Anomaly & Outliers Detection

UNIVERSITÀ DI PISA

What is an Outlier?

- Anomaly is a pattern in the data that does not conform to the expected behaviour (also referred to as outlier/exception)

Definition of Hawkins [Hawkins 1980]:

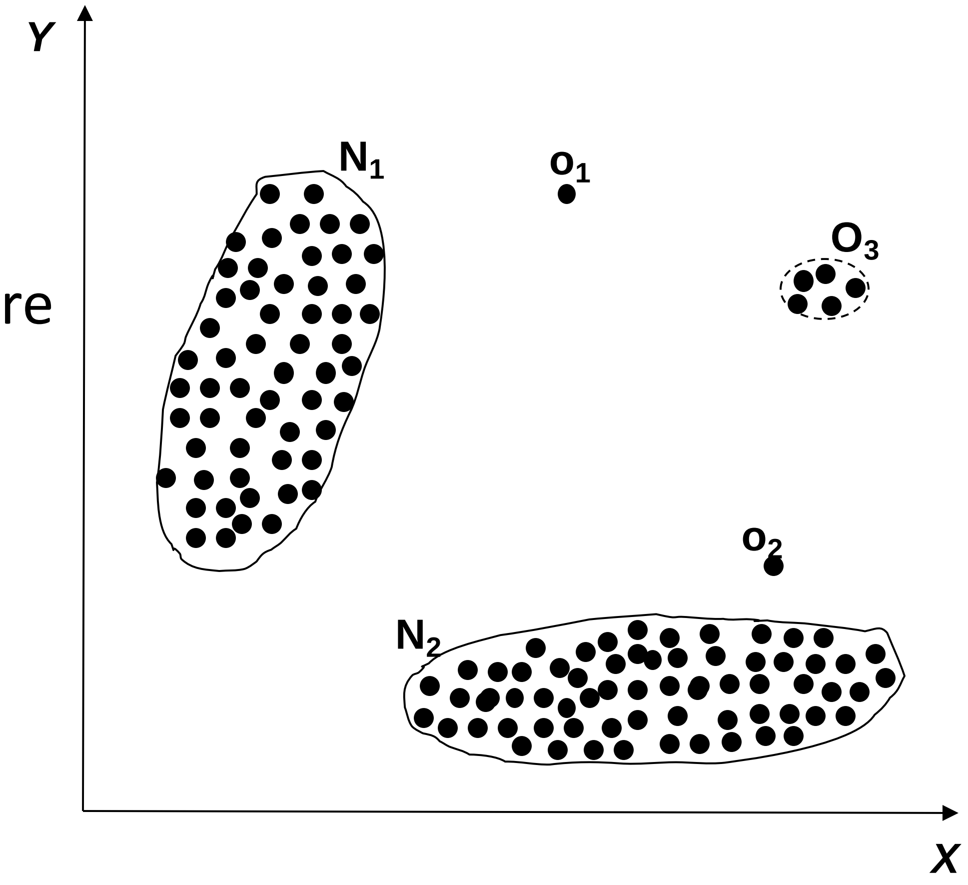
- “An outlier is an observation which deviates so much from the other observations as to arouse suspicions that it was generated by a different mechanism”

Statistics-based intuition

- Normal data objects follow a “generating mechanism”, e.g. some given statistical process
- **Abnormal objects** deviate from this generating mechanism

Anomaly/Outlier Detection

- What are anomalies/outliers?
  - The set of data points that are considerably different than the remainder of the data
- Natural implication is that anomalies are relatively rare
  - One in a thousand occurs often if you have lots of data
  - Context is important, e.g., freezing temps in July
- Can be important or a nuisance
  - 10 foot tall 2 years old
  - Unusually high blood pressure



Applications of Outlier Detection

- Fraud detection
  - Purchasing behavior of a credit card owner usually changes when the **card is stolen**
  - Abnormal buying patterns can characterize **credit card abuse**
- Medicine
  - Unusual symptoms or test results may indicate **potential health problems** of a patient
  - Whether a particular test **result is abnormal** may depend on other characteristics of the patients (e.g. gender, age, …)
- Public health
  - The **occurrence of a particular disease**, e.g. tetanus, scattered across various hospitals of a city indicate **problems** with the corresponding **vaccination** program in that city

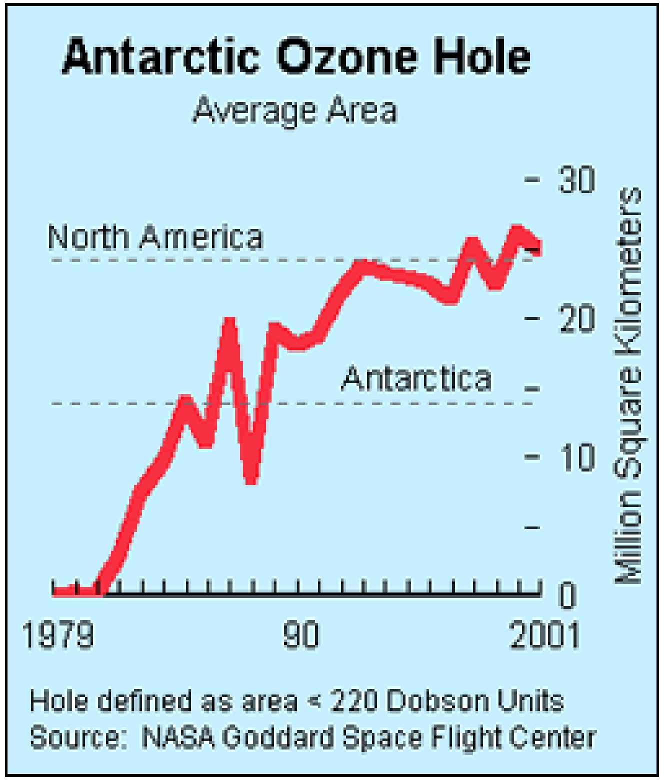
Importance of Anomaly Detection

Ozone Depletion History

- In 1985 three researchers (Farman, Gardinar and Shanklin) were puzzled by data gathered by the British Antarctic Survey showing that ozone levels for Antarctica had dropped 10% below normal levels

- **Why did the Nimbus 7 satellite, which had instruments aboard for recording ozone levels, not record similarly low ozone concentrations?**

- The ozone concentrations recorded by the satellite were so low they were being **treated as outliers** by a computer program and discarded!



Causes of Anomalies

- Data from different classes
  - Measuring the weights of oranges, but a few grapefruit are mixed in
- Natural variation
  - Unusually tall people
- Data errors/ Data Measurement and Collection Errors
  - 200 pound 2 year old

Distinction Between Noise and Anomalies

- Noise is erroneous, perhaps random, values or contaminating objects
  - Weight recorded incorrectly
  - Grapefruit mixed in with the oranges

- Noise doesn’t necessarily produce unusual values or objects
- Noise is not interesting
- Anomalies may be interesting if they are not a result of noise
- Noise and anomalies are related but distinct concepts

General Issues: Number of Attributes

- Many anomalies are defined in terms of a single attribute
  - Height
  - Shape
  - Color

- Can be hard to find an anomaly using all attributes
  - Noisy or irrelevant attributes
  - Object is only anomalous with respect to some attributes

- However, an object may not be anomalous in any one attribute

General Issues: Anomaly Scoring

- Many anomaly detection techniques provide only a binary categorization
  - An object is an anomaly, or it isn’t
  - This is especially true of classification-based approaches

- Other approaches assign a score to all points
  - This score measures the degree to which an object is an anomaly
  - This allows objects to be ranked

- In the end, you often need a binary decision
  - Should this credit card transaction be flagged?
  - Still useful to have a score
- How many anomalies are there?

Other Issues for Anomaly Detection

- Find all anomalies at once or one at a time
  - **Swamping**: normal objects are classified as outliers
  - **Masking**: the presence of several anomalies masks the presence of all

- **Evaluation**
  - How do you measure performance?
  - Supervised vs. unsupervised situations

- **Efficiency**
  - Classification models – expensive for learning the model and inexpensive to apply the mode
  - Proximity-based approaches – cost of the proximity matrix

- **Context**: global versus local perspective
  - A person is unusually tall w.r.t. the general population but not w.r.t. professional basketball team

Variants of Anomaly Detection Problems

- Given a data set D, find all data points **x** $\in$ D with anomaly scores greater than some threshold *t*

- Given a data set D, find all data points **x** $\in$ D having the top-*n* largest anomaly scores

- Given a data set D, containing mostly normal (but unlabeled) data points, and a test point **x**, compute the anomaly score of **x** with respect to D

Model-Based Anomaly Detection

Build a model for the data and see

- Unsupervised
  - Anomalies are those points that don’t fit well
  - Anomalies are those points that distort the model
  - Examples:
    - Statistical distribution
    - Clusters
- Supervised
  - Anomalies are regarded as a rare class
  - Need to have training data

Machine Learning for Outlier Detection

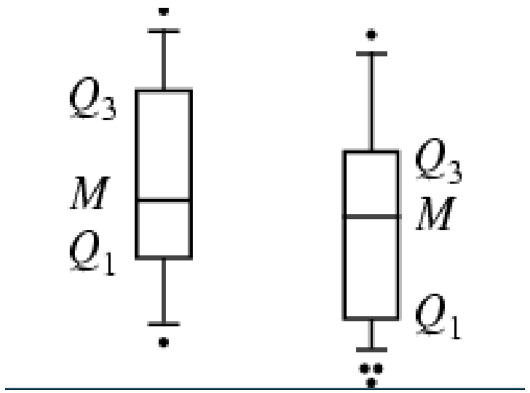
- If the ground truth of anomalies is available we can prepare a classification problem to unveil outliers.
- As classifiers we can use all the available machine learning approaches: Ensembles, SVM, DNN.
- The problem is that the dataset would be very unbalanced
- Thus, ad-hoc formulations/implementation should be adopted.

