Outlier detection

Mining Massive Datasets

Carlos Castillo

Topic 09

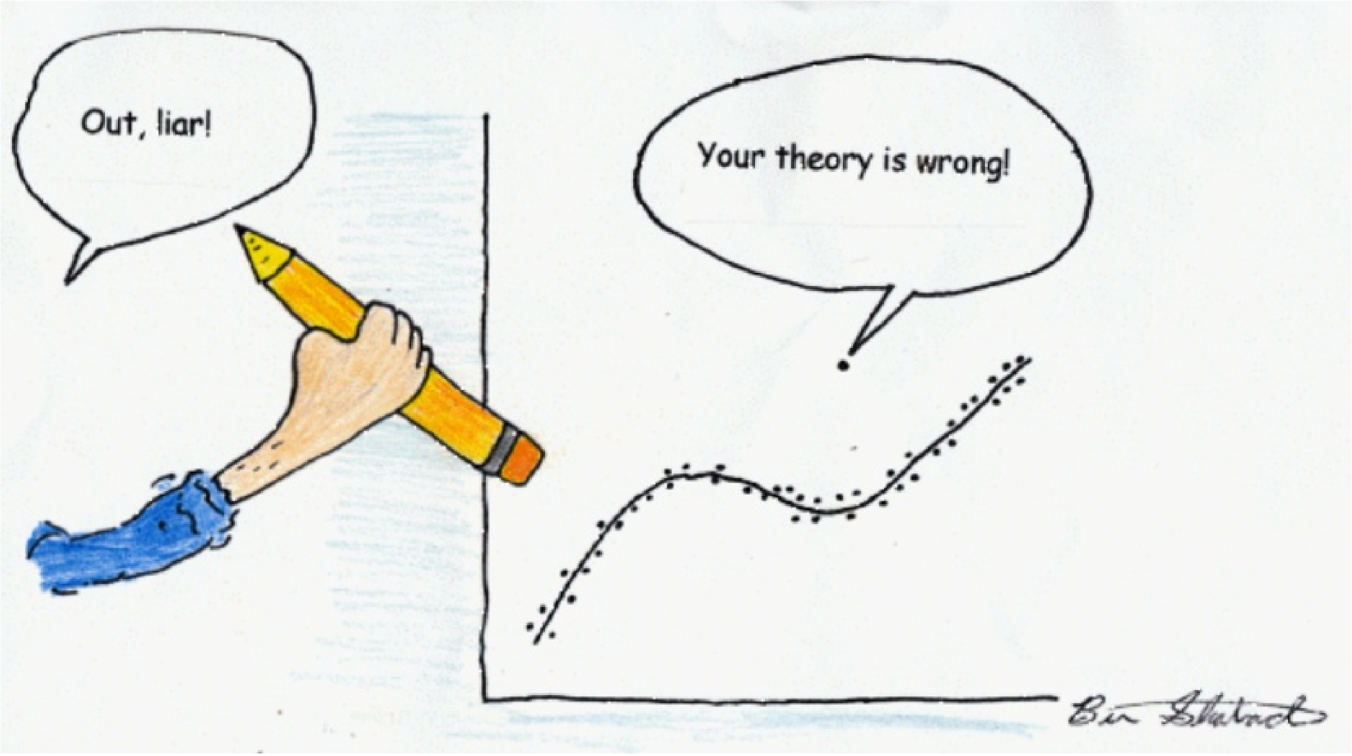
Universitat Pompeu Fabra Barcelona

Sources

- Data Mining, The Textbook (2015) by Charu Aggarwal (chapter 8) – slides by Lijun Zhang

What is an outlier?

- Informally, “An outlier is an observation which deviates so much from the other observations as to arouse suspicions that it was generated by a different mechanism.”
  - **Clustering** seeks to group points that are **similar**
  - **Outlier detection** seeks points that are **different** from the remaining data


Out, liar!
Your theory is wrong!

Some applications

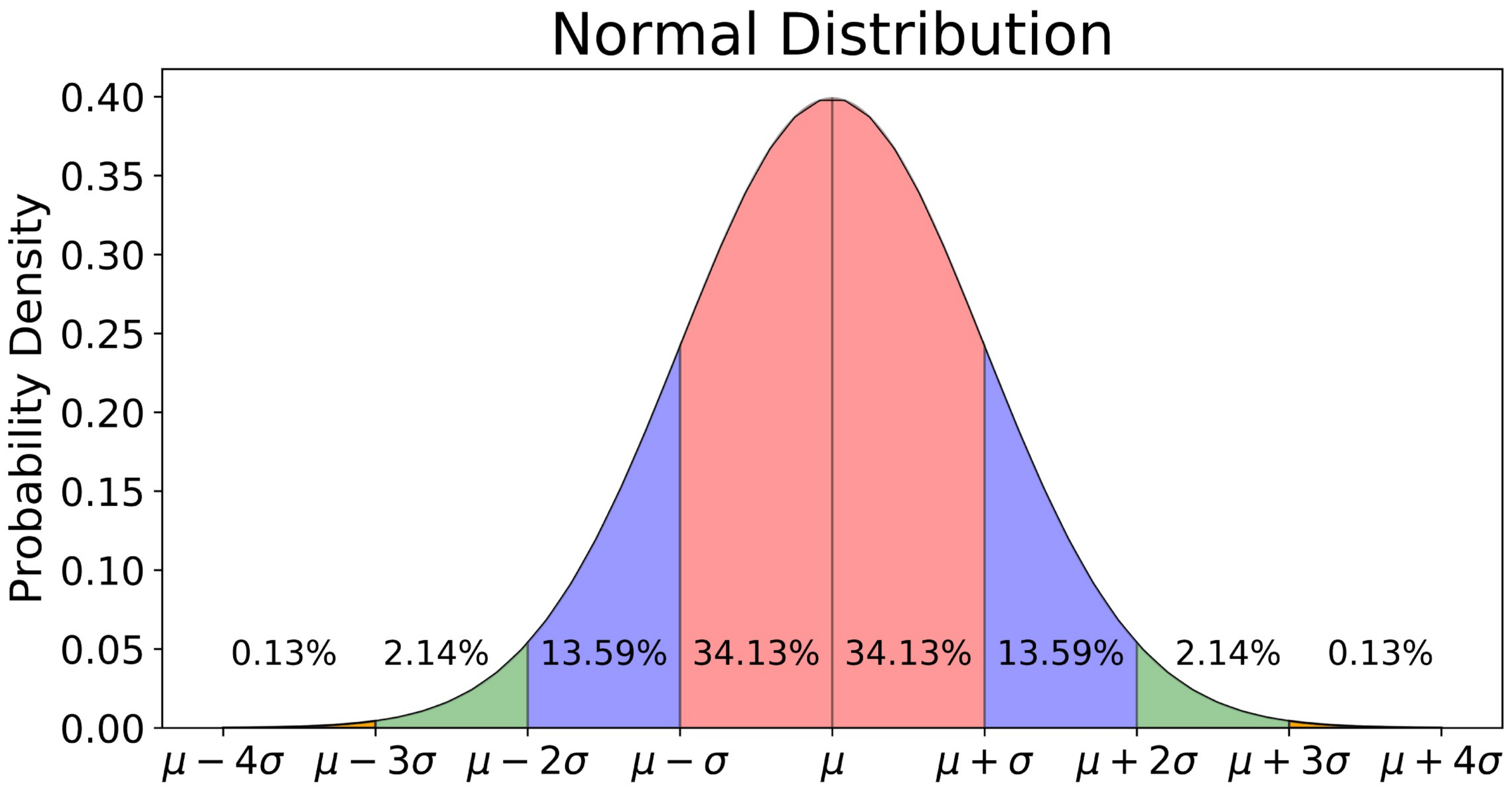
- Data cleaning
  - Remove noise in data
- Credit card fraud
  - Unusual patterns of credit card activity
- Network intrusion detection
  - Unusual records/changes in network traffic

Outlier detection methods

- Key idea
  - Create a model of normal patterns
  - Outliers are data points that do not naturally fit within this normal model
  - The "*outlierness*" of a data point is quantified by a outlier score
- Outputs of Outlier Detection Algorithms
  - Real-valued outlier score
  - Binary label (outlier / not outlier)

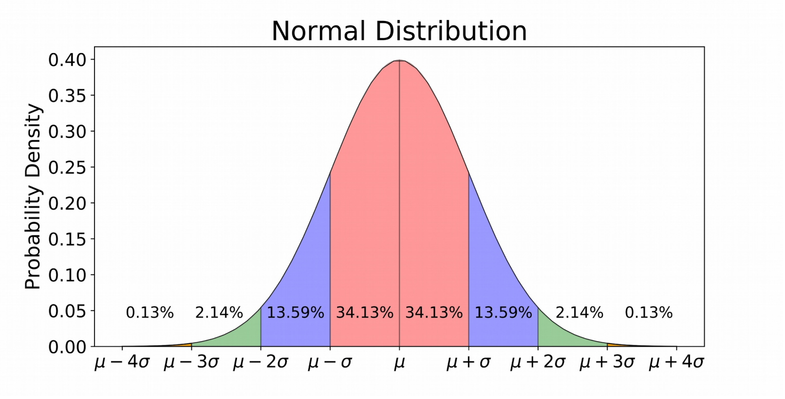
Extreme values analysis

Extreme value analysis: Statistical Tails



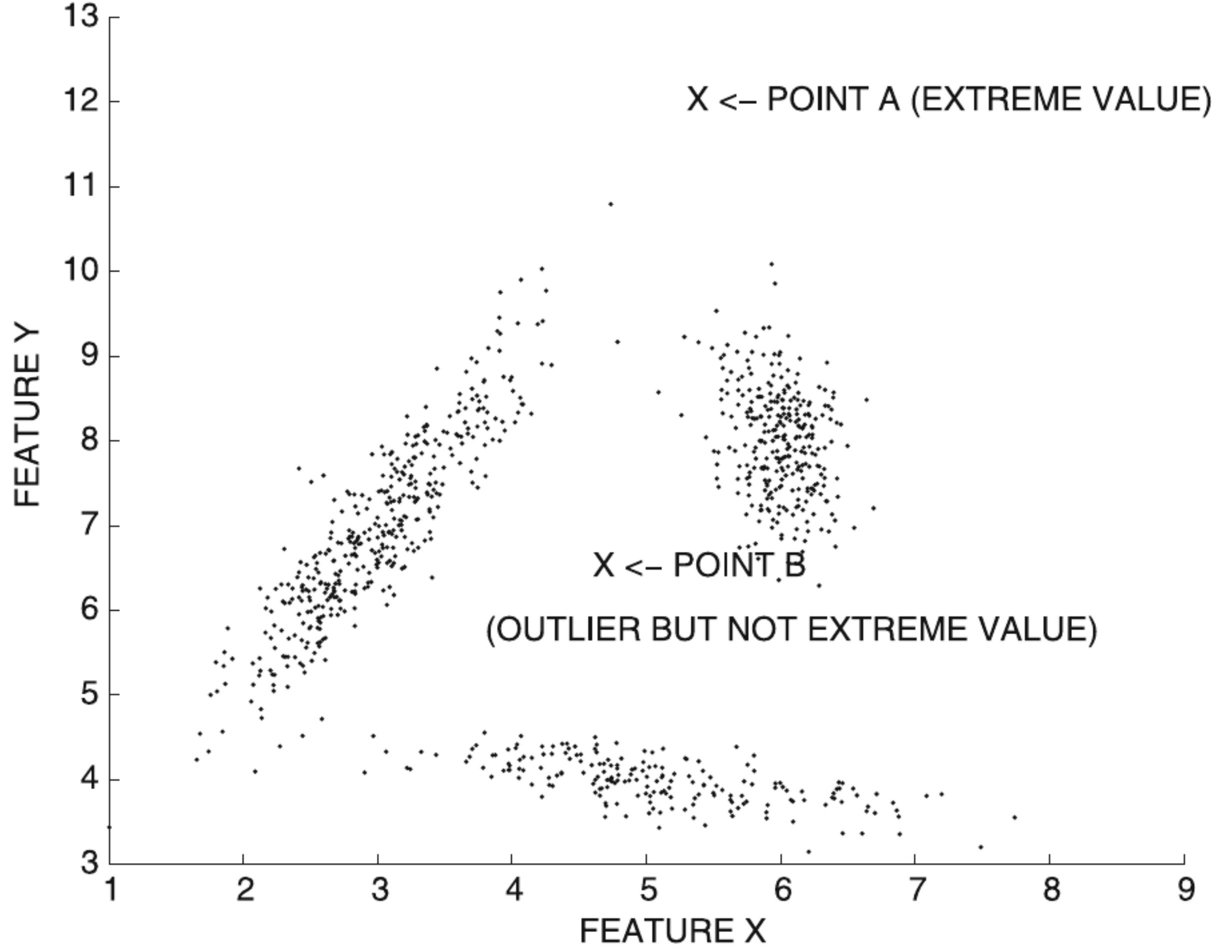
Extreme value analysis (cont.)

