\overline{Y})\}$$

$$AR_k(\overline{X}) = \underset{\overline{Y} \in L_k(\overline{X})}{E} \left[R_k(\overline{X}, \overline{Y})\right]$$

- Local outlier factor

$$\text{LOF}_k(\overline{X}) = \underset{\overline{Y} \in L_k(\overline{X})}{E} \frac{AR_k(\overline{X})}{AR_k(\overline{Y})}$$

Outlier score

$$\max_k \text{LOF}_k(\overline{X})$$

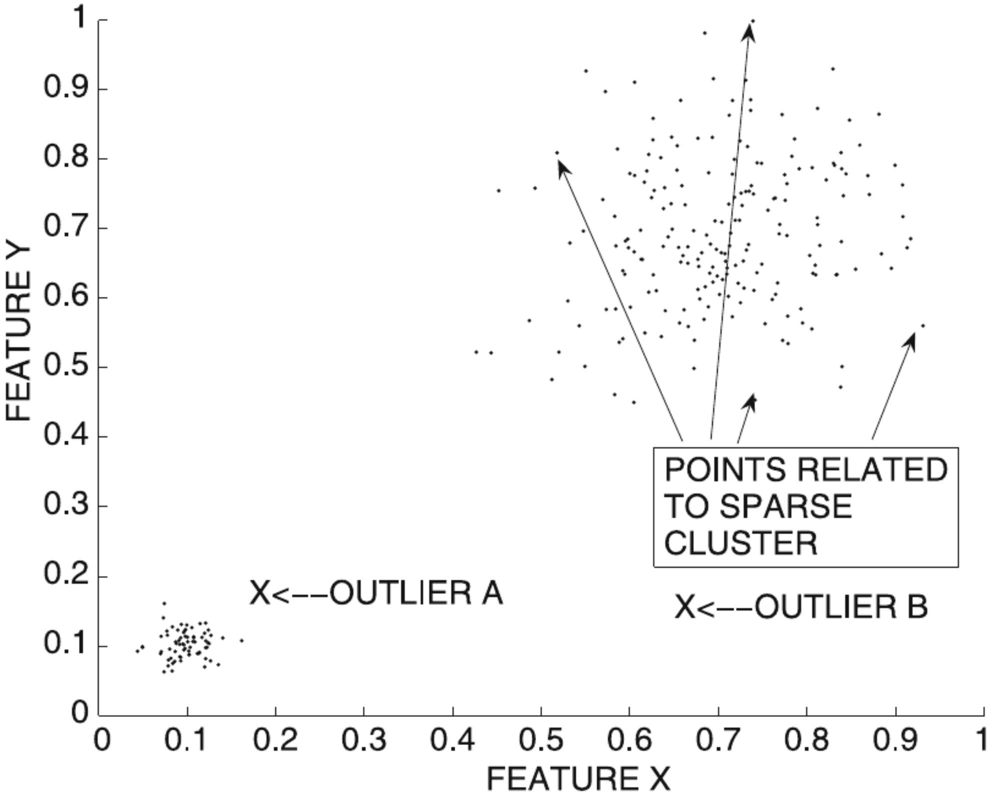
- Large for outliers, close to 1 for others

Local Outlier Factor (LOF) (cont.)

- Local outlier factor

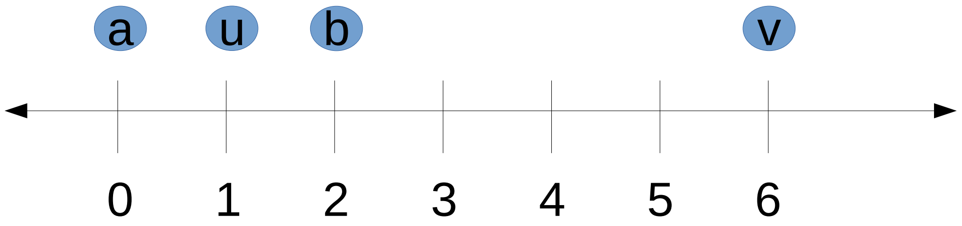
$$\mathrm{LOF}_k(\overline{X}) = \underset{\overline{Y} \in L_k(\overline{X})}{E} \frac{AR_k(\overline{X})}{AR_k(\overline{Y})}$$

- LOF values for points inside a cluster are close to one if cluster is homogeneous
- LOF values much higher for outliers: they are computed in terms of average distances of near-by clusters



Try it!

compare outlier score LOF(u), LOF(v)



- Let k=2
- $LOF_2(u)$ = E[{$AR_2(u)$ /$AR_2(a)$, $AR_2(u)$/$AR_2(b)$}] = ________
- $LOF_2(v)$ = E[{$AR_2(v)$ /$AR_2(b)$, $AR_2(v)$/$AR_2(u)$}] = ________
- $AR_2(u)$ = E[{$R_k(u,a)$, $R_k(u,b)$ }] = _______
- $AR_2(v)$ = E[{$R_k(v,b)$, $R_k(v,u)$ }] = _______
- $AR_2(a)$ = E[{$R_k(a,u)$, $R_k(a,b)$ }] = ______
- $AR_2(b)$ = E[{$R_k(b,u)$, $R_k(b,a)$ }] = ______
- $R_k(a,u)$ = _____; $R_k(a,b)$ = _____; $R_k(b,u)$ = ______; $R_k(b,a)$ = _______
- $R_k(u,a)$ = _____; $R_k(u,b)$ = ______; $R_k(v,b)$ = _______; $R_k(v,u)$ = ______
- V_2 = distance to 2nd nearest neighbor: $V_2(u)$ = _____ ; $V_2(v)$ = _____; $V_2(a)$ = _____ ; $V_2(b)$ = _____

$$\mathrm{LOF}_k(\overline{X}) = \underset{\overline{Y} \in L_k(\overline{X})}{E} \frac{AR_k(\overline{X})}{AR_k(\overline{Y})}$$

$$AR_k(\overline{X}) = \underset{\overline{Y} \in L_k(\overline{X})}{E} \left[R_k(\overline{X}, \overline{Y})\right]$$

$$R_k(\overline{X}, \overline{Y}) = \max\{\mathrm{Dist}(\overline{X}, \overline{Y}), V_k(\overline{Y})\}$$

Summary

Things to remember

- Isolation forest
- Clustering-based methods
- Distance-based methods

Exercises for TT19-TT21

- Data Mining, The Textbook (2015) by Charu Aggarwal
  - Exercises 8.11 → all except 10, 15, 16, 17