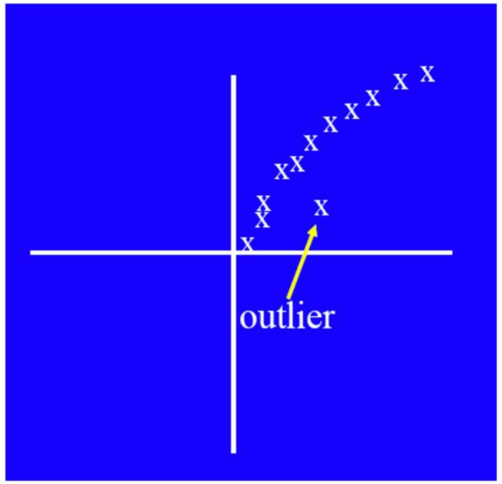
Additional Anomaly Detection Techniques

- **Proximity-based**
  - Anomalies are points far away from other points
  - Can detect this graphically in some cases
- **Density-based**
  - Low density points are outliers
- **Pattern matching**
  - Create profiles or templates of atypical but important events or objects
  - Algorithms to detect these patterns are usually simple and efficient

Graphical Approches

Boxplot (1-D), Scatter plot (2-D)





Limitation: It is Time Consuming

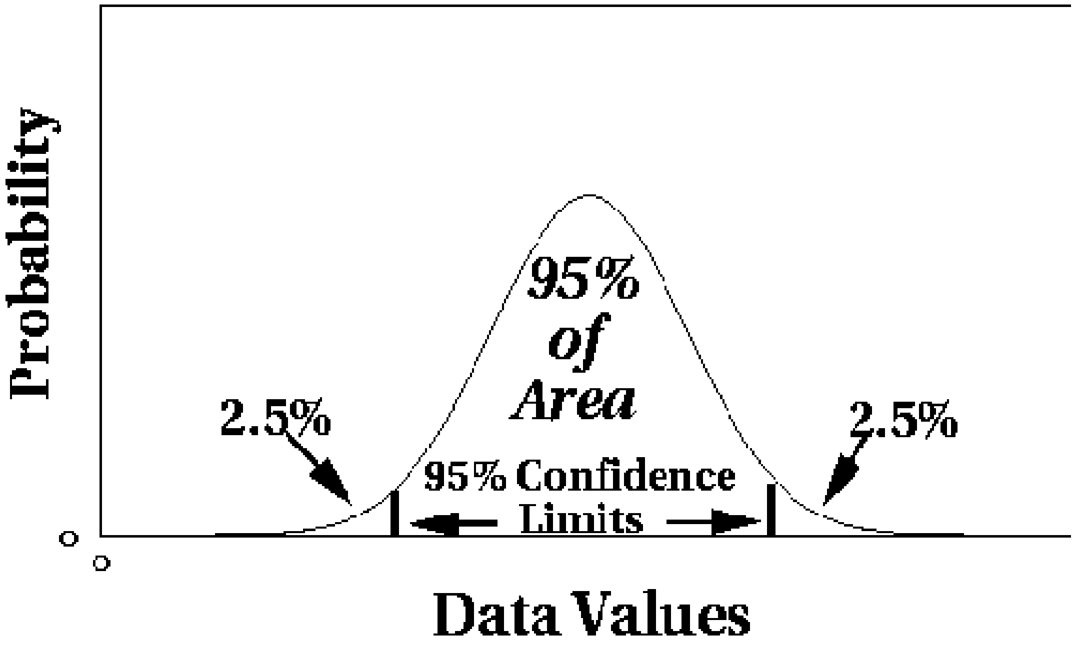
Statistical Approaches

Statistical Approaches

Probabilistic definition of an outlier: An outlier is an object that has a low probability with respect to a probability distribution model of the data.

- Usually **assume a parametric model** describing the distribution of the data (e.g., normal distribution)
- Apply a statistical test that depends on
  - Data distribution
  - Parameters of distribution (e.g., mean, variance)
  - Number of expected outliers (confidence limit)
- Issues
  - Identifying the distribution of a data set
    - Heavy tailed distribution
  - Number of attributes: most of these approaches are applicable to single attributes
  - Is the data a mixture of distributions?

Normal Distributions



One-dimensional Gaussian

The distance of a value x from the center of a N(0,1) distribution is directly related to the prob(x)

- Low probability for values in the tails
- **A data point x is an Outlier if |x| > c and prob(|x|>c)=α (when c increas and α decreases)**
- We can apply this method on z-score values
- **α should be specified to use this method**

Interquartile Range

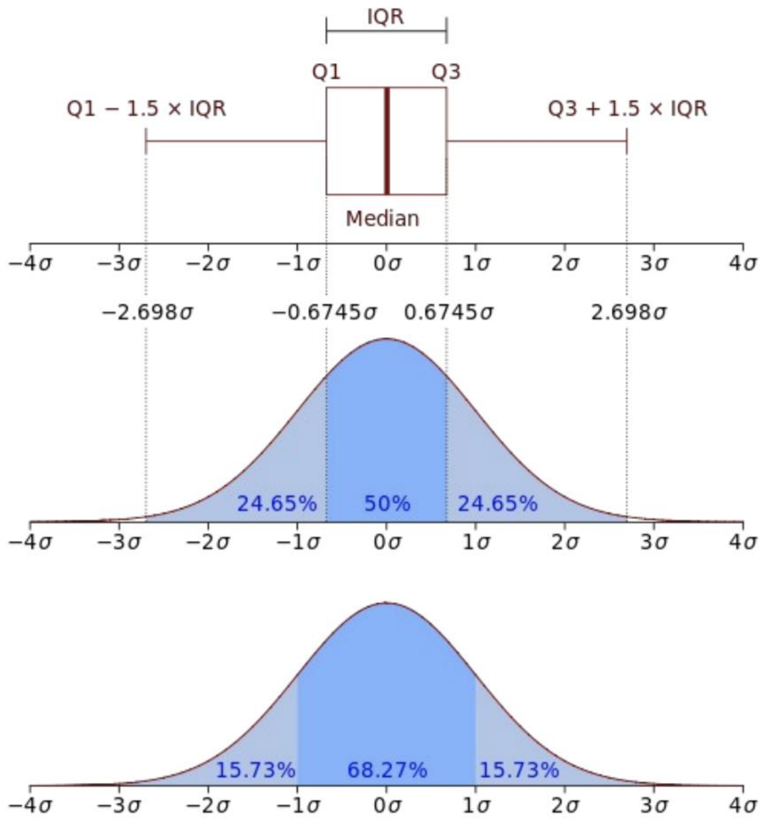
- Divides data in quartiles
  - Q1: first quartile
  - Q3: third quartile

Definitions:
- IQR = Q3-Q1
- **Outlier detection:**
  - **All x values outside** [median-1.5*IQR ; median+1.5*IQR]

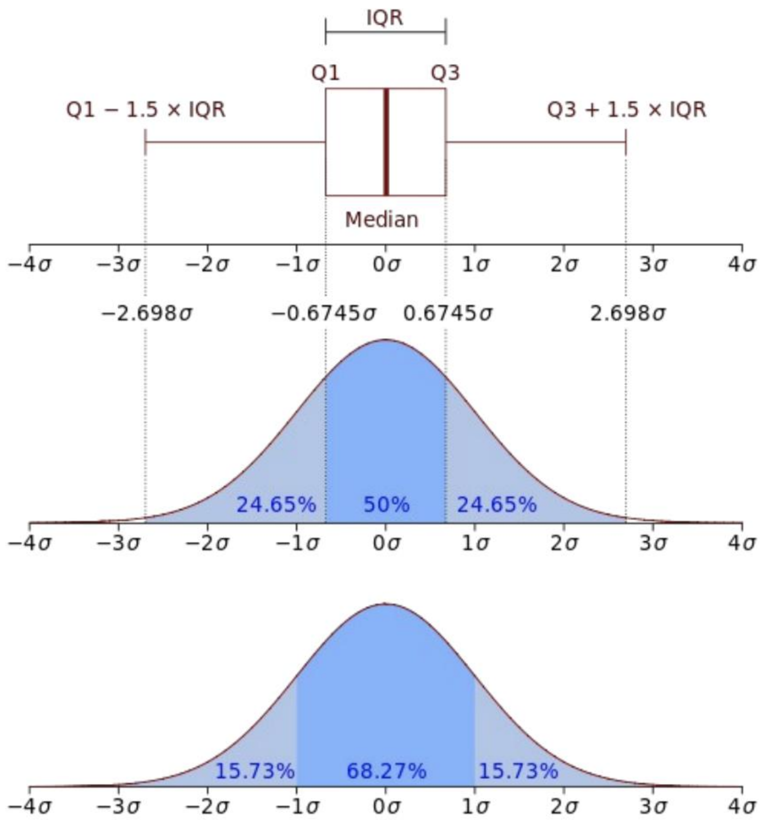
Example:

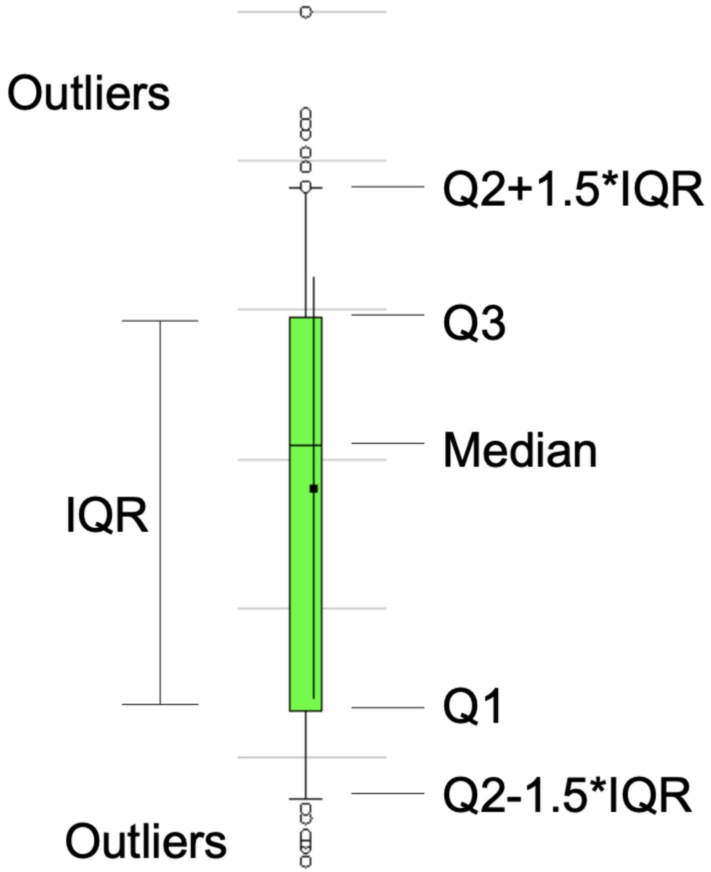
X = 0,1,1,3,3,5,7,42

- **Median= 3**, **Q1=1**, **Q3=7** → **IQR = 6**
- Allowed interval: [3 - 1.5*6 ; 3+1.5*6] = [-6 ; 12]
- Thus, 42 is an outlier



IQR vs Box Plots





Median Absolute Deviation (MAD)

- MAD is the median deviation from the median of a sample, i.e.

$$MAD := median_i(X_i - median_j(X_j))$$

- MAD can be used for outlier detection
- all values that are k*MAD away from the median are considered to be outliers
- e.g., k=3

Example

X= 0,1,1,3,5,7,42

- Median = 3, Deviations: 3,2,2,0,2,4,39 → MAD = 2
- allowed interval: [3-3*2 ; 3+3*2] = [-3;9]
- therefore, 42 is an outlier

Statistical-based – Grubbs' Test

- Detect outliers in univariate data
- Assume data comes from normal distribution
- Detects one outlier at a time, remove the outlier, and repeat
  - H_0: There is no outlier in data
  - H_A: There is at least one outlier
- Grubbs' test statistic:

mean

$$G = \frac{\max \left| X - \overline{X} \right|}{s}$$

std dev

the z-score of largest magnitude

- Reject H0 at **significance level** α if:

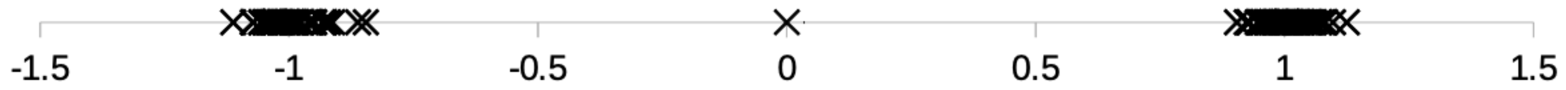
$$G > \frac{(N-1)}{\sqrt{N}} \sqrt{\frac{t^2_{(\alpha/N, N-2)}}{N - 2 + t^2_{(\alpha/N, N-2)}}}$$

the upper critical value of the **t-distribution** with N – 2 degrees of freedom and a significance level of α/(2N).