- Hypothesis: all extreme values are outliers
- However, outliers may not be extreme values:
  - {1,3,3,3,50,97,97,100}
  - 1 and 100 are extreme values
  - 50 is an outlier but no an extreme value



Extreme value analysis (cont.)

- Point A is an extreme value (hence, an outlier)
- Point B is an outlier but not an extreme value



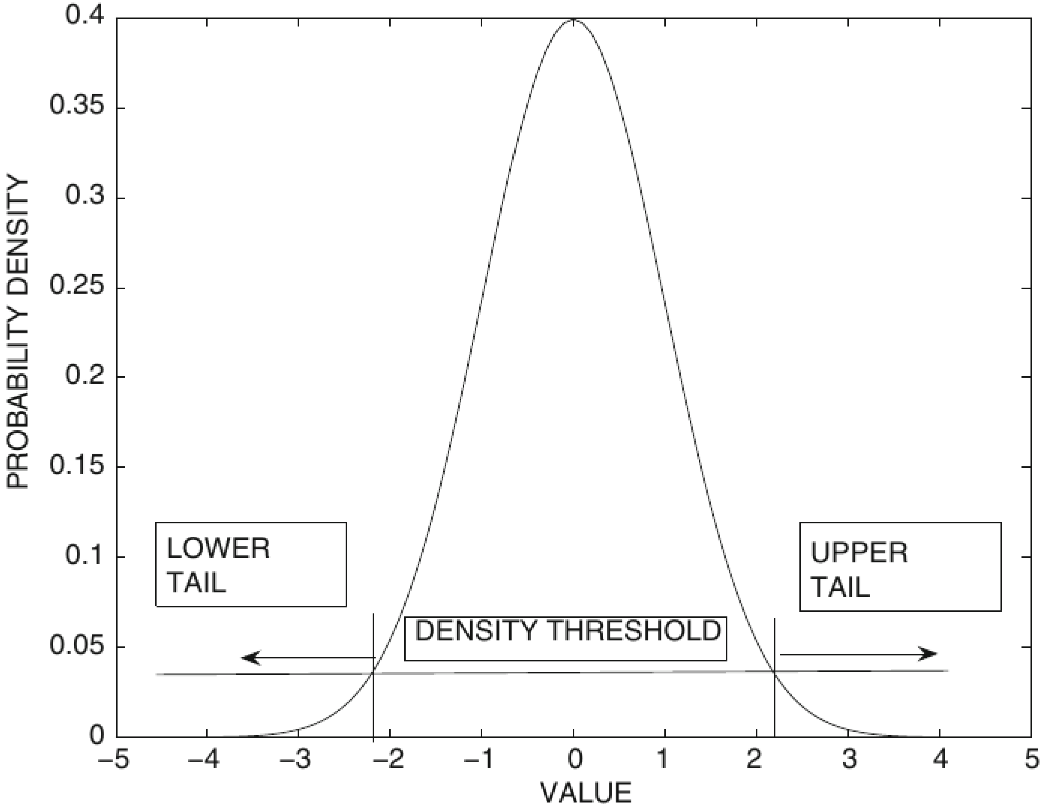
Univariate extreme value analysis

- Statistical tail confidence test
  - Let density be $f_X(x)$
  - Tails are **extreme** regions s.t. $f_X(x) \leq \theta$

Univariate extreme value analysis (cont.)

Let density be $f_X(x)$; tails are extreme regions s.t. $f_X(x) \leq \theta$

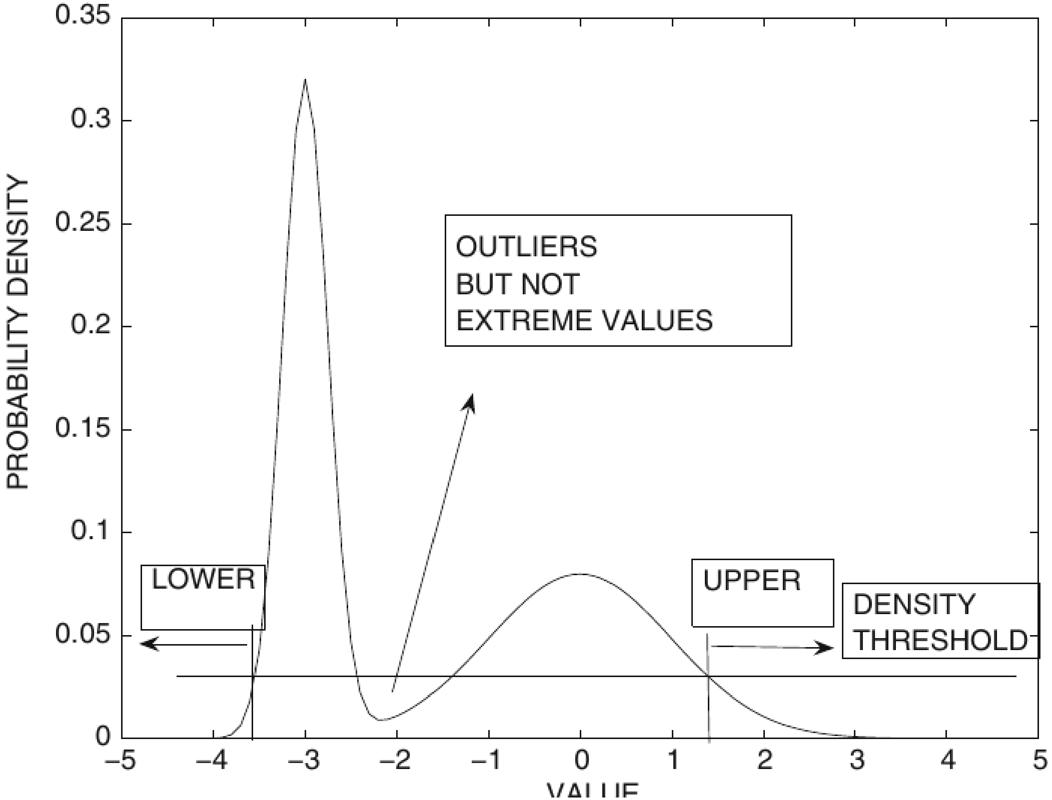
- Symmetric distribution
  - Two symmetric tails
  - Areas inside tails represent cumulative distribution



Univariate extreme value analysis (cont.)

Let density be $f_X(x)$; tails are extreme regions s.t. $f_X(x) \leq \theta$

- Asymmetric distribution
  - Areas in two tails are different
  - Regions in the interior are not tails



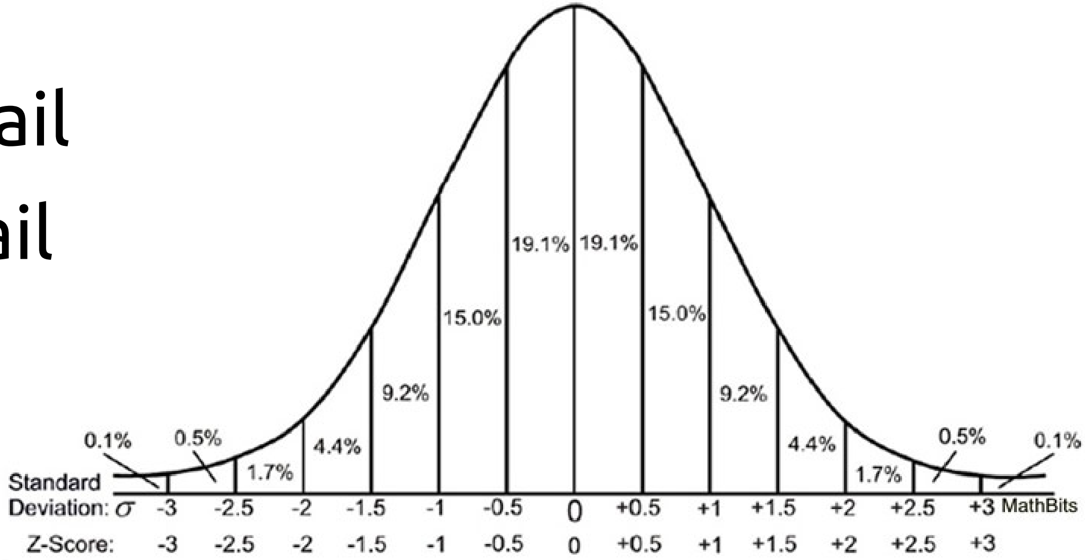
Standardization for univariate outlier analysis

- Normal distribution assumed $f_X(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$
- Parameters
  - From prior knowledge
  - Estimated from data

$$\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} x_i \qquad \hat{\sigma}^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \hat{\mu})^2$$

Standardization for univariate outlier analysis (cont.)

- z-value $z_i = \dfrac{x_i - \mu}{\sigma}$
- Follows normal distribution with mean 0 and standard deviation 1
  - Large z-value: upper tail
  - Small z-value: lower tail



Multivariate extreme values

- Probability density function of a Multivariate Gaussian distribution in d dimensions

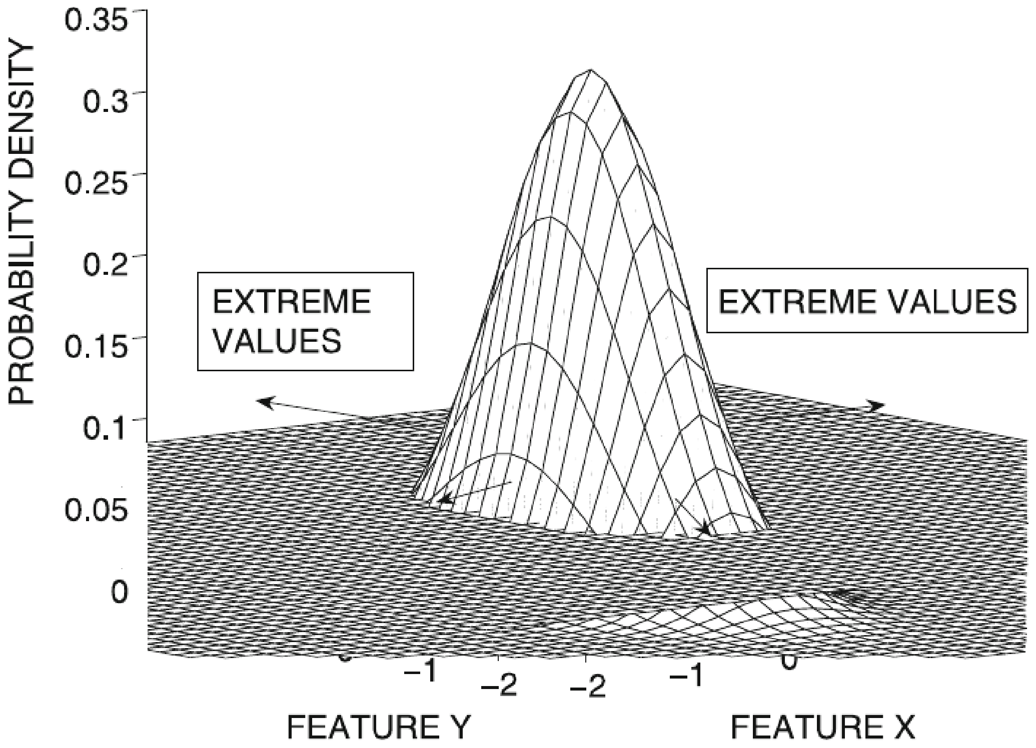
$$f(\overline{X}) = \frac{1}{\sqrt{|\Sigma|} \cdot (2 \cdot \pi)^{(d/2)}} \cdot e^{-\frac{1}{2} \cdot (\overline{X} - \overline{\mu}) \Sigma^{-1} (\overline{X} - \overline{\mu})^T}$$

$$= \frac{1}{\sqrt{|\Sigma|} \cdot (2 \cdot \pi)^{(d/2)}} \cdot e^{-\frac{1}{2} \cdot Maha(\overline{X}, \overline{\mu}, \Sigma)^2}$$

- $Maha(\overline{X}, \overline{\mu}, \Sigma)$ is the Mahalanobis distance
- $|\Sigma|$ *i*s the determinant of the covariances matrix

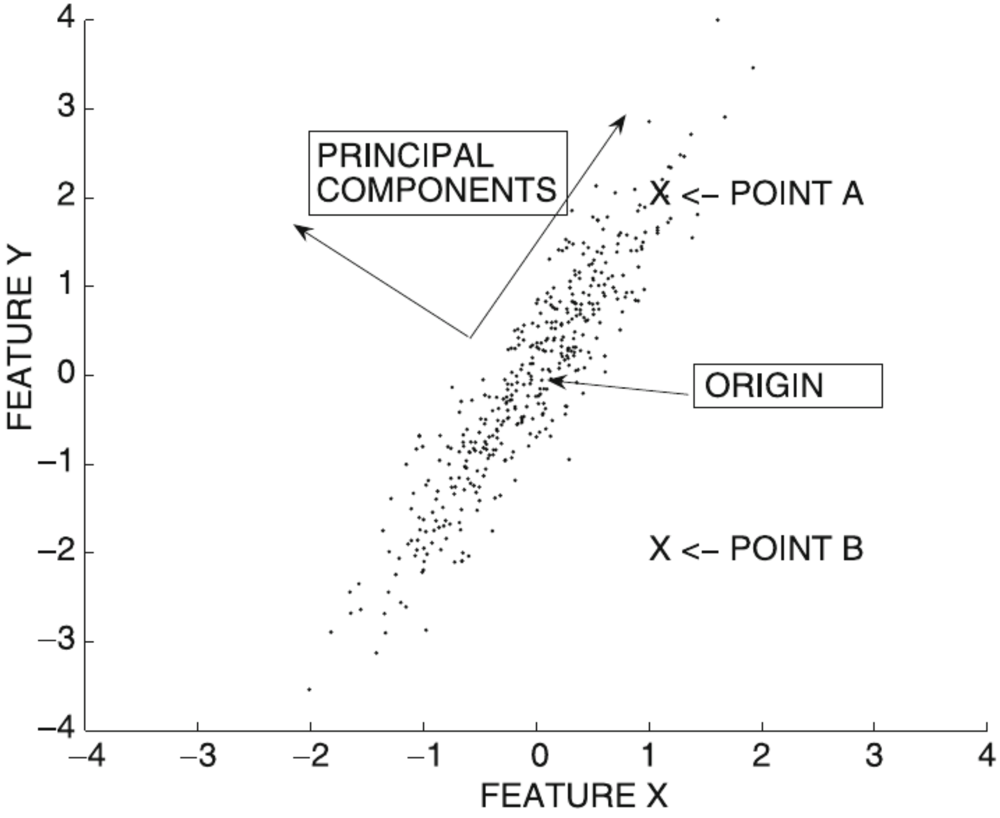
Multivariate extreme values (cont.)

- Extreme value score of $\overline{X}$
  - $Maha(\overline{X}, \overline{\mu}, \Sigma)$
  - Mahalanobis distance to the mean of the data
  - Larger values imply more extreme behavior



Multivariate extreme values (cont.)

- Extreme value score of $\overline{X}$
  - $Maha(\overline{X}, \overline{\mu}, \Sigma)$
  - Mahalanobis distance to the mean of the data
  - Larger values imply more extreme behavior
  - The Mahalanobis distance is the Euclidean distance in a transformed (axes-rotated) data set after dividing each of the transformed coordinate values by the standard deviation along its direction

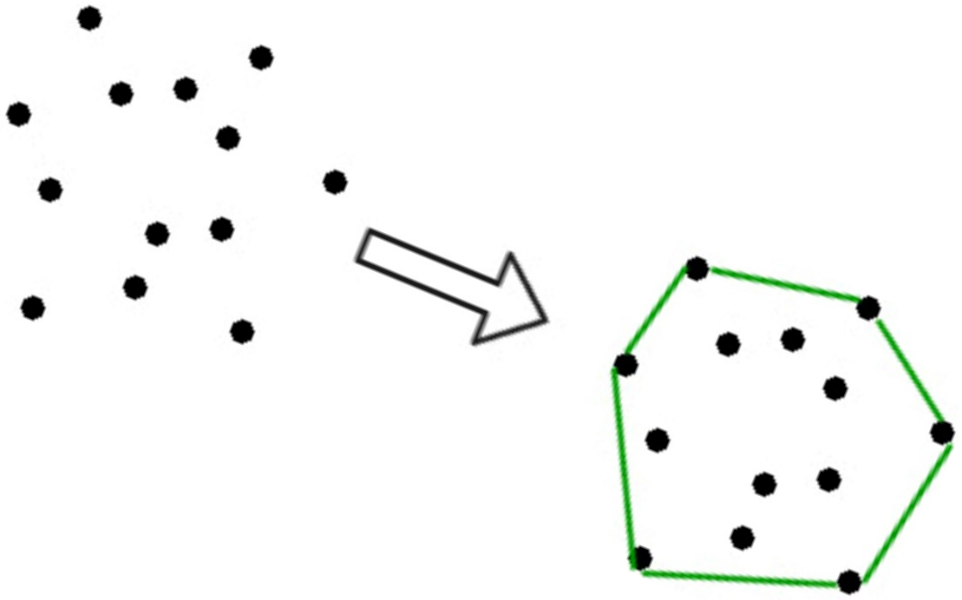


Depth-based methods

Key concept: convex hull

The convex hull of a set C is the set of all convex combinations of points in C

$$\begin{aligned} \text{conv } C = \{\theta_1 x_i + \cdots + \theta_k x_k | \\ x_i \in C, \\ \theta_i \geq 0, \\ \theta_1 + \cdots + \theta_k = 1\} \end{aligned}$$

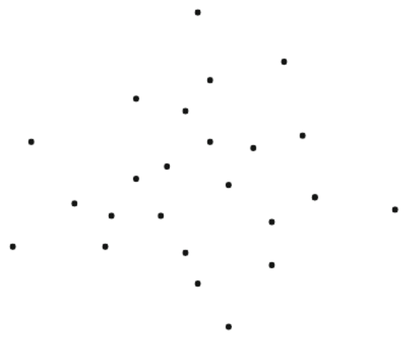


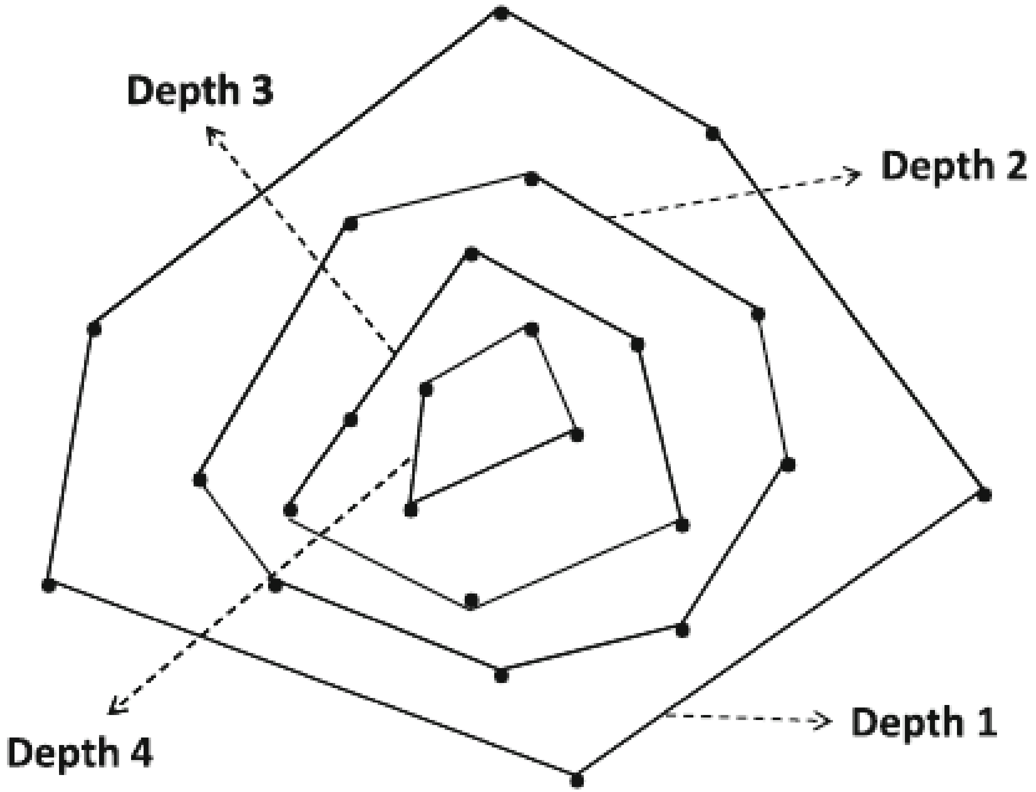
Algorithm

Algorithm *FindDepthOutliers*(Data Set: $\mathcal{D}$, Score Threshold: r)
begin
  $k = 1$;
  repeat
    Find set S of corners of convex hull of $\mathcal{D}$;
    Assign depth k to points in S;
    $\mathcal{D} = \mathcal{D} - S$;
    $k = k + 1$;
  until(D is empty);
  Report points with depth at most r as outliers;
end

Explanation: peeling layers

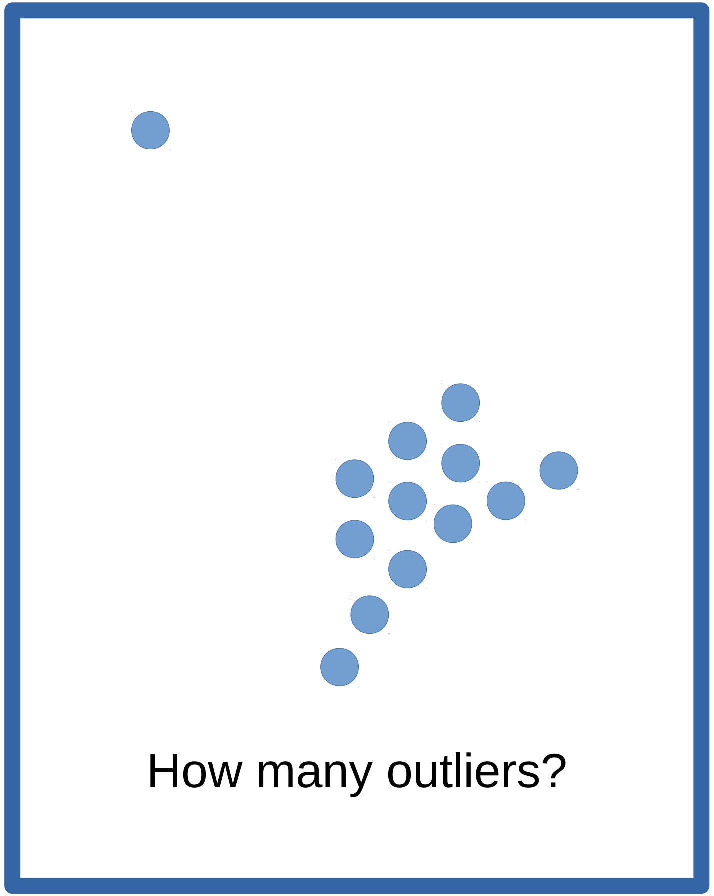
Algorithm *FindDepthOutliers*(Data Set: $\mathcal{D}$, Score Threshold: r)
begin
  $k = 1$;
  repeat
    Find set S of corners of convex hull of $\mathcal{D}$;
    Assign depth k to points in S;
    $\mathcal{D} = \mathcal{D} - S$;
    $k = k + 1$;
  until(D is empty);
  Report points with depth at most r as outliers;
end





Limitations of this method

- No normalization
- Computational complexity increases significantly with dimensionality
- Many data points are indistinguishable



Probabilistic methods

Related to probabilistic model-based clustering

- Assume data is generated from a mixture-based generative model
- Learn the parameters of the model from data
  - EM algorithm
- Evaluate the probability of each data point being generated by the model
  - Points with low values are outliers

Mixture-based generative model

- Data is generated by a mixture of k distributions with probability distributions $G_1, \ldots, G_k$
- Each point $\overline{X}$ is generated as follows:
  1) Select a mixture component with probability α_i
     - Suppose it's component r
  2) Sample a data point from distribution G_r