Outliers vs. Extreme Values

So far, we have looked at extreme values only

- But outliers can occur as non-extremes
- Methods presented until now are able to detect 0 as an outlier?
    - In that case, methods like IQR fails



Strengths/Weaknesses of Statistical Approaches

Pros

- Firm mathematical foundation
- Can be very efficient
- Good results if distribution is known

Cons

- In many cases, data distribution may not be known
- For high dimensional data, it may be difficult to estimate the true distribution
- Anomalies can distort the parameters of the distribution
    - Mean and standard deviation are very sensitive to outliers

Distance-based Approaches

Distance-based Approaches

- General Idea
  - Judge a point based on the distance(s) to its neighbors
  - Several variants proposed
- Basic Assumption
  - Normal data objects have a dense neighborhood
  - Outliers are far apart from their neighbors, i.e., have a less dense neighborhood

Distance-based Approaches

- Several different techniques

- **Approach 1:** The outlier score of an object is the distance to its *k*-th nearest neighbor

- **Approach 2:** An object is an outlier if a specified fraction of the objects is more than a specified distance away (Knorr, Ng 1998)

Distance-based Approaches

Definition of Outlier:

Proximity-based definition of outlier using distance to k-nearest neighbor

Anomaly score function:

Given a data instance x from a dataset D and a value k

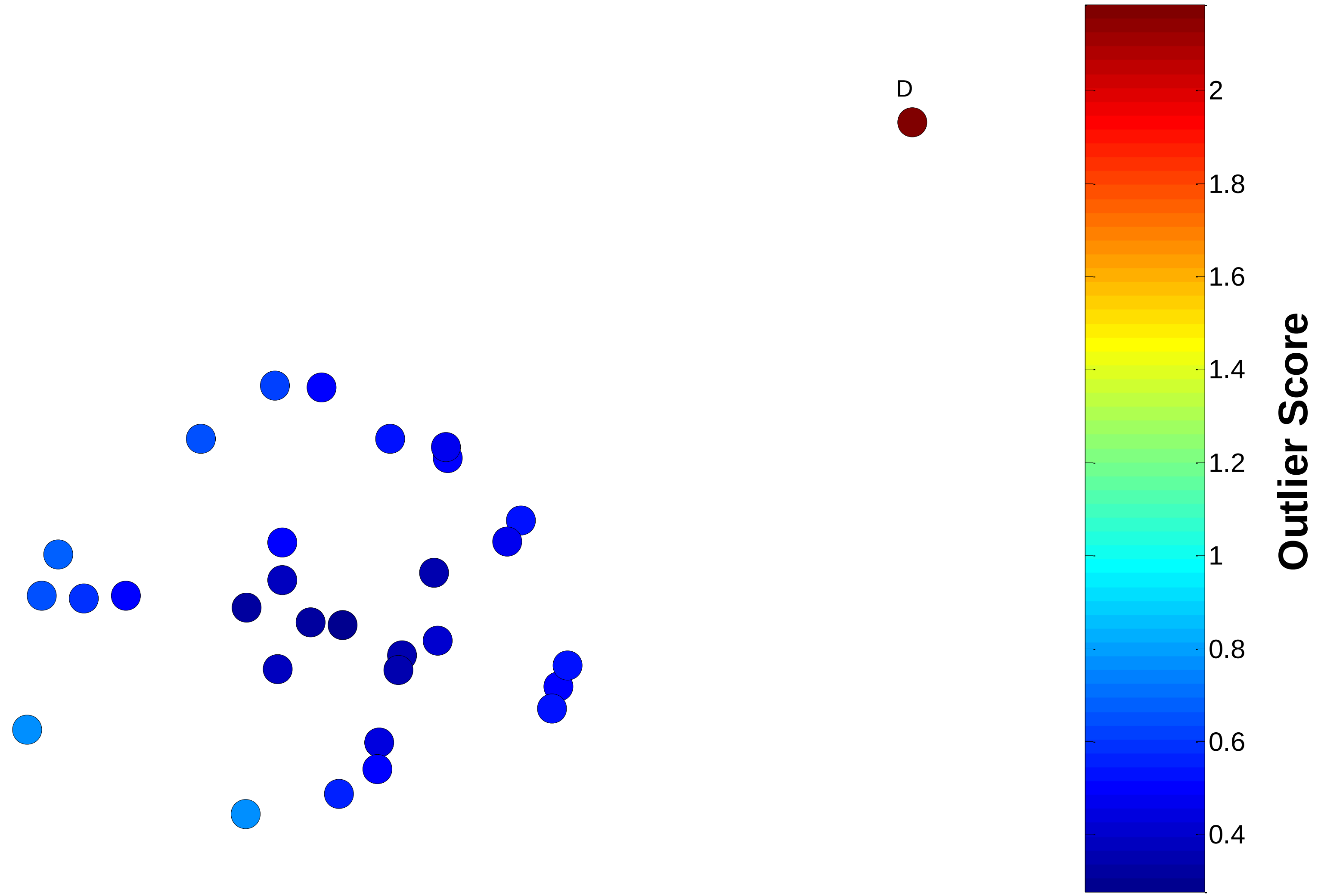
- f(x) = Distance between x and its k-nearest neighbor
- f(x) = Average distance between x and its k-nearest neighbors (less sensitive to k small or large)

How does the approach work? (in general):

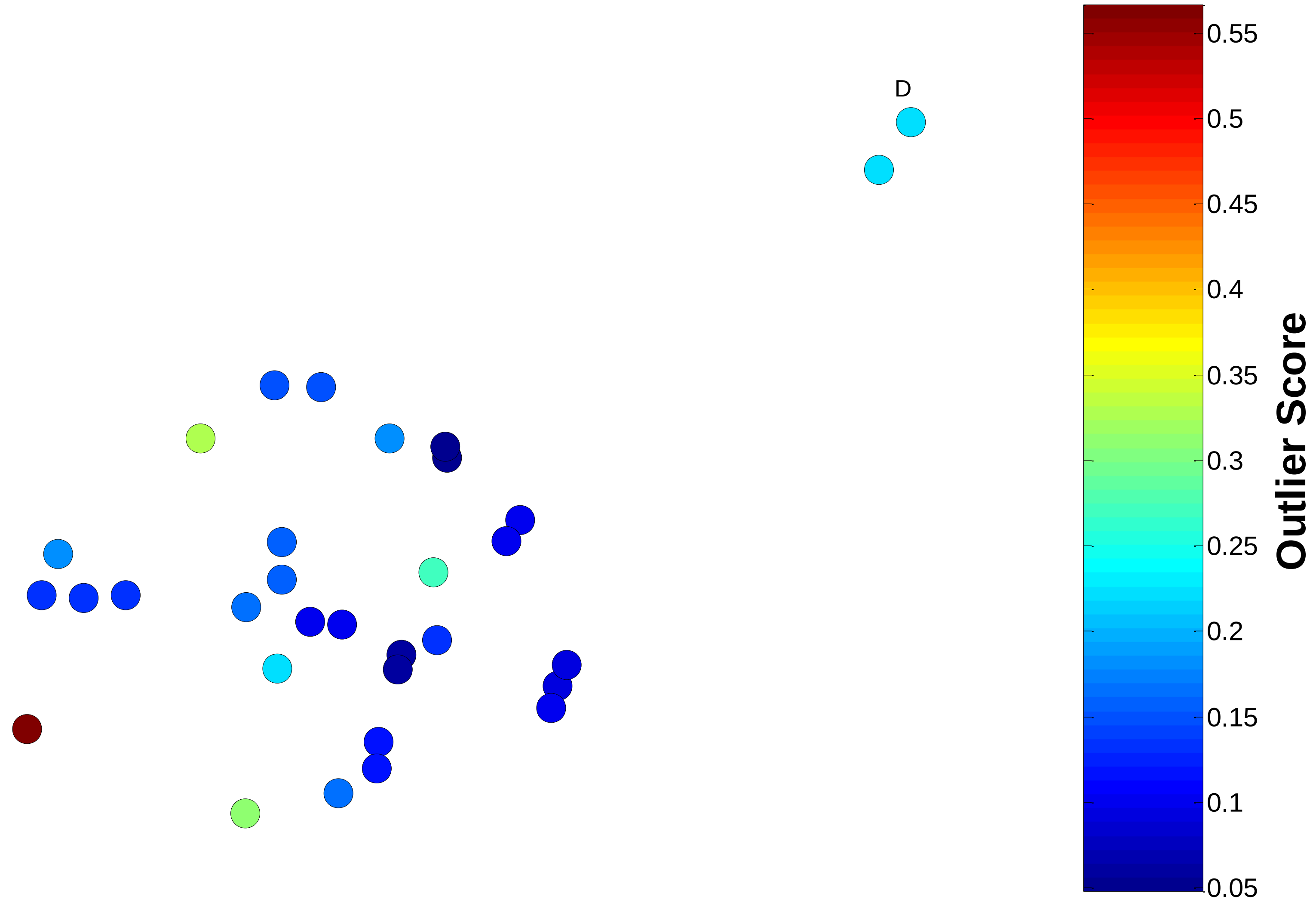
1. Calculate the anomaly score, f(x), for each data point in the dataset.
2. Use a threshold t on this score to determine outliers.

x is an outlier iff f(x) > t

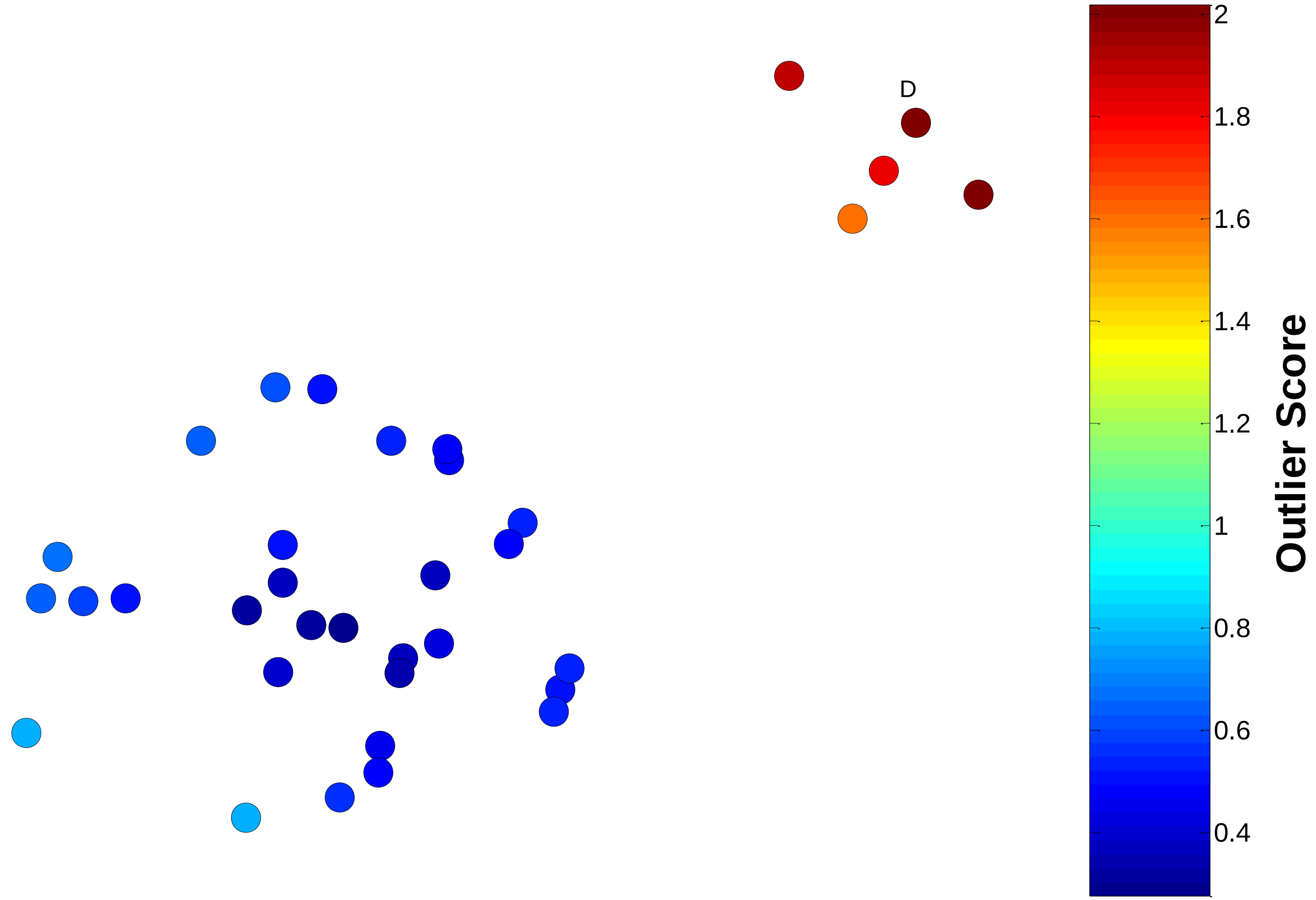
1 Nearest Neighbor - One Outlier



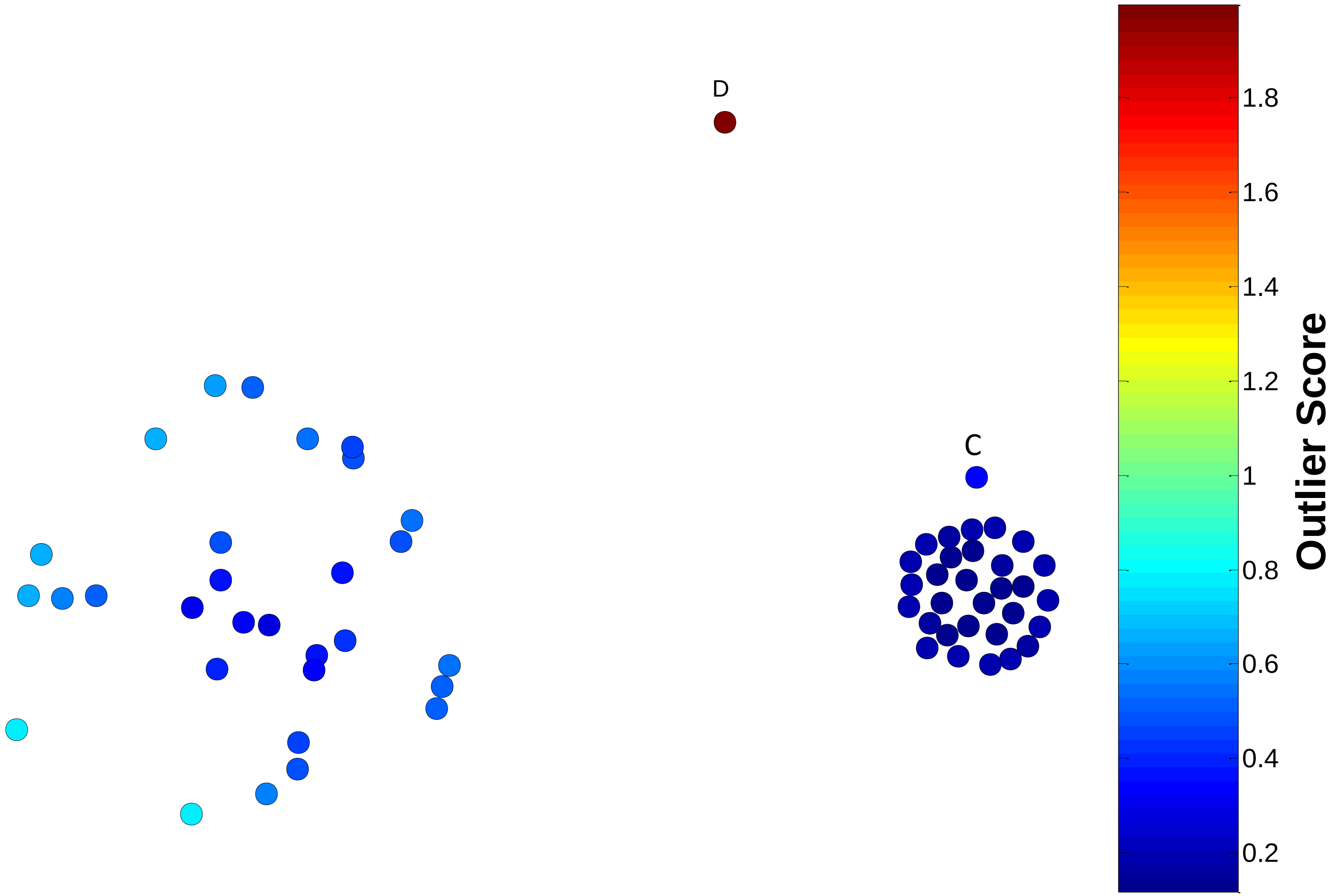
1 Nearest Neighbor - Two Outliers



5 Nearest Neighbors - Small Cluster



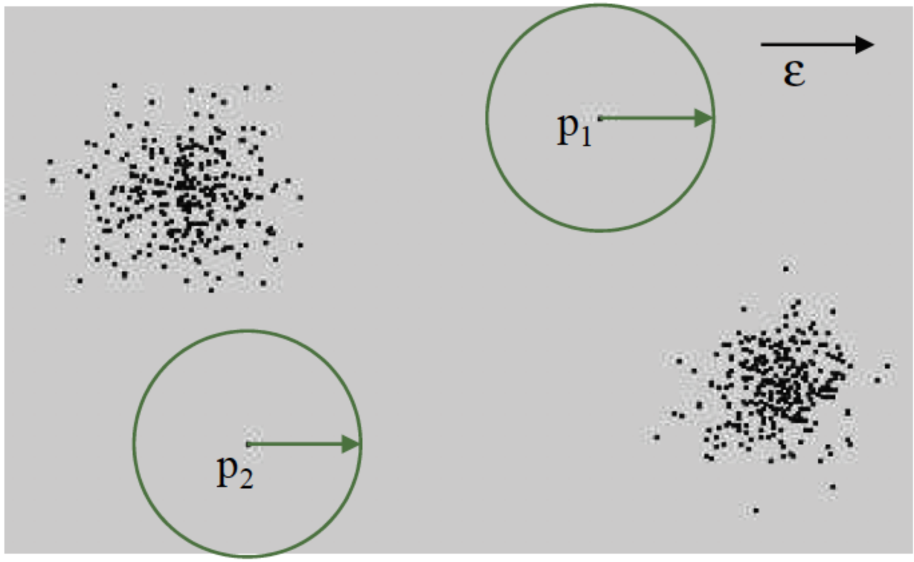
5 Nearest Neighbors - Differing Density



Distance-based Approaches

DB(ε,π)-Outliers

- Basic model [Knorr and Ng 1997]
- Given a radius *ε* and a percentage *π*
- A point *p* is considered an outlier if **at most *π* percent of all other points have a distance to *p* less than ε**, *i.e., it is close to few points*



$$OutlierSet(\varepsilon,\pi) = \{p \mid \frac{Card(\boxed{\{q \in DB \mid dist(p,q) < \varepsilon\}})}{Card(DB)} \leq \pi\}$$

range-query with radius ε

General approach for computation

- **Efficient computation:** Nested loop algorithm
  - For any object p, calculate its distance from other objects
    - count the # of other objects in the ε-neighborhood.
    - If $\pi \cdot n$ other objects are within ε distance, terminate the inner loop
  - Otherwise, p is a DB(ε, π) outlier
- **Efficiency**:
  - Actually, CPU time is not $O(n^2)$ but linear to the data set size since for most non-outlier objects, the inner loop terminates early

Strengths/Weaknesses of Distance-Based Approaches

Pros

- Simple

Cons

- Expensive – $O(n^2)$
- Sensitive to parameters
- Sensitive to variations in density
- Distance becomes less meaningful in high-dimensional space

Density-based Approaches

Density-based Approaches

- General idea
  - **Compare the density around a point with the density around its local neighbors**
  - The relative density of a point compared to its neighbors is computed as an outlier score
  - Approaches differ in how to estimate density

- Basic assumption
  - The **density around a normal data** object is similar to the density around its neighbors
  - The **density around an outlier** is considerably different to the density around its neighbors

Density-based Approaches

- **Density-based Outlier:** The outlier score of an object is the inverse of the density around the object.
  - Can be defined in terms of the *k* nearest neighbors
    - One definition: Inverse of distance to *k*th neighbor
    - Another definition: Inverse of the average distance to *k* neighbors
  - DBSCAN definition

- If there are regions of different density, this approach can have problems

Local Outlier Factor (LOF) [Breunig et al. 1999], [Breunig et al. 2000]

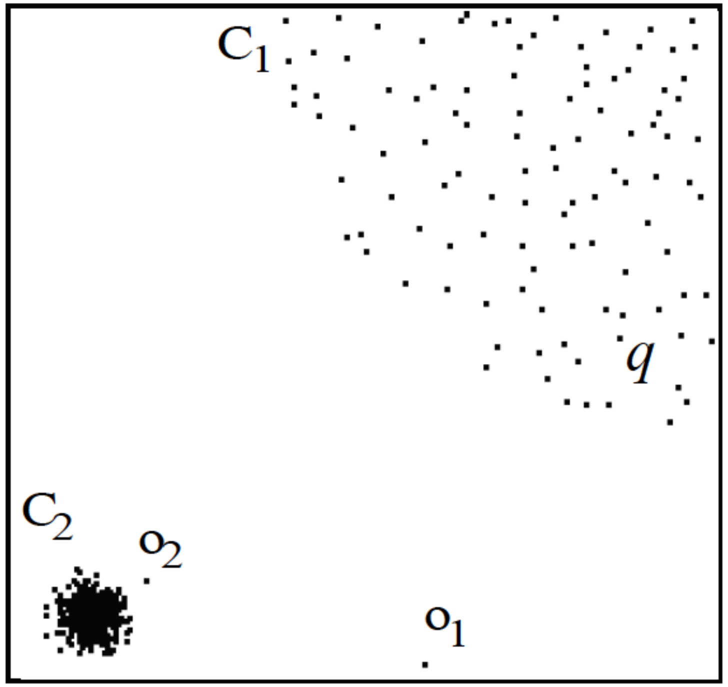
Motivation:

- Distance-based outlier detection models have problems with different densities
- How to compare the neighborhood of points from areas of different densities?