Learning parameters from data

- Probability of generating a point

$$f^{\text{point}}\left(\overline{X_j}|\mathcal{M}\right) = \sum_{i=1}^{k} P\left(\mathcal{G}_i, \overline{X_j}\right)$$

$$= \sum_{i=1}^{k} P(\mathcal{G}_i)P(\overline{X_j}|\mathcal{G}_i)$$

$$= \sum_{i=1}^{k} \alpha_i f^i(\overline{X_j})$$

Learning parameters from data

- Probability of generating a point

$$\mathrm{f}^{\text{point}}\left(\overline{X_j}|\mathcal{M}\right) = \textstyle\sum_{i=1}^{k} \alpha_i f^i(\overline{X_j})$$

- Probability of generating a dataset

$$f^{\text{data}}(\mathcal{D}|\mathcal{M}) = \prod_{j=1}^{n} f^{\text{point}}(\overline{X_j}|\mathcal{M})$$

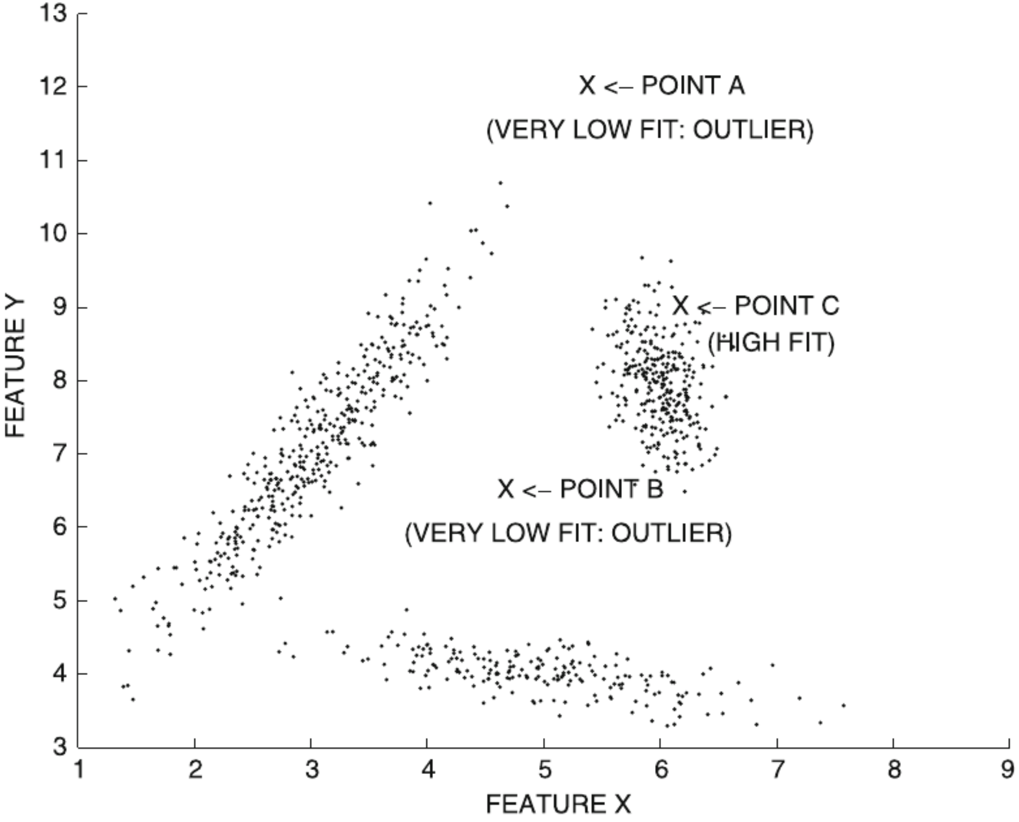
- Learning: min log loss

$$\min \mathcal{L}\left(\mathcal{D}|\mathcal{M}\right) = \log\left(\prod_{j=1}^{n} f^{\text{point}}\left(\overline{X_j}|\mathcal{M}\right)\right) = \sum_{j=1}^{n} \log\left(\sum_{i=1}^{k} \alpha_i f^i\left(\overline{X_j}\right)\right)$$

Identifying an outlier

Outlier score:

$$f^{\text{point}}\left(\overline{X_j}|\mathcal{M}\right) = \sum_{i=1}^{k} \alpha_i f^i(\overline{X_j})$$



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

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)

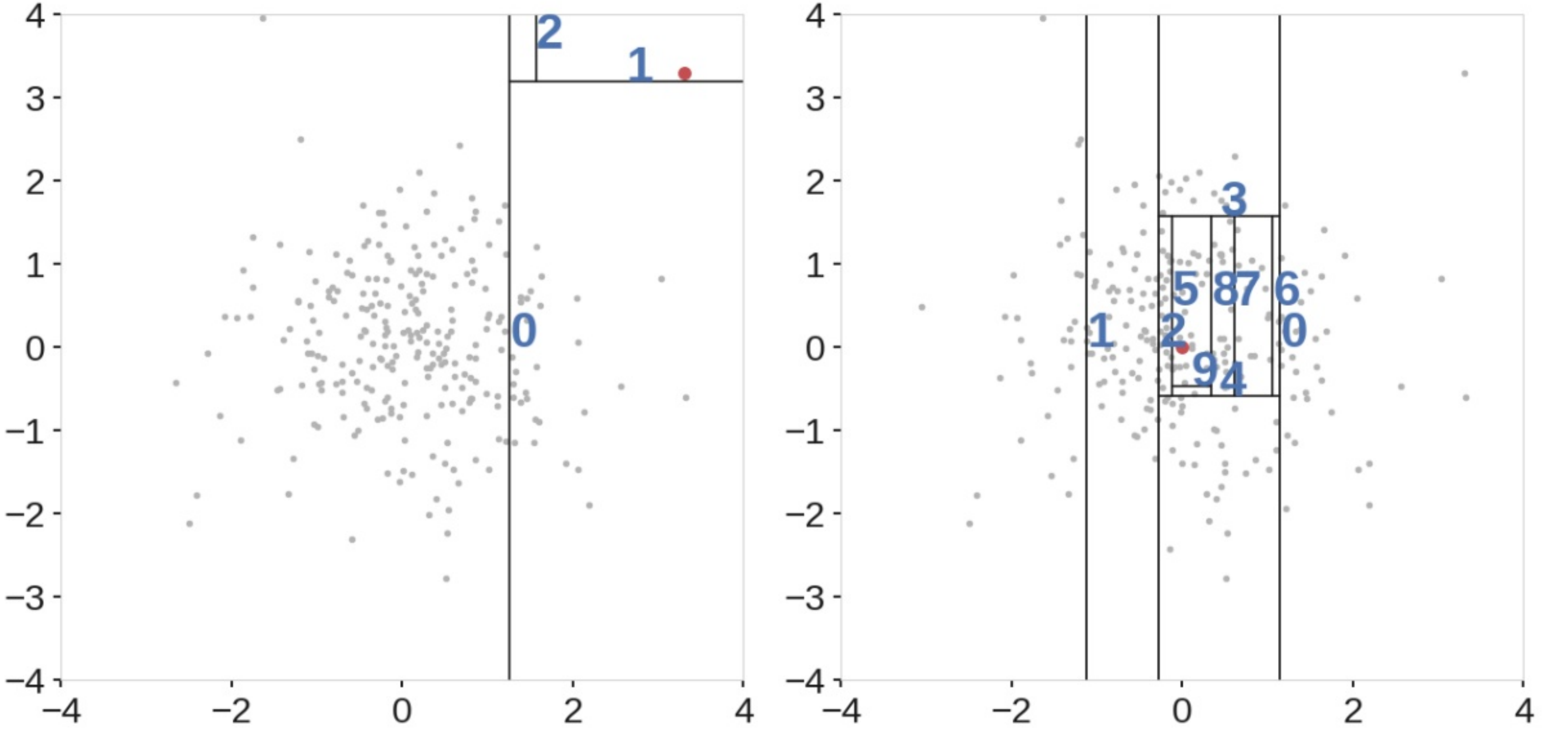
Stopping criteria for recursion

- 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

$$c(n) = 2H(n-1) - (2(n-1)/n)$$

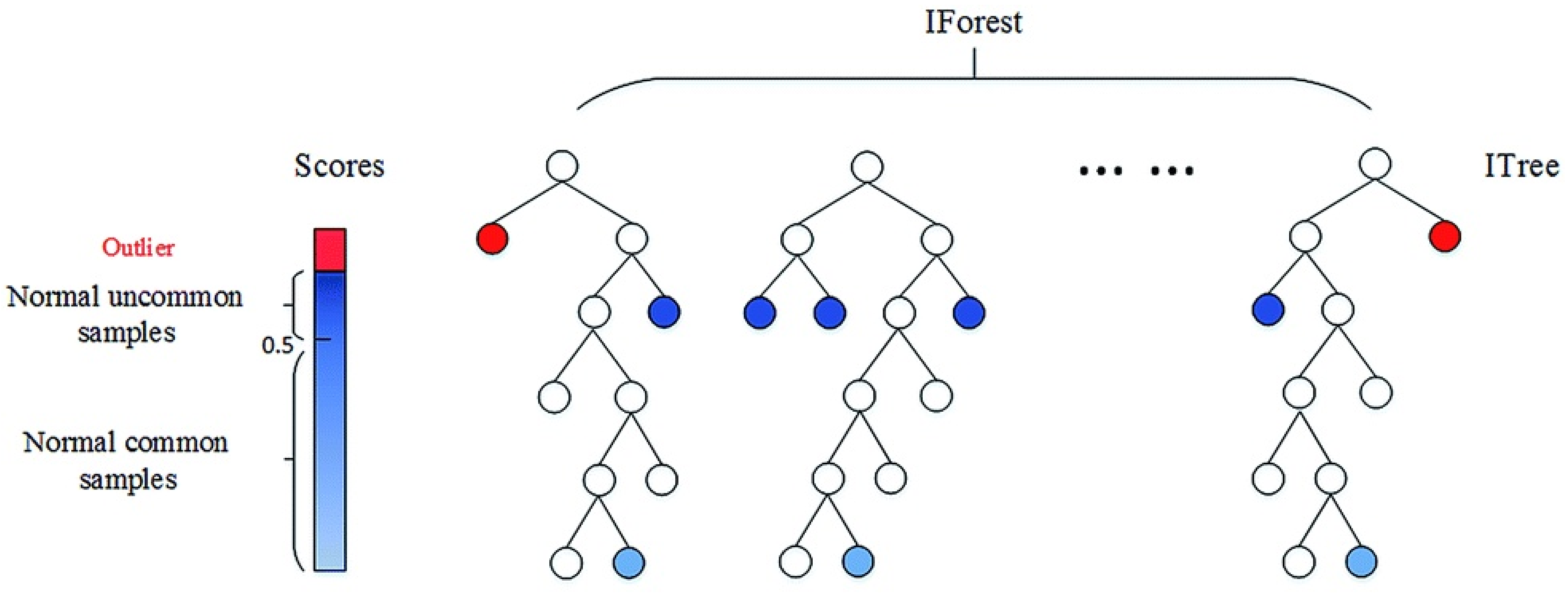
$$H(n) = \sum_{k=1}^{n} \frac{1}{k}$$

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

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

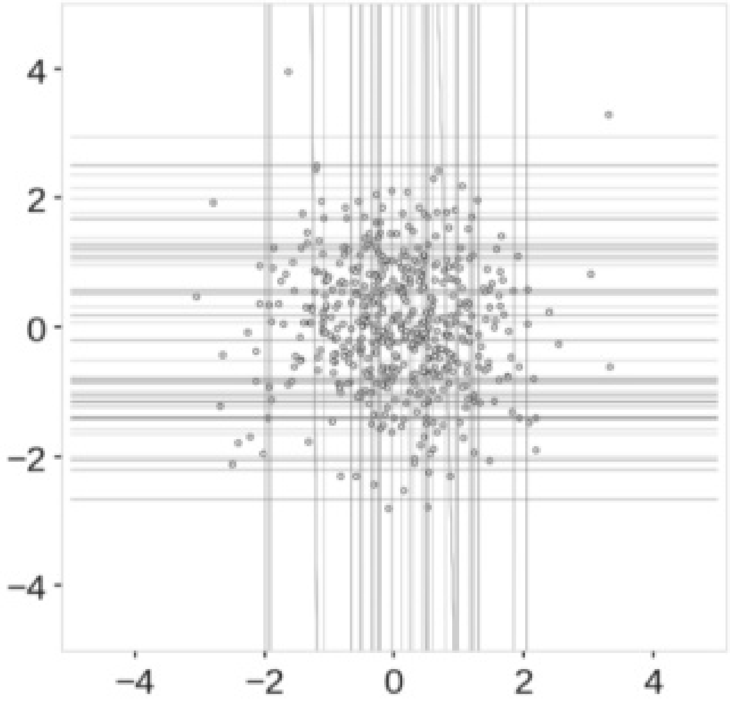
Outlier scores in isolation forests

(each tree is built from a sub-sample of original data)