Example

- DB(ε,π)-outlier model
  - Parameters ε and π cannot be chosen so that o_2 is an outlier but none of the points in cluster C_1 (e.g. q) is an outlier
- Outliers based on kNN-distance
  - kNN-distances of objects in C_1 (e.g. q) are larger than the kNN-distance of o_2

Solution: consider relative density



Relative Density

- Consider the density of a point relative to that of its k nearest neighbors

$$density(\mathbf{x}, k) = \left(\frac{\sum_{\mathbf{y} \in N(\mathbf{x},k)} distance(\mathbf{x}, \mathbf{y})}{|N(\mathbf{x}, k)|} \right)^{-1}$$

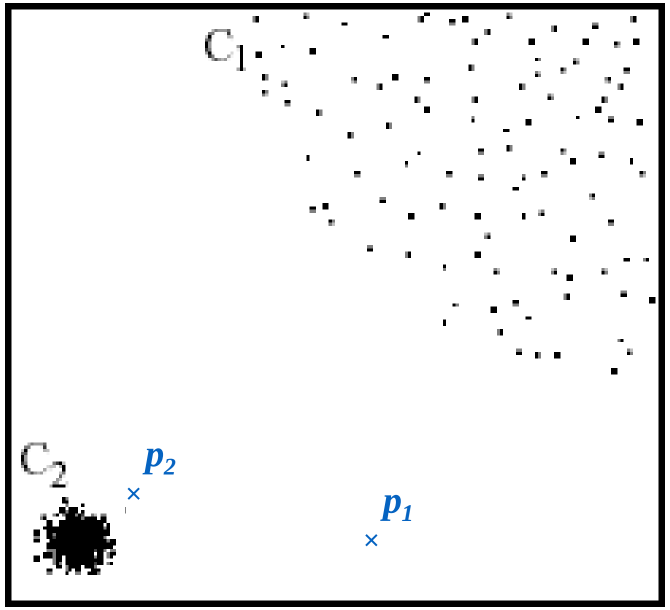
$$average\ relative\ density(\mathbf{x}, k) = \frac{density(\mathbf{x}, k)}{\sum_{\mathbf{y} \in N(\mathbf{x},k)} density(\mathbf{y}, k)/|N(\mathbf{x}, k)|}. \quad (10.7)$$

Algorithm 10.2 Relative density outlier score algorithm.

{k is the number of nearest neighbors}
for all objects $\mathbf{x}$ **do**
    Determine $N(\mathbf{x}, k)$, the k-nearest neighbors of $\mathbf{x}$.
    Determine $density(\mathbf{x}, k)$, the density of $\mathbf{x}$, using its nearest neighbors, i.e., the objects in $N(\mathbf{x}, k)$.
end for
for all objects $\mathbf{x}$ **do**
    Set the $outlier\ score(\mathbf{x}, k) = average\ relative\ density(\mathbf{x}, k)$ from Equation 10.7.
end for

Local Outlier Factor (LOF)

- For each point, compute the density of its local neighborhood
- Compute local outlier factor (LOF) of a sample *p* as the average of the ratios of the density of sample *p* and the density of its nearest neighbors
- Outliers are points with largest LOF value

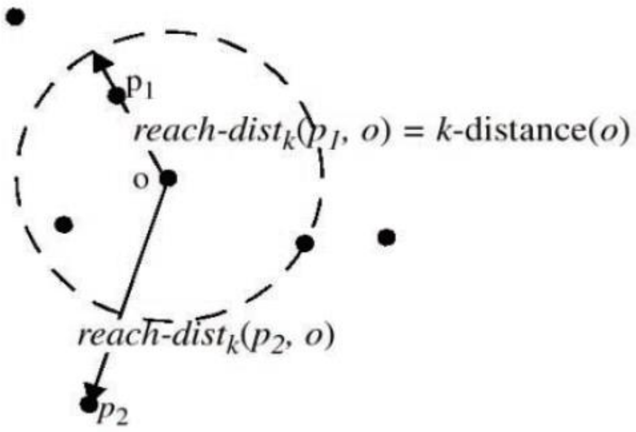


In the NN approach, p_2 is not considered as outlier, while LOF approach find both p_1 and p_2 as outliers

Local Outlier Factor (LOF)

- Reachability distance
  - Introduces a smoothing factor

$$reach-dist_k(p,o) = \max\{k-\text{distance}(o), dist(p,o)\}$$



- Local reachability distance (*lrd*) of point *p*
  - Inverse of the average reach-dists of the kNNs of *p*

$$lrd_k(p) = 1 / \left(\frac{\sum_{o \in kNN(p)} reach-dist_k(p,o)}{Card(kNN(p))} \right)$$

- Local outlier factor (LOF) of point *p*
  - Average ratio of *lrds* of neighbors of *p* and *lrd* of *p*

$$LOF_k(p) = \frac{\sum_{o \in kNN(p)} \frac{lrd_k(o)}{lrd_k(p)}}{Card(kNN(p))}$$

Strengths/Weaknesses of Density-Based Approaches

Pros

- Simple

Cons

- Expensive – $O(n^2)$
- Sensitive to parameters
- Density becomes less meaningful in high-dimensional space

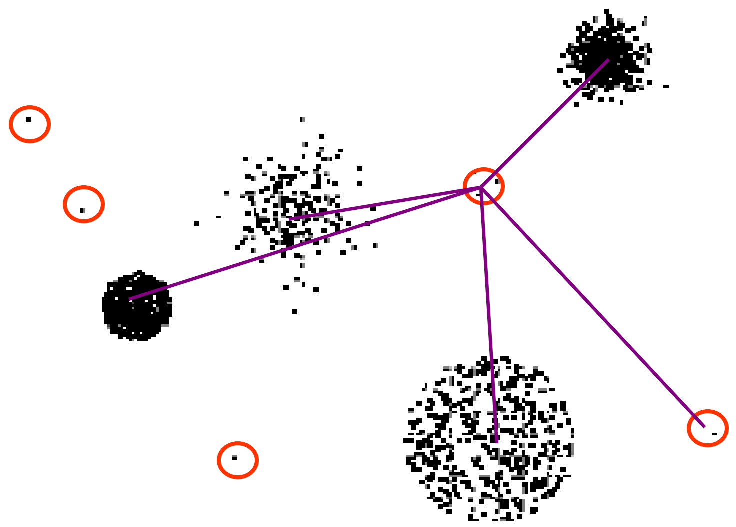
Clustering-based Approaches

Clustering and Anomaly Detection

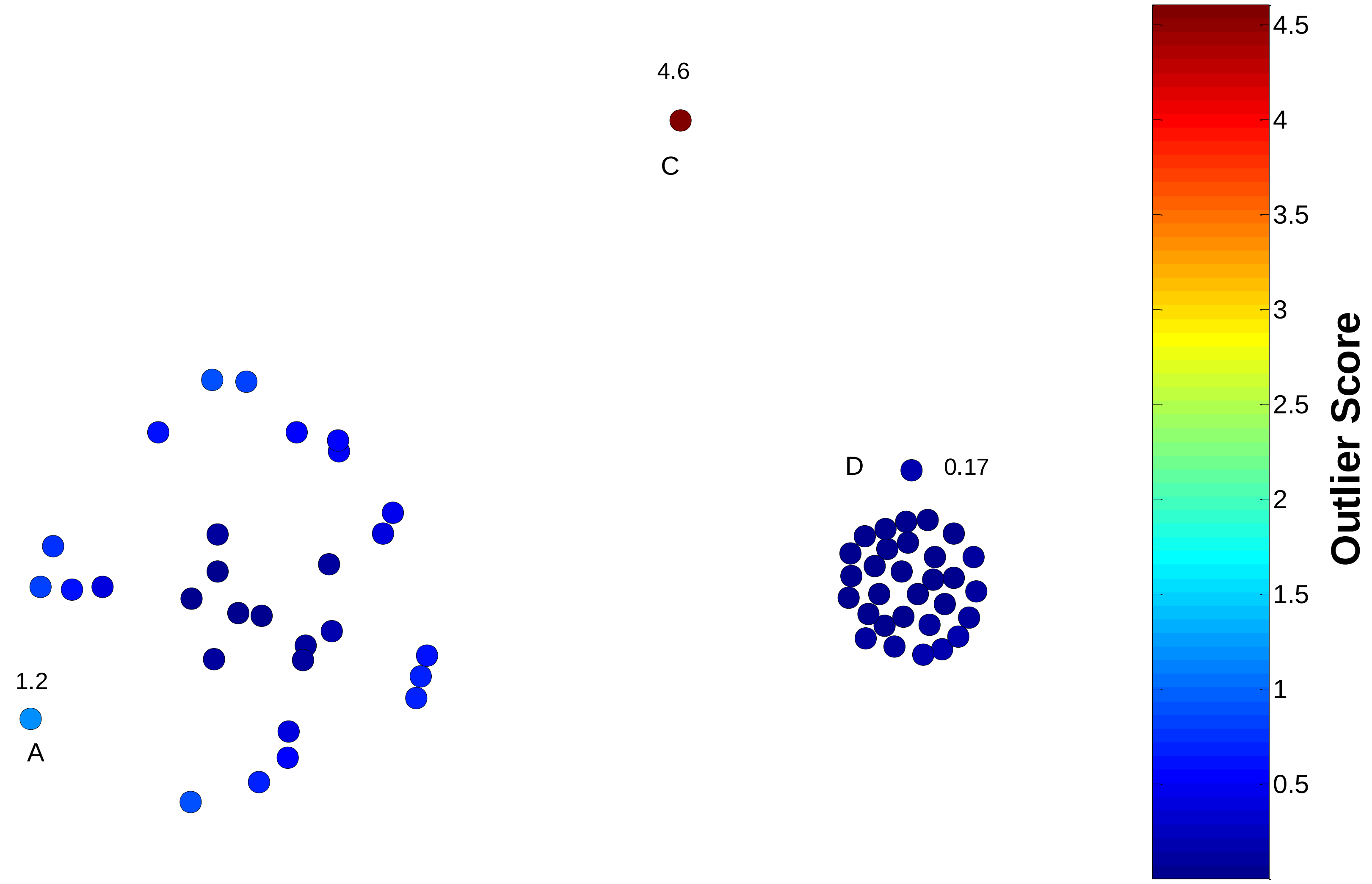
- Are outliers just a side product of some clustering algorithms?
  - Many clustering algorithms do not assign all points to clusters but account for noise objects (e.g. DBSCAN, OPTICS)
  - Look for outliers by applying one algorithm and retrieve the noise set
- Problem:
  - Clustering algorithms are optimized to find clusters rather than outliers
  - Accuracy of outlier detection depends on how good the clustering algorithm captures the structure of clusters
  - A set of many abnormal data objects that are similar to each other would be recognized as a cluster rather than as noise/outliers