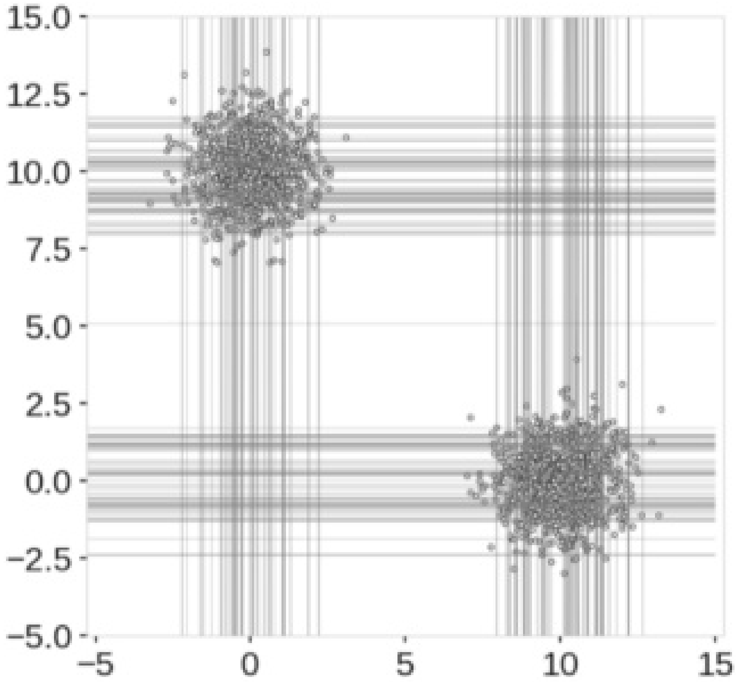


https://donghwa-kim.github.io/iforest.html

Example



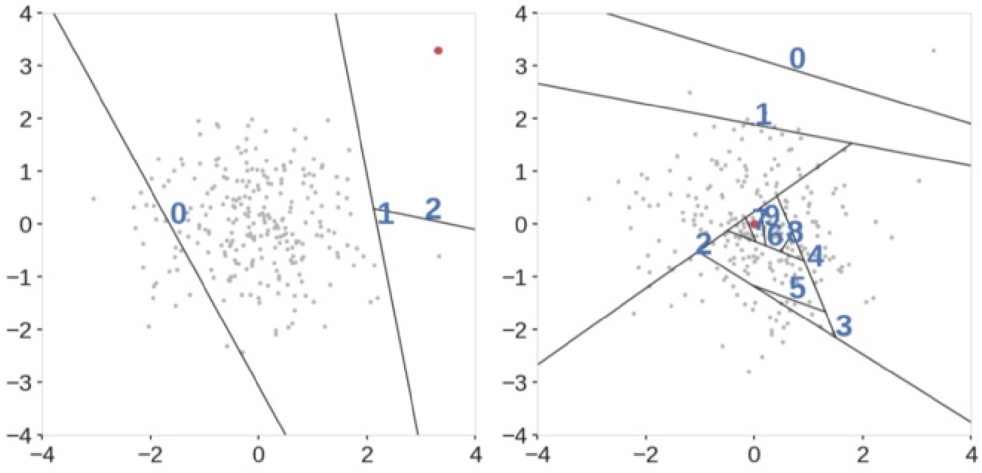
(a) Single blob



(b) Multiple Blobs

Extended Isolation Forest

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



(a) Anomaly (b) Nominal

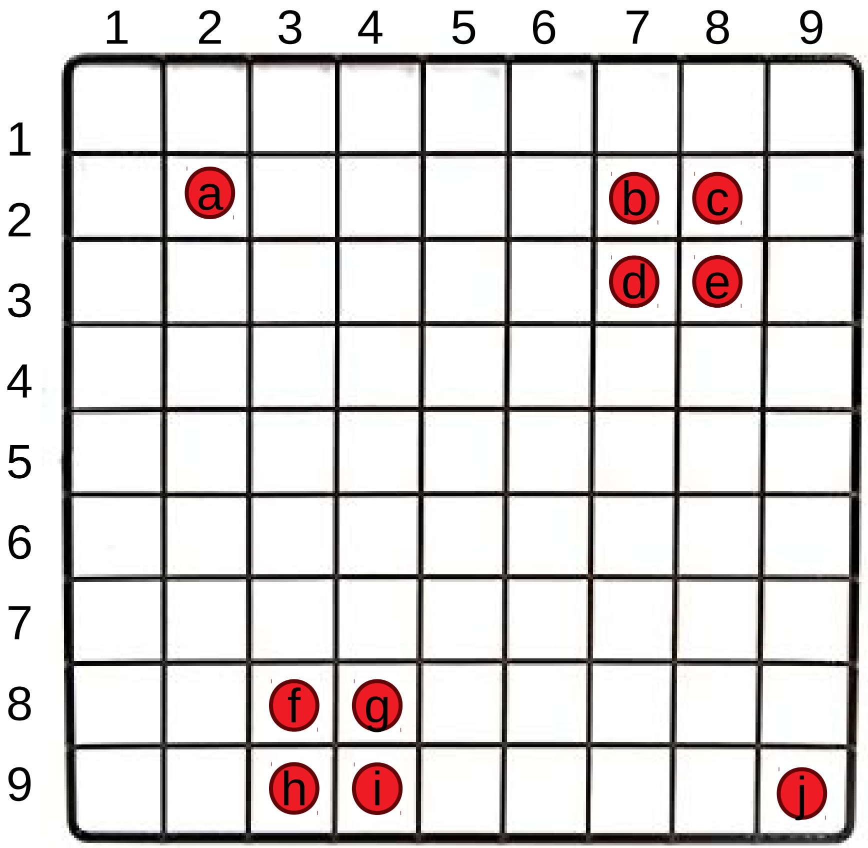
https://towardsdatascience.com/outlier-detection-with-extended-isolation-forest-1e248a3fe97b

Try it!

- Use a random number generator
- Dimensions of rows=1, cols=2
- Label each point with its score

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

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



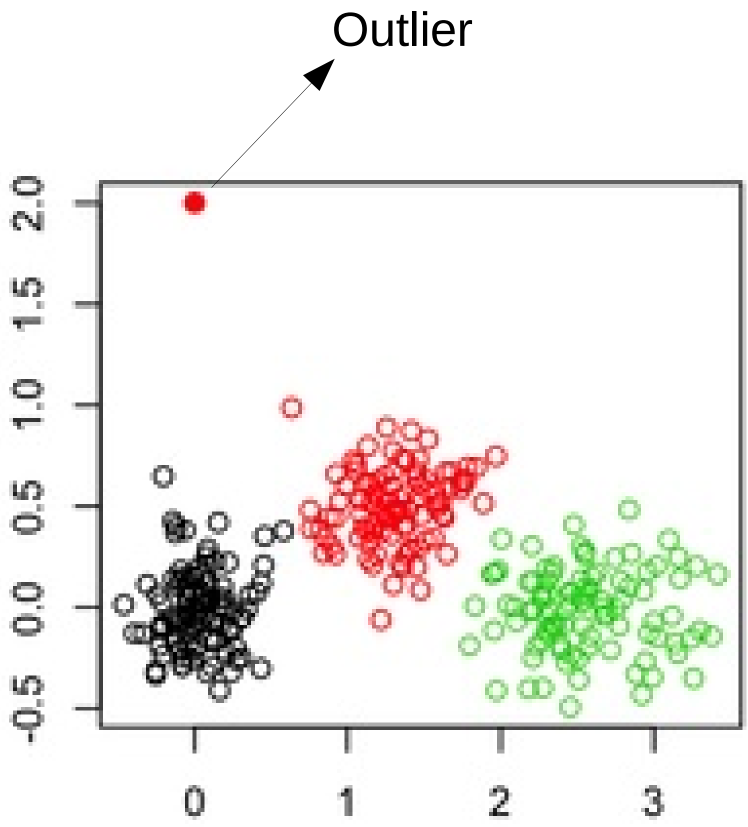
Clustering-based methods

Clustering for outlier analysis

- Clustering associate points to similar points
- Points either clearly belong to a cluster or are outliers
- Some clustering algorithms also detect outliers
  - Examples: DBSCAN, DENCLUE

Simple method

- Cluster data, associating each point to a centroid, e.g., using k-means
- Outlier score = distance of point to its centroid



Improved method

- Cluster data
- Outlier score = local Mahalanobis distance with respect to center of cluster r

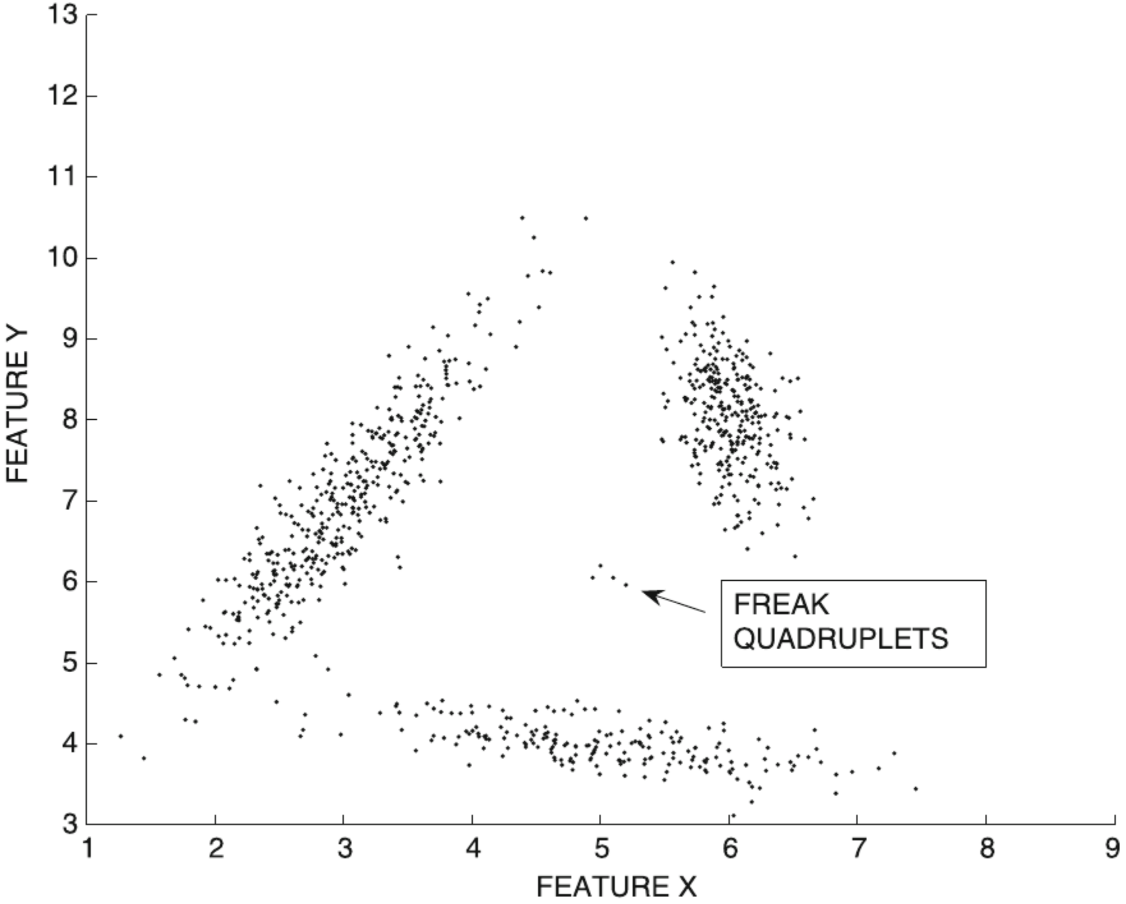
$$\mathrm{Maha}(\overline{X}, \overline{\mu_r}, \Sigma_r) = \sqrt{(\overline{X} - \overline{\mu_r})\Sigma_r^{-1}(\overline{X} - \overline{\mu_r})^T}$$

$\overline{\mu_r}$ is the mean of the cluster r

Σ_r is the covariance matrix of cluster r

Improved method (cont.)

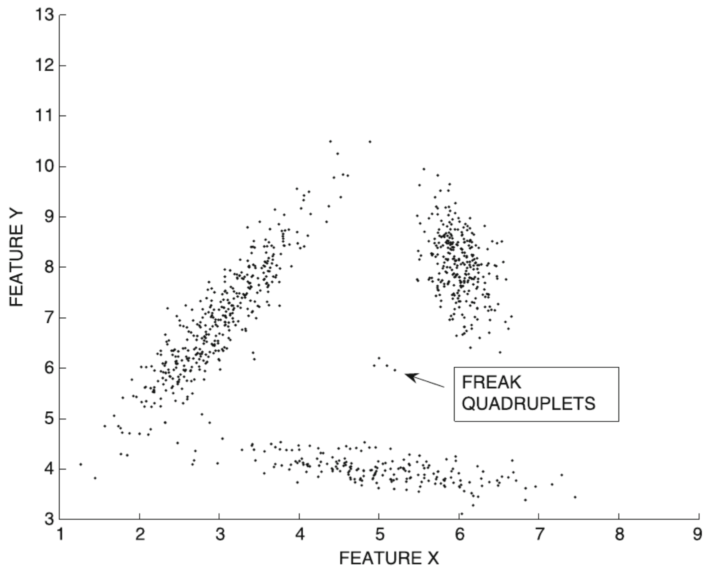
- Remove tiny clusters



Distance-based methods

Instance-specific definition

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

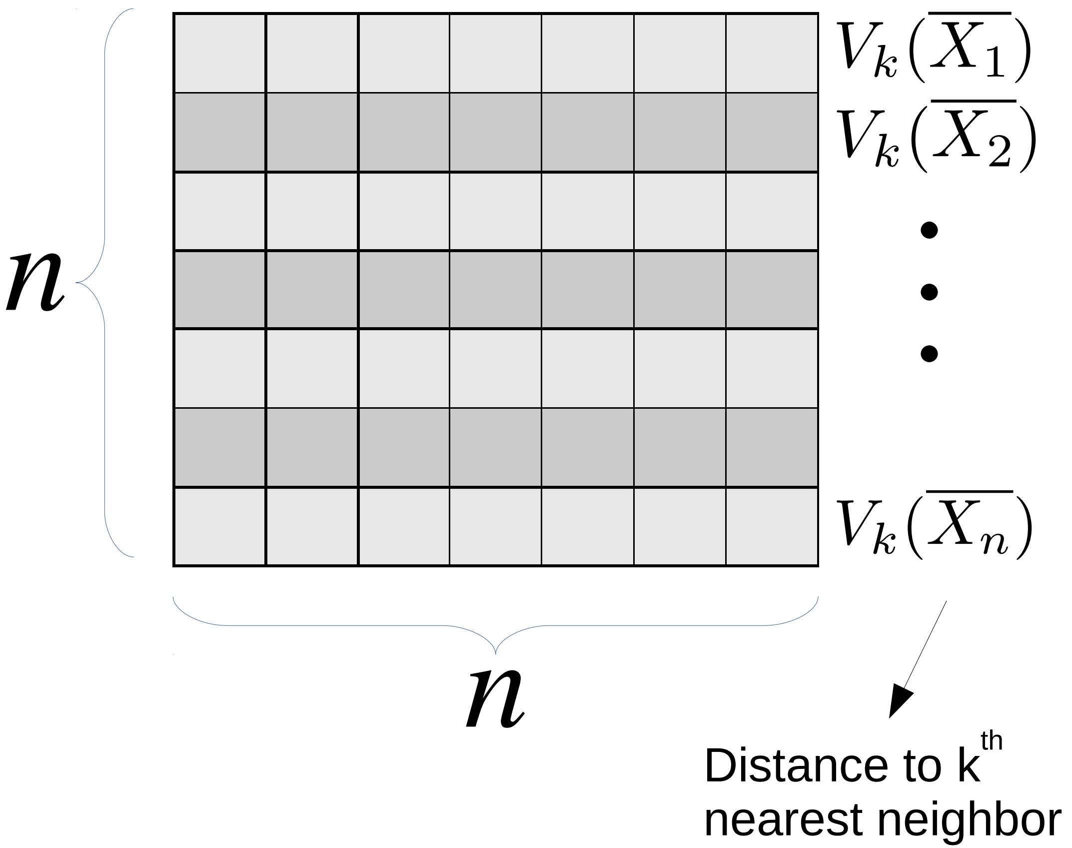


Problem: computational cost

- 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

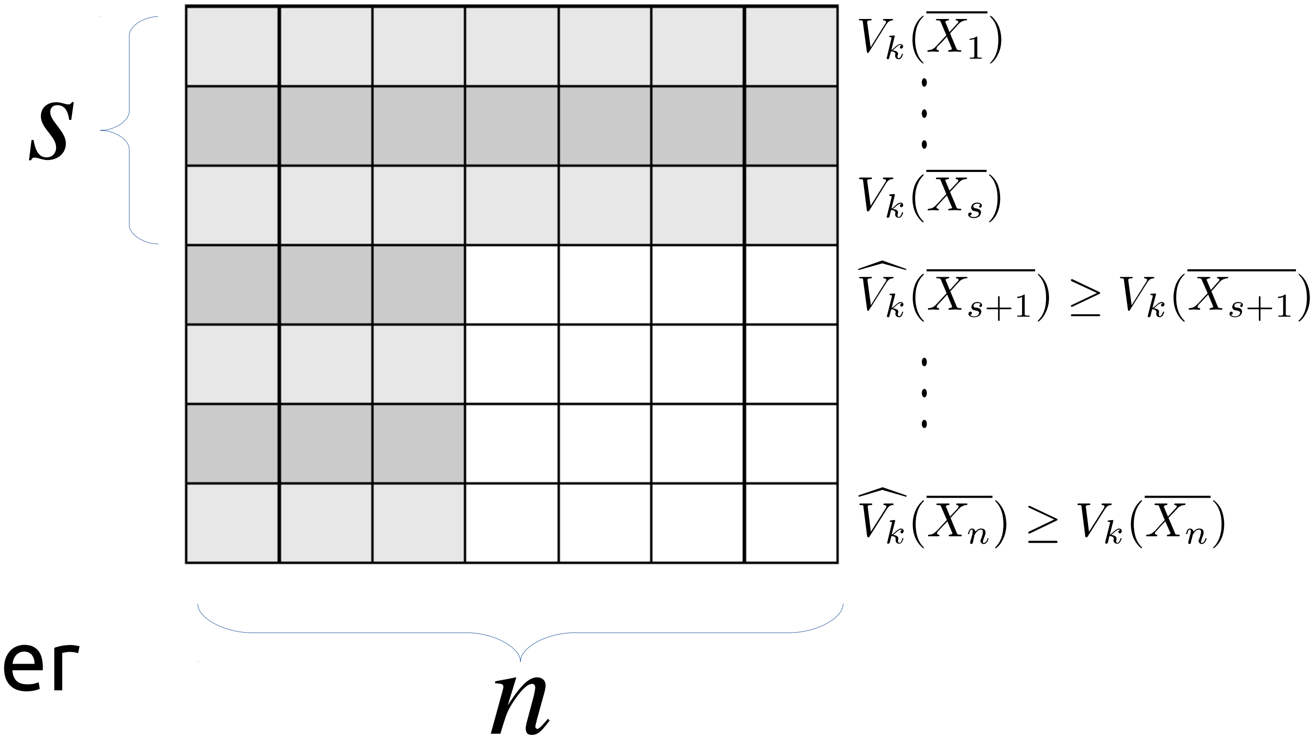
Problem: computational cost

- The distance-based outlier score of an item x is its distance to its k^{th} 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

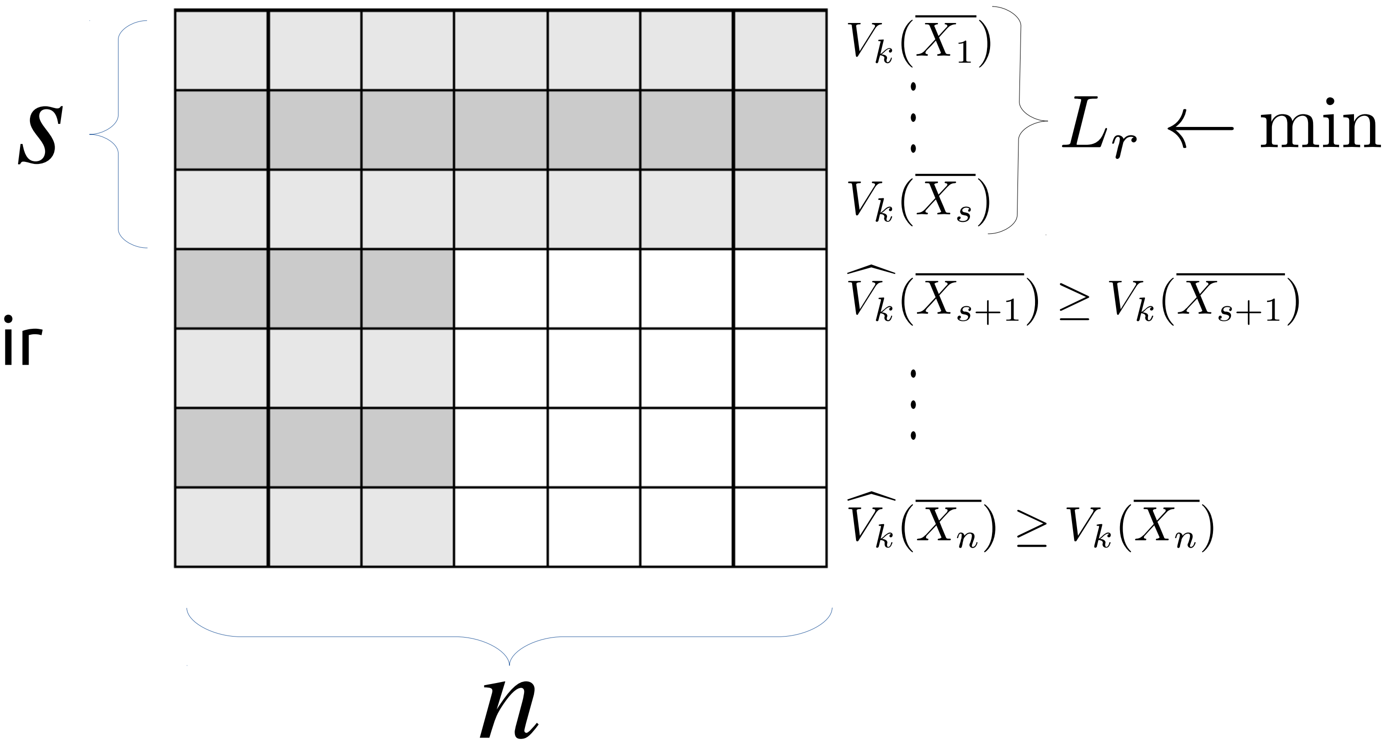


Pruning method: sampling (cont.)

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

($r \leq s$ nodes with the larger distance to their k^{th} nearest neighbor)

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

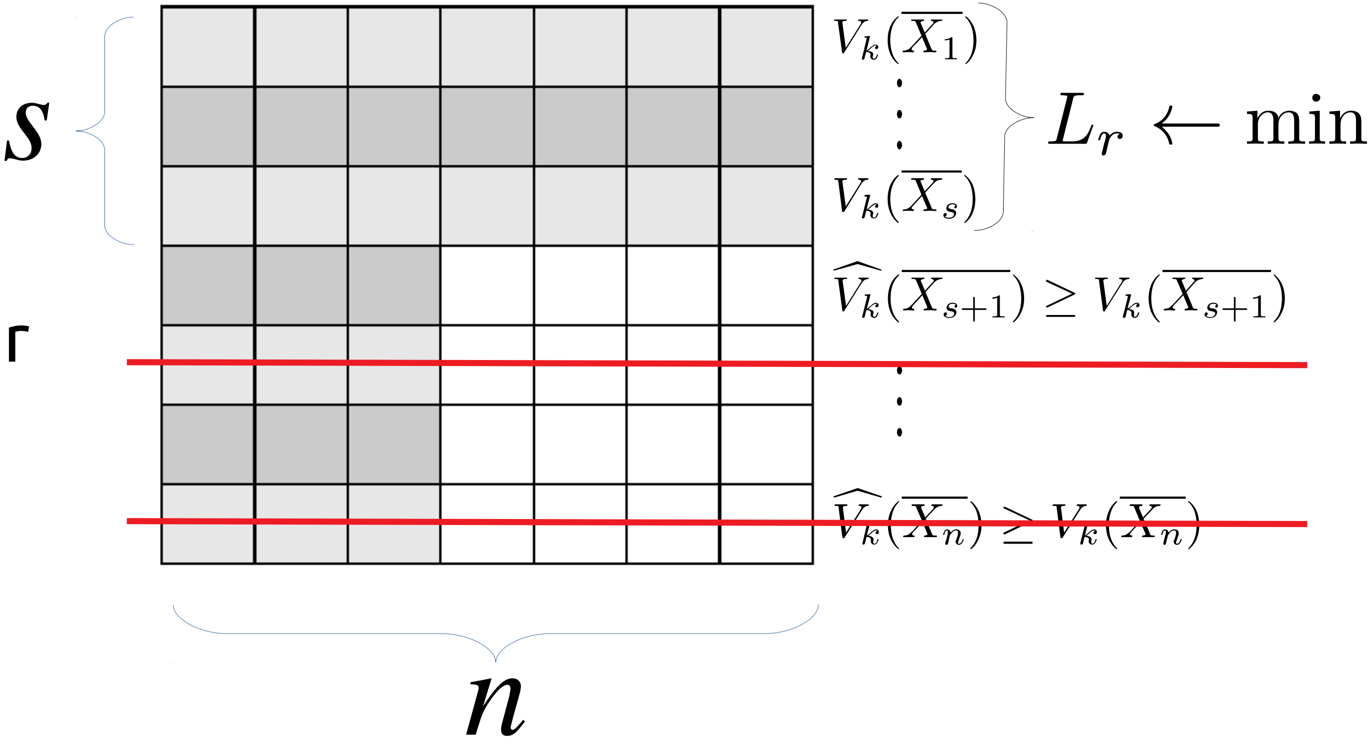


Pruning method: sampling (cont.)