Clustering-Based Approaches

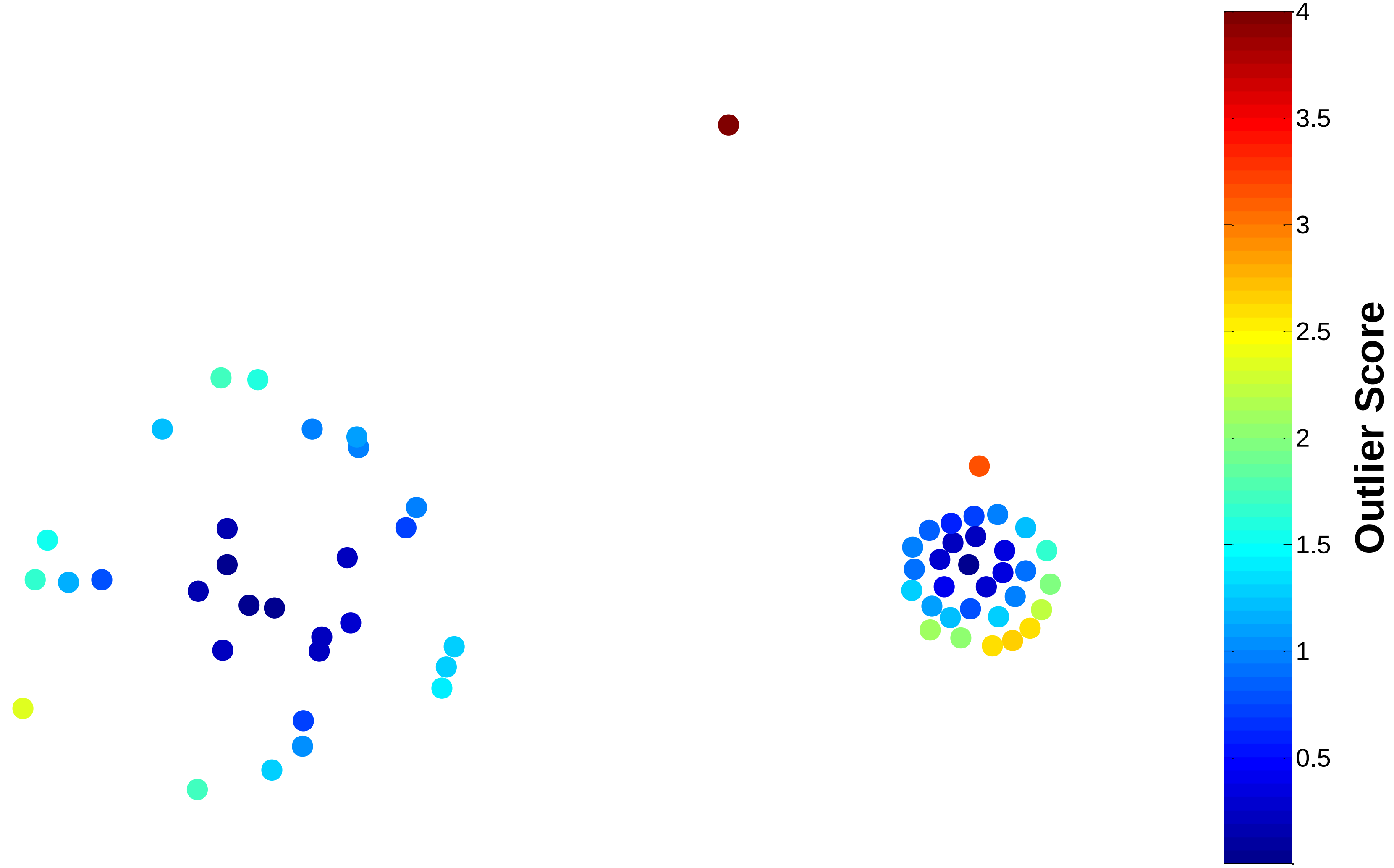
- **Clustering-based Outlier:** An object is a cluster-based outlier if it does not strongly belong to any cluster
  - For prototype-based clusters, an object is an outlier if it is not close enough to a cluster center
  - For density-based clusters, an object is an outlier if its density is too low
  - For graph-based clusters, an object is an outlier if it is not well connected
- Other issues include the impact of outliers on the clusters and the number of clusters



Distance of Points from Closest Centroids



Relative Distance of Points from Closest Centroid



Strengths/Weaknesses of Clustering-Based Approaches

Pros

- Simple
- Many clustering techniques can be used

Cons

- Can be difficult to decide on a clustering technique
- Can be difficult to decide on number of clusters
- Outliers can distort the clusters

Isolation forests

- Proposed in 2008
- Perfect for high dimensional data
- Linear in time complexity
- Small memory requirement
- It is composed of binary search trees
- The main idea is that anomalies are few and different from the main distribution, hence they can be isolated using few partitions of the tree, i.e. a short path

Isolation forests

1. Use the training dataset to build a number of iTree:
   1. For each iTree, randomly select one feature, and a random split on that feature, in the range (min, max) of that feature
2. For each data point in the test set:
   1. Pass it in all the iTrees and count the path length for each tree
   2. Assign an anomaly score to the instance
   3. Label the point as an anomaly if its score is greater than a user defined threshold

Summary

- Different models are based on different assumptions
- Different models provide different types of output (labeling/scoring)
- Different models consider outlier at different resolutions (global/local)
- Thus, different models will produce different results
- A thorough and comprehensive comparison between different models and approaches is still missing

Time Series anomalies

In this case we can find 3 categories of algorithms:

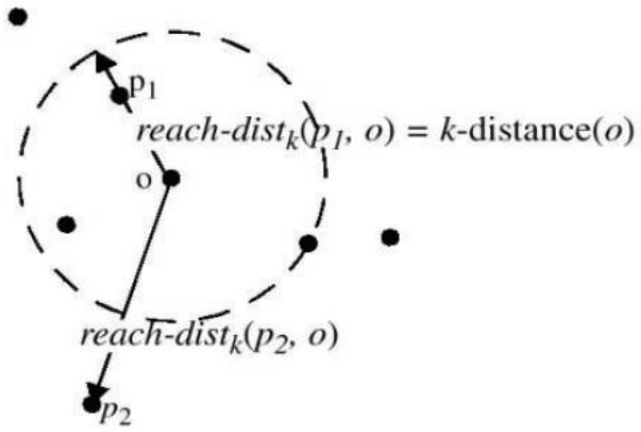
1. Supervised
2. Unsupervised
3. Semi-supervised, in which only the normal behavior is learned (and the anomalies are considered as complementary)

Time Series anomalies

D	L	Method	Area	Family	Origin	Language
univariate	unsupervised	NoveltySVR [86]	Classic ML	● distance	own	Python
		PS-SVM [85]	Classic ML	● distance	own	Python
		Ensemble GI [43]	Data Mining	▼ encoding	own	Python
		GrammarViz [120]	Data Mining	▼ encoding	original	Java
		HOT SAX [70]	Data Mining	● distance	original	Python
		TSBitmap [144]	Data Mining	▼ encoding	community	Python
		NormA-SJ [15]	Data Mining	● distance	original	Python
		SAND [17]	Data Mining	● distance	original	Python
		Series2Graph [16]	Data Mining	▼ encoding	original	Python
		STAMP [156]	Data Mining	● distance	original	R
		STOMP [164]	Data Mining	● distance	original	R
		VALMOD [82]	Data Mining	● distance	original	R
		Left STAMPi [156]	Data Mining	● distance	original	Python
		SSA [155]	Data Mining	● distance	own	Python
		PST [128]	Data Mining	✚ trees	own	R
		NumentaHTM [3]	Deep L.	★ forecasting	original	Python
		Sub-LOF [22]	Outlier Det.	● distance	own	Python
		Sub-IF [83]	Outlier Det.	✚ trees	own	Python
		DWT-MLEAD [134]	Signal A.	▲ distribution	own	Python
		FFT [111]	Signal A.	■ reconstruction	own	Python
		SR [112]	Signal A.	■ reconstruction	original	Python
		S-H-ESD [62]	Statistics	▲ distribution	own	R
		DSPOT [122]	Statistics	▲ distribution	original	Python
		ARIMA [65]	Statistics	★ forecasting	own	Python
		MedianMethod [10]	Statistics	★ forecasting	own	Python
		SARIMA [52]	Statistics	★ forecasting	own	Python
		Triple ES [1]	Statistics	★ forecasting	own	Python
		PCI [157]	Statistics	■ reconstruction	own	Python
	semi-supervised	RForest [21]	Classic ML	★ forecasting	own	Python
		XGBoosting [34]	Classic ML	★ forecasting	own	Python
		TARZAN [71]	Data Mining	▼ encoding	original	Python
		HealthESN [32]	Deep L.	★ forecasting	own	Python
		OceanWNN [143]	Deep L.	★ forecasting	own	Pytorch
		Bagel [79]	Deep L.	■ reconstruction	original	Python
		Donut [150]	Deep L.	■ reconstruction	original	Pytorch
		IE-CAE [44]	Deep L.	■ reconstruction	own	Pytorch
		SR-CNN [112]	Deep L.	■ reconstruction	original	Pytorch
		Sub-Fast-MCD [115]	Statistics	▲ distribution	own	Python
multivariate	unsupervised	PCC [121]	Classic ML	■ reconstruction	community	Python
		HBOS [47]	Classic ML	● distance	community	Python
		k-Means [151]	Classic ML	● distance	own	Python
		KNN [110]	Classic ML	● distance	community	Python
		EIF [58]	Classic ML	✚ trees	original	Python
		Torsk [60]	Deep L.	★ forecasting	original	Pytorch
		CBLOF [59]	Outlier Det.	● distance	community	Python
		COF [130]	Outlier Det.	● distance	community	Python
		DBStream [55]	Outlier Det.	● distance	original	R
		LOF [22]	Outlier Det.	● distance	community	Python
		COPOD [80]	Outlier Det.	▲ distribution	community	Python
		IF-LOF [36]	Outlier Det.	✚ trees	own	Python
		iForest [83]	Outlier Det.	✚ trees	community	Python
	semi-supervised	RobustPCA [101]	Classic ML	■ reconstruction	community	Python
		RBForest [165]	Classic ML	★ forecasting	own	Python
		Hybrid KNN [124]	Deep L.	● distance	own	Pytorch
		DeepAnT [94]	Deep L.	★ forecasting	own	Pytorch
		DeepNAP [72]	Deep L.	★ forecasting	own	Pytorch
		LSTM-AD [89]	Deep L.	★ forecasting	own	Pytorch
		MTAD-GAT [161]	Deep L.	★ forecasting	own	Pytorch
		Telemanom [64]	Deep L.	★ forecasting	original	Tensorflow
		MSCRED [159]	Deep L.	■ reconstruction	own	Tensorflow
		AE [117]	Deep L.	■ reconstruction	own	Tensorflow
		DAE [117]	Deep L.	■ reconstruction	own	Tensorflow
		EncDec-AD [88]	Deep L.	■ reconstruction	own	Pytorch
		LSTM-VAE [106]	Deep L.	■ reconstruction	own	Tensorflow
		OmniAnomaly [125]	Deep L.	■ reconstruction	original	Tensorflow
		TAnoGan [8]	Deep L.	■ reconstruction	own	Pytorch
		Fast-MCD [115]	Statistics	▲ distribution	own	Python
		LaserDBN [100]	Stochastic L.	▼ encoding	own	Python
	superv.	NF [116]	Deep L.	▲ distribution	own	Pytorch
		HIF [91]	Outlier Det.	✚ trees	original	Python
		MultiHMM [78]	Stochastic L.	▼ encoding	own	Python

Subsequence LOF (SubLOF)

The algorithm is the same used for the tabular data, but on time series: to handle them, a sliding window is applied to split the time series, then the LOF is applied to the smaller splits.