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

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

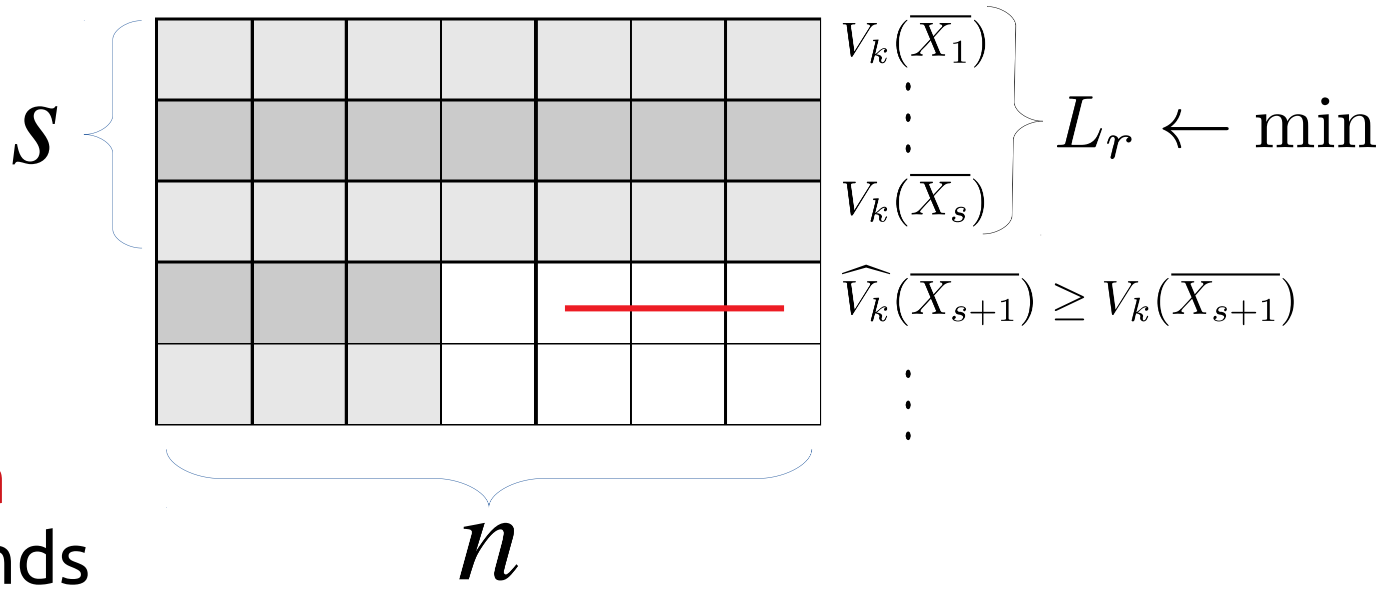
Any point having $V_k < L_s$ cannot be among the top r outliers



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

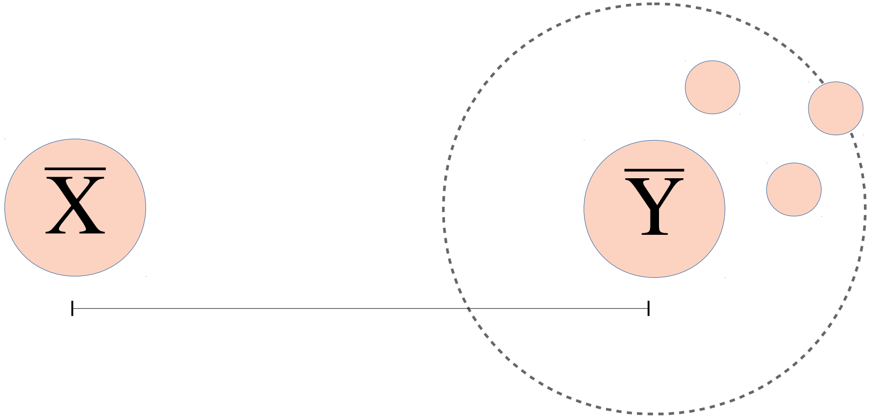
$$R_k(\overline{X}, \overline{Y}) = \max\{\mathrm{Dist}(\overline{X}, \overline{Y}), V_k(\overline{Y})\}$$

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.)

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

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

- Local outlier factor

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

Outlier score

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

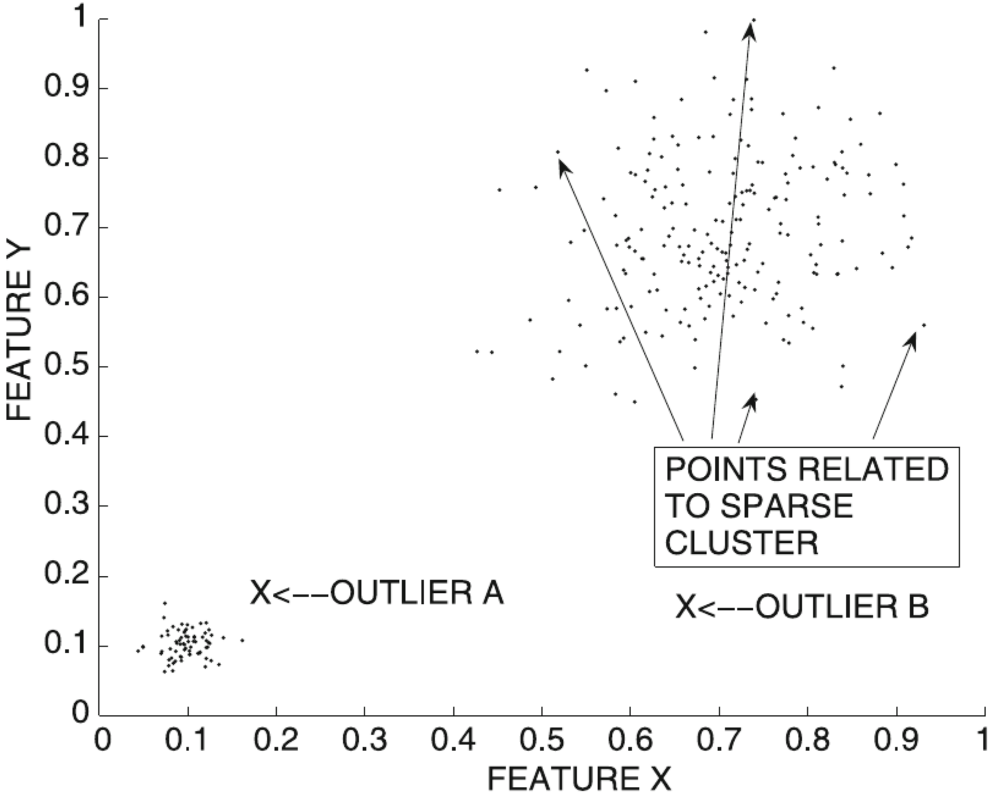
- Large for outliers, close to 1 for others

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: $V_2(u)$ = _____ ; $V_2(v)$ = _____; $V_2(a)$ = _____ ; $V_2(b)$ = _____

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

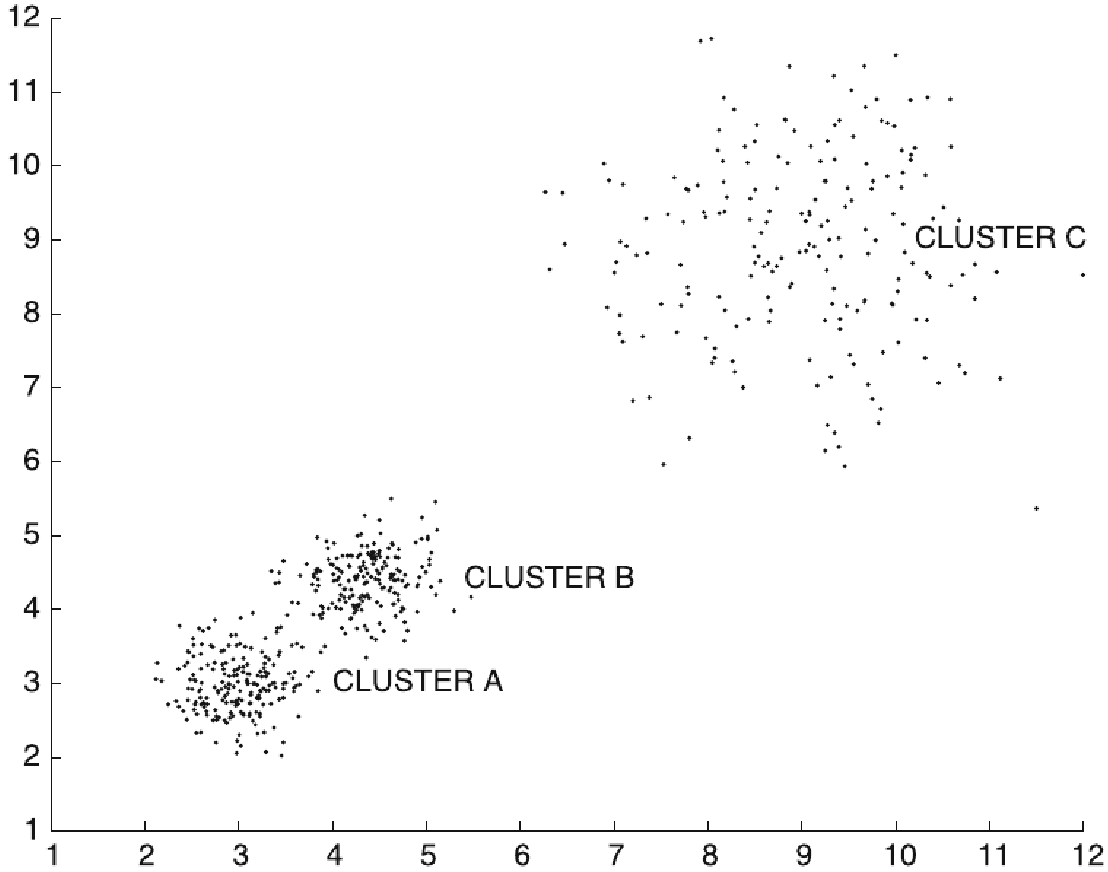
$$AR_k(\overline{X}) = \underset{\overline{Y} \in L_k(\overline{X})}{E} \left[R_k(\overline{X}, \overline{Y}) \right]$$

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

Density-based methods

Density-based methods

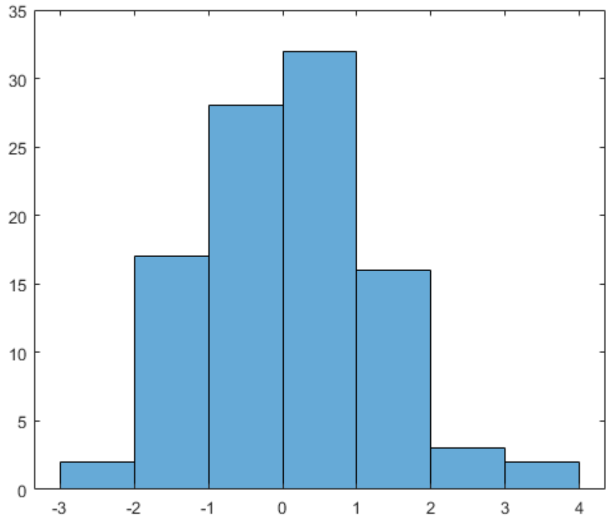
- Key idea:
  find sparse regions
  in the data
- Limitation:
  cannot handle
  variations of
  density



Histogram- and grid-based methods

Histogram-based method:

1) Put data into **bins**
2) Outlier score: *num* – 1, where *num* is the number of items in the same **bin**

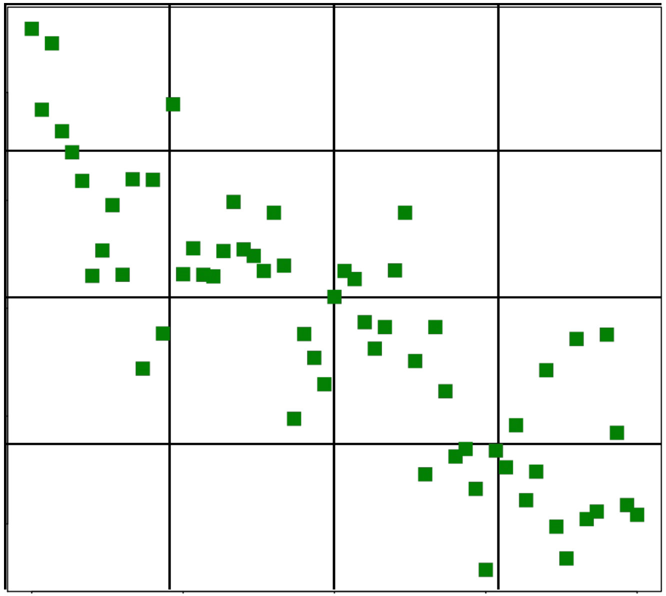


Clear outliers are alone or almost alone in a **bin**

Histogram- and grid-based methods

Grid-based method

1) Put data into a **grid**
2) Outlier score: *num* – 1, where *num* is the number of items in the same **cell**



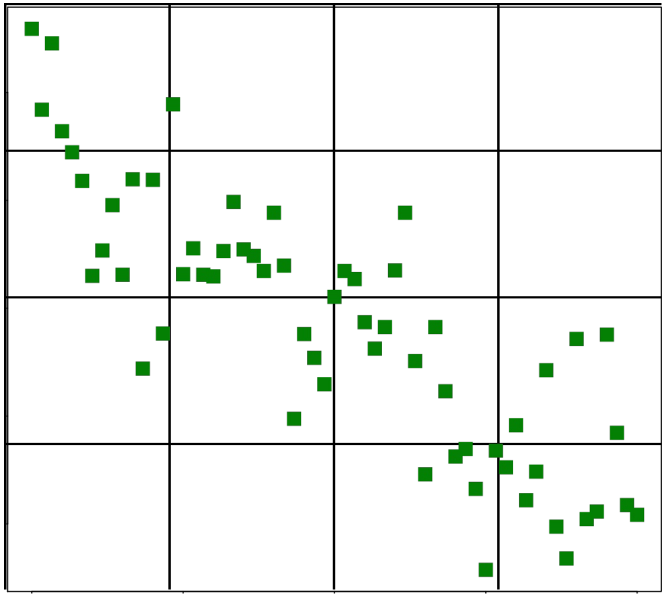
Clear outliers are alone or almost alone in a **cell**

Problems with grid-based methods

How to choose the grid size?

Grid size should be chosen considering data density, but density might vary across regions

If dimensionality is high, then most cells will be empty



Kernel-based methods

- Given n points $\overline{X_1}, \overline{X_2}, \ldots, \overline{X_n}$

$$f(\overline{X}) = \frac{1}{n} \sum_{i=1}^{n} K_h(\overline{X} - \overline{X_i})$$

- K_h is a function peaking at $\overline{X_i}$ with *bandwidth* h
- For instance, a Gaussian kernel:

$$K_h(\overline{X} - \overline{X_i}) = \left(\frac{1}{\sqrt{2\pi} \cdot h} \right)^d \cdot e^{-\|\overline{X} - \overline{X_i}\|^2 / (2h^2)}$$

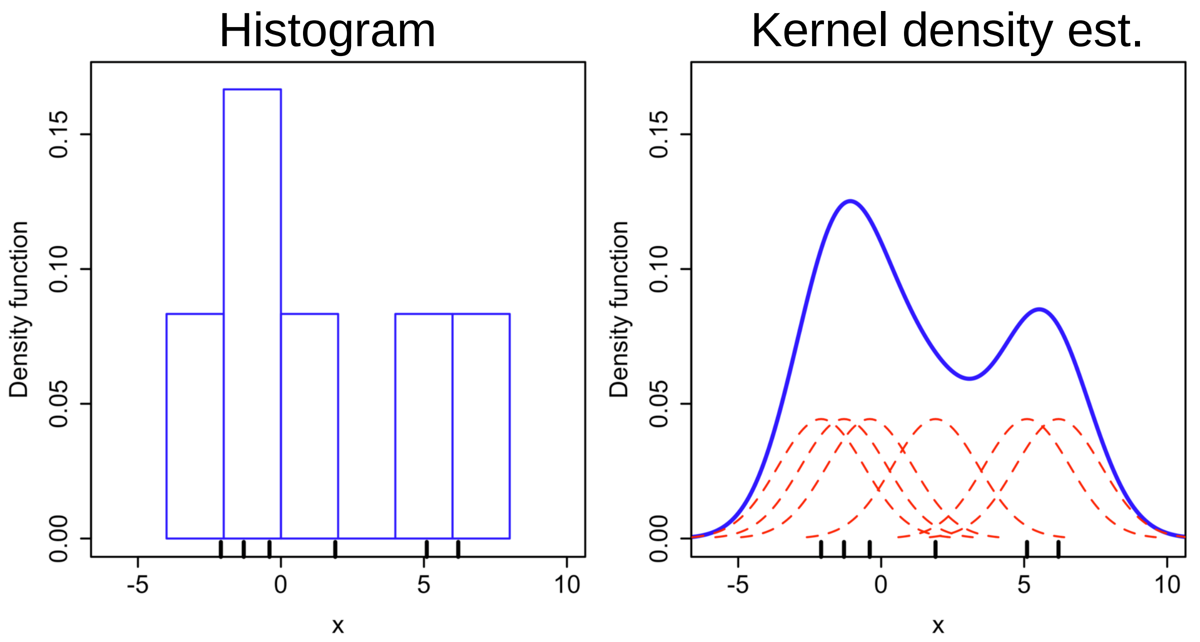
Kernel-based methods (cont.)

- Example with a Gaussian kernel

$$\overline{X} = < -2.1, -1.3, -0.4, 1.9, 5.1, 6.2 >$$

- Each K_h in **red**
- f = sum of K_h in **blue**

$$f(\overline{X}) = \frac{1}{n}\sum_{i=1}^{n} K_h(\overline{X} - \overline{X_i})$$



[Wikipedia: Kernel density estimation]

Information-theoretic models

- Describe “ABABABABABABABABABABABABABABABABAB”
  - “AB” 17 times
- Describe “ABABACABABABABABABABABABABABABABAB”
  - Minimum description length increased
- Information-theoretic models: learn a model, then look at increases in model size due to a data point

Evaluation
(outlier validity)

Internal (unsupervised) criteria

- Rarely used in outlier analysis
- For any method, a measure can be created that will favor that method (~*overfit*)
- Solution space is small
  - Maybe there is just one outlier, finding it or missing it makes all the difference between perfect and useless performance

External (supervised) criteria

- **Known outliers** from a synthetic dataset or **rare items** (e.g., belonging to smallest class)
- Suppose D is the data, G are the real outliers, and S(t) are found when threshold t is used

$$TPR(t) = \mathrm{Recall}(t) = 100 \cdot \frac{|S(t) \cap G|}{|G|}$$

$$FPR(t) = 100 \cdot \frac{|S(t) - G|}{|D - G|}$$

ROC curve = $(FPR(t), TPR(t))_t$

Rank of ground-truth outliers:

- Algorithm A

  1, 5, 8, 15, 20

- Algorithm B

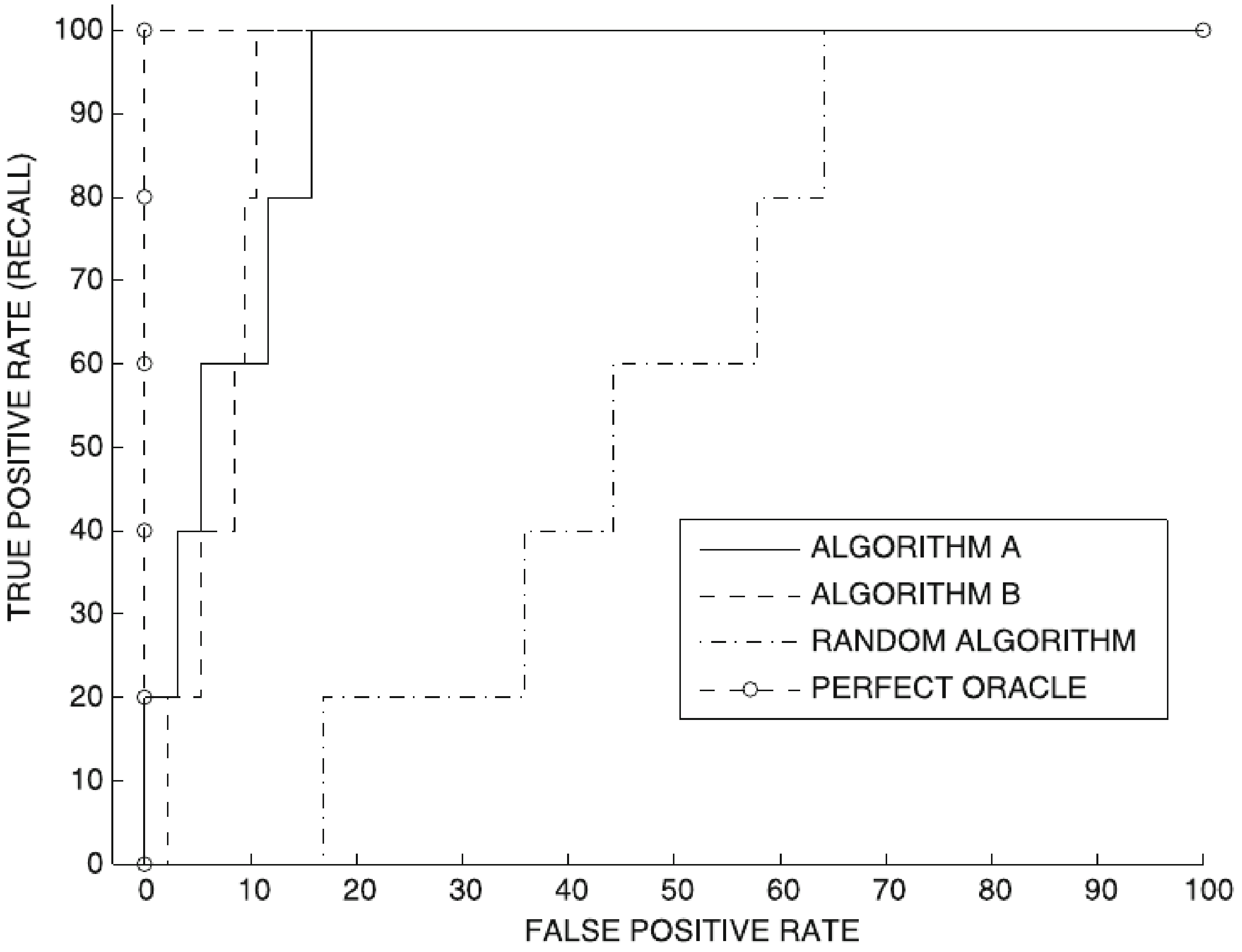
  3, 7, 11, 13, 15

- Random

  17, 36, 45, 59, 66

- Perfect oracle

  1, 2, 3, 4, 5



Summary

Things to remember

- Extreme value analysis
  - Univariate, multivariate, depth based
- Isolation forest
- Clustering-based methods
- Distance-based methods
- Density-based methods
- Outlier validity

Exercises for this topic

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