All the other methods

Similarly to LOF, we can apply all the other methods seen so far to the time series, e.g. clustering techniques, distance based ones, density based ones.

Problems?

When dealing with time series, the major problem is the efficiency. A lot of computations are needed, hence the algorithms seen so far, even if they are applicable, can be extremely expensive to apply.

Discords

Discords can be defined as the most unsual subsequence in a time series.

Finding discords mean finding anomalies.

1. We can use the same concept as before: k-nn. In this case, a discord is a subsequence that has the largest k-distance among all the subsequence of the time series.

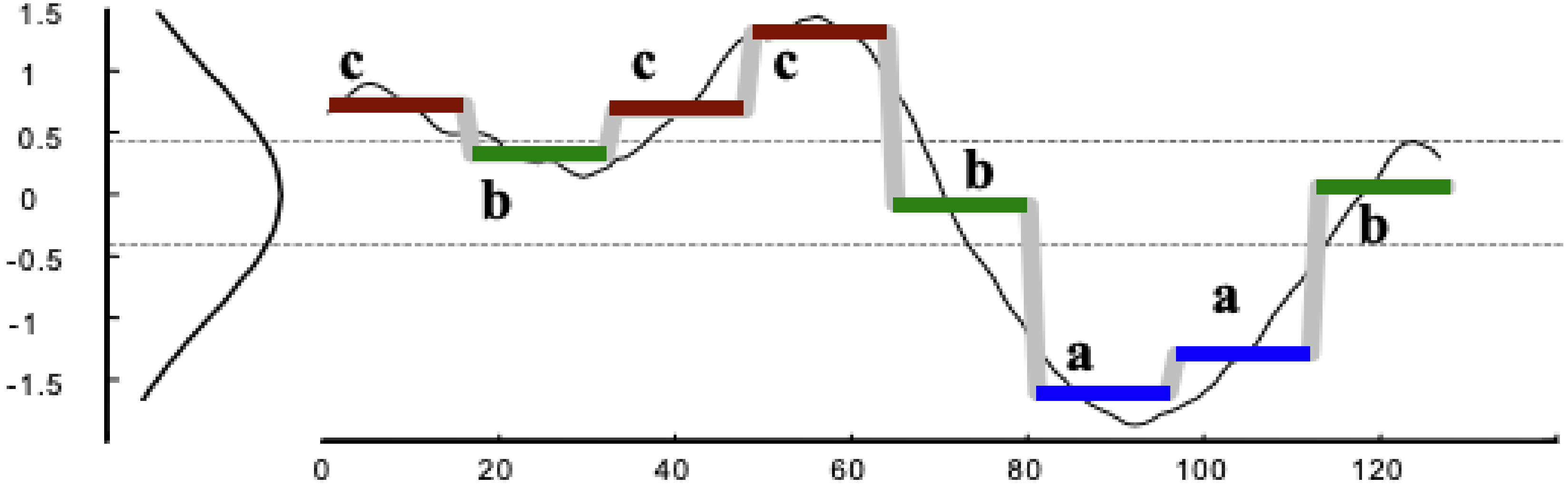
Discords

Data: Symbolic representation of time series data (T), sliding window size (w)
Result: $worst\_discord\_distance$, $worst\_discord\_location$

$worst\_discord\_distance = 0$;
$worst\_discord\_location = NaN$;
for $i = 1... \mid T \mid -w + 1$ **do**
    $nearest\_neighbor\_distance = infinity$;
    for $j = 1... \mid T \mid -w + 1$ **do**
        if $\mid i - j \mid \geq w$ **then**
            if $nearest\_neighbor\_distance > Distance(t_i...t_{i+w-1}, t_j...t_{j+w-1})$ **then**
                $nearest\_neighbor\_distance = Distance(t_i...t_{i+w-1}, t_j...t_{j+w-1})$;
            end
        end
    end
    if $nearest\_neighbor\_distance > worst\_discord\_distance$ **then**
        $worst\_discord\_distance = nearest\_neighbor\_distance$;
        $worst\_discord\_location = i$;
    end
end

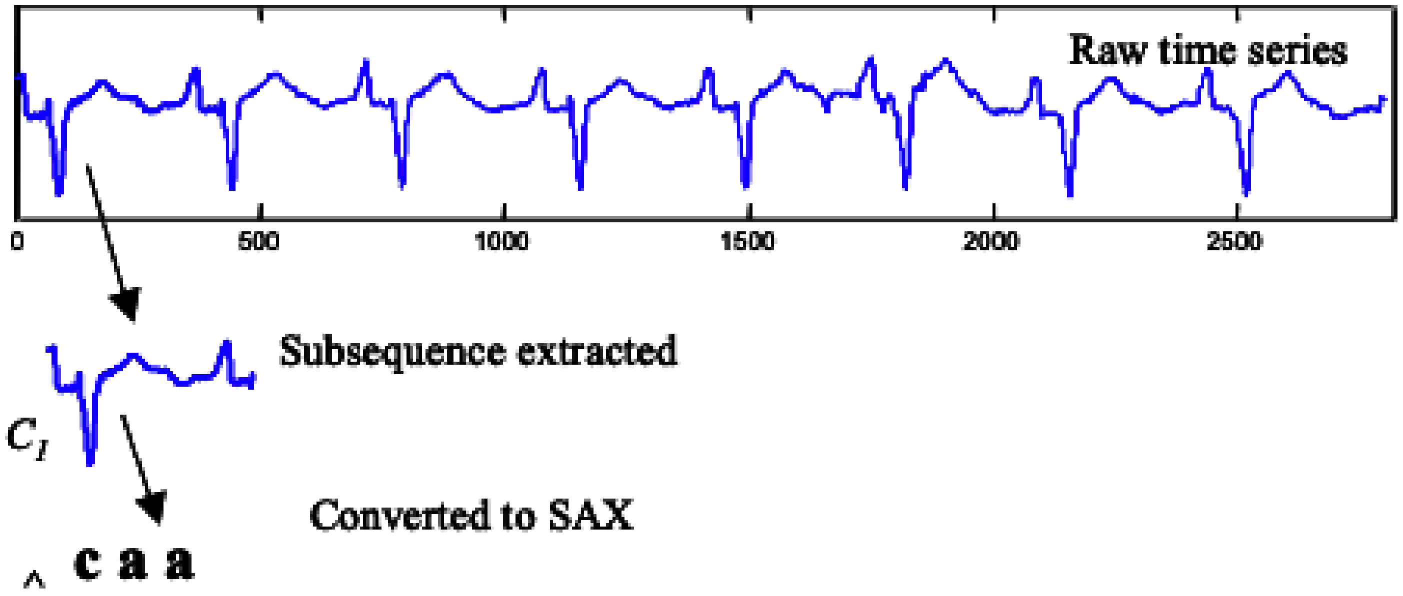
Discords - problems

The application of the algorithm requires O(n^2), that for big time series is intractable! We need to find a faster algorithm, epxloiting some heuristics. To start, we can exploit SAX representation:



Discords – HOT SAX

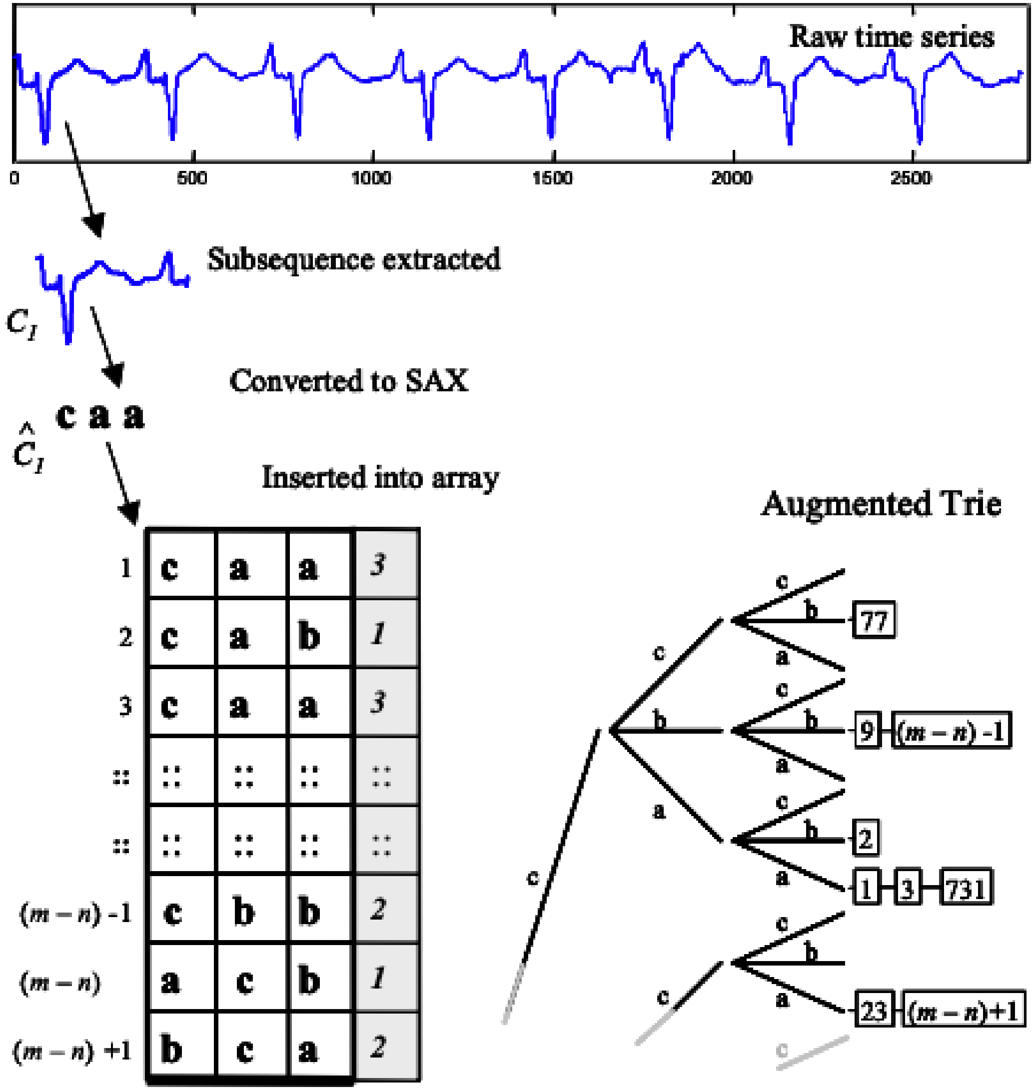
The algorithm begins by constructing the SAX representation of the time series, by exploiting a sliding window of length n. To do so, we need to fix the length of the window and also the alphabet size for SAX, which is the parameter a.



Discords – HOT SAX

While creating the SAX representation, we can also create a better data structure, to simplify the brute force computation:

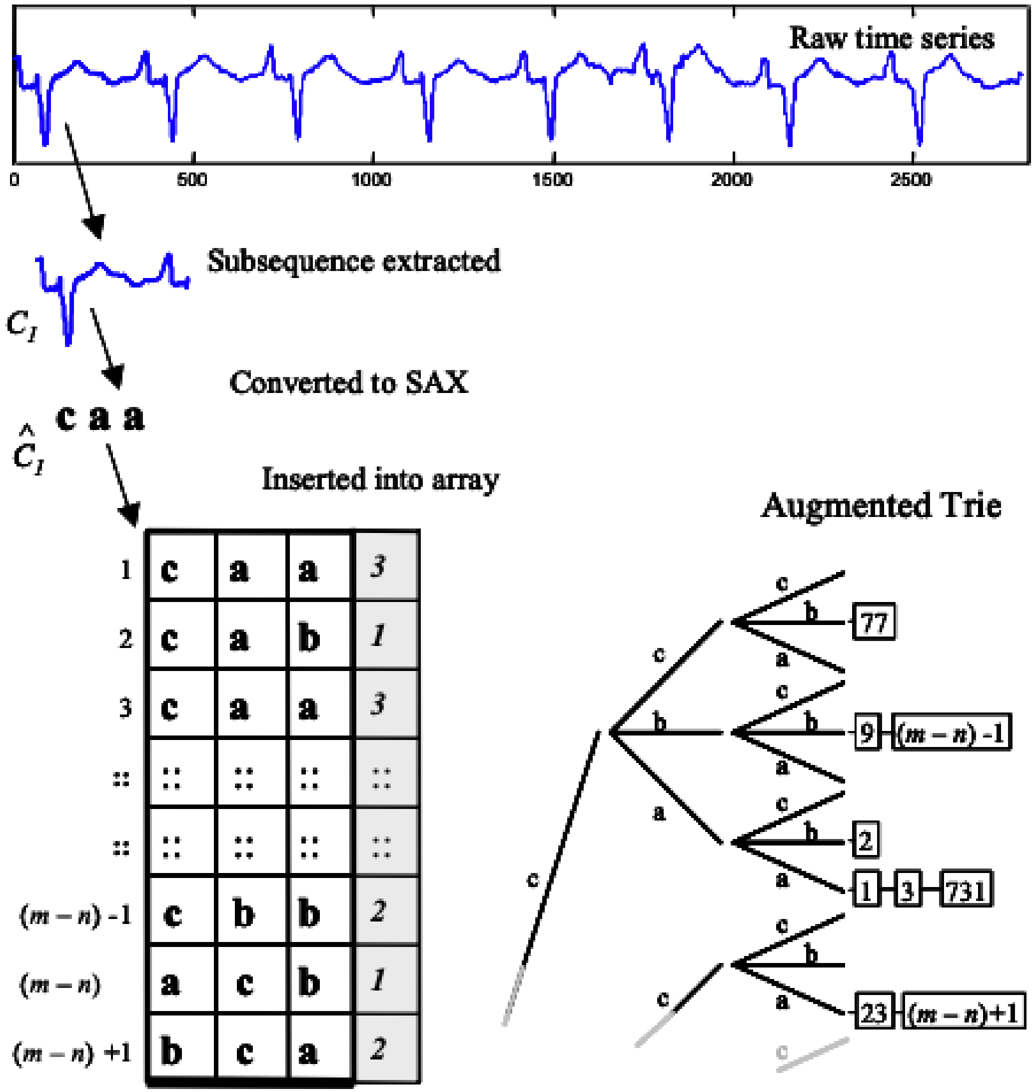
1. An array, in which we store the SAX sequence and the number of times we saw it
2. A trie, in which the leaf nodes contain the locations of that Sax representation



1	c	a	a	3
2	c	a	b	1
3	c	a	a	3
::	::	::	::	::
::	::	::	::	::
(m – n) -1	c	b	b	2
(m – n)	a	c	b	1
(m – n) +1	b	c	a	2

Why?

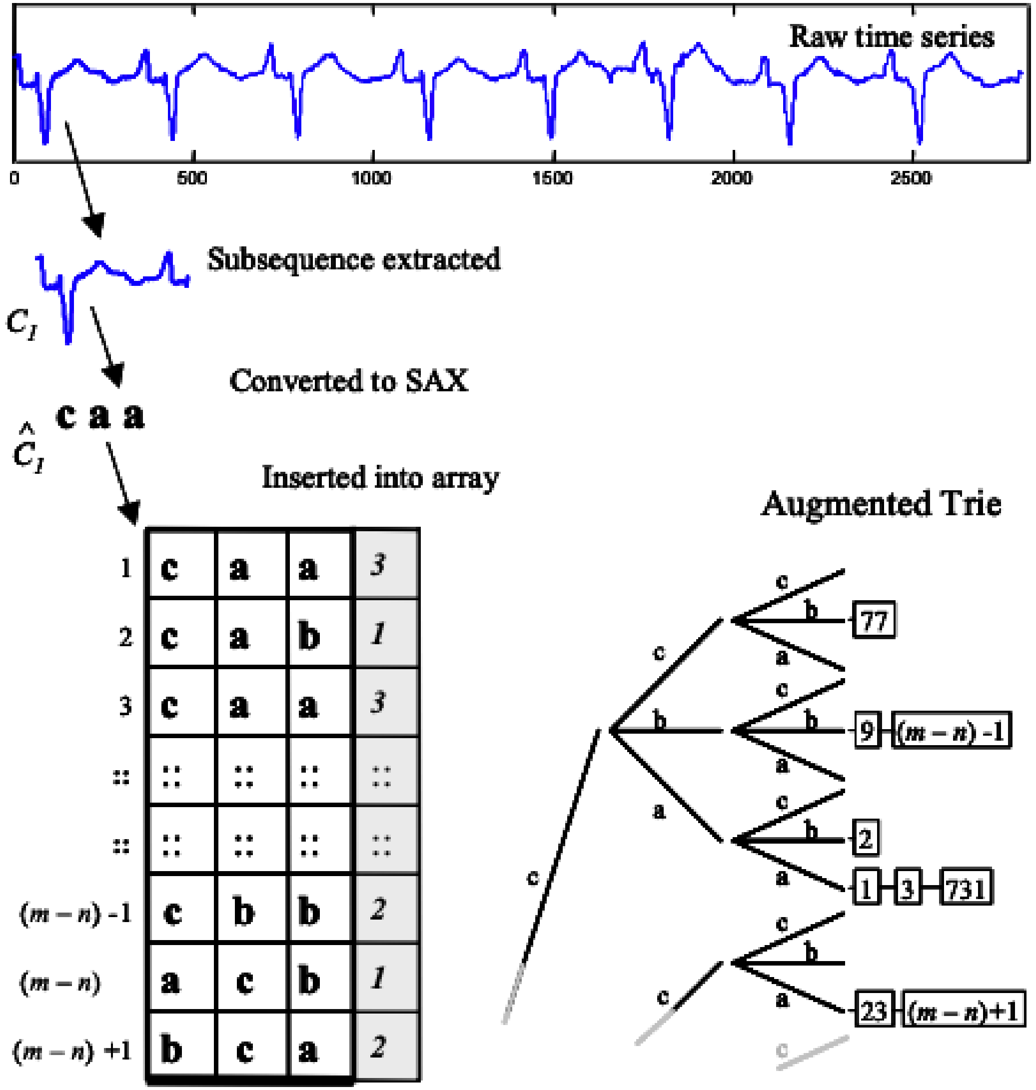
In the outer loop of the brute force approach, we were looking at all the SAX representation. Since this is time consuming, we want to look only at the ones more promising (i.e. the most probable to be an anomaly). For doing so, we can exploit the heuristic that an anomaly is something that doesn't happen often and hence its SAX representation would be with a lower count.



Outer loop

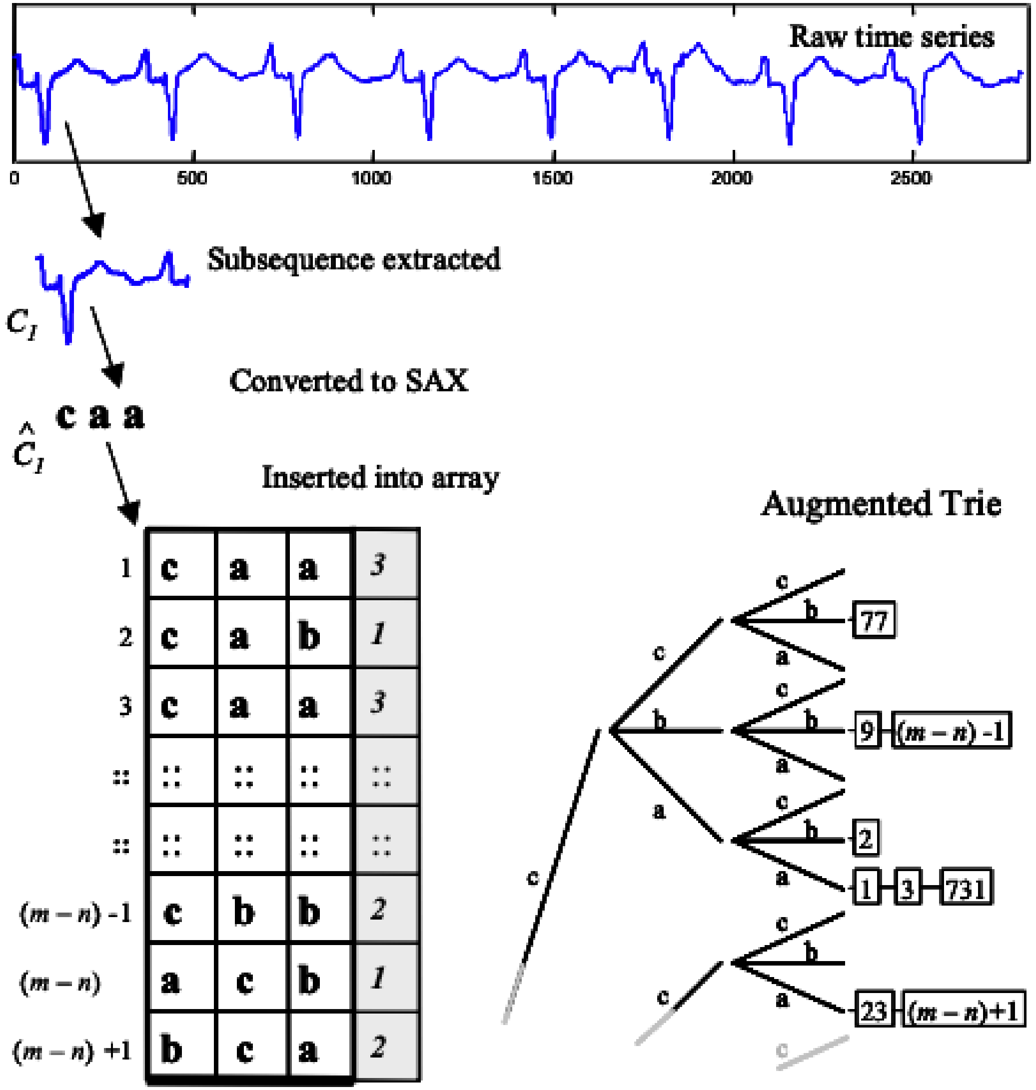
1. Look for the lowest counts in the array
2. Give them to the outer loop as starters
3. For the other SAX representation, search randomly a fixed number of times

The data structures can be created in time and space linear in the length of T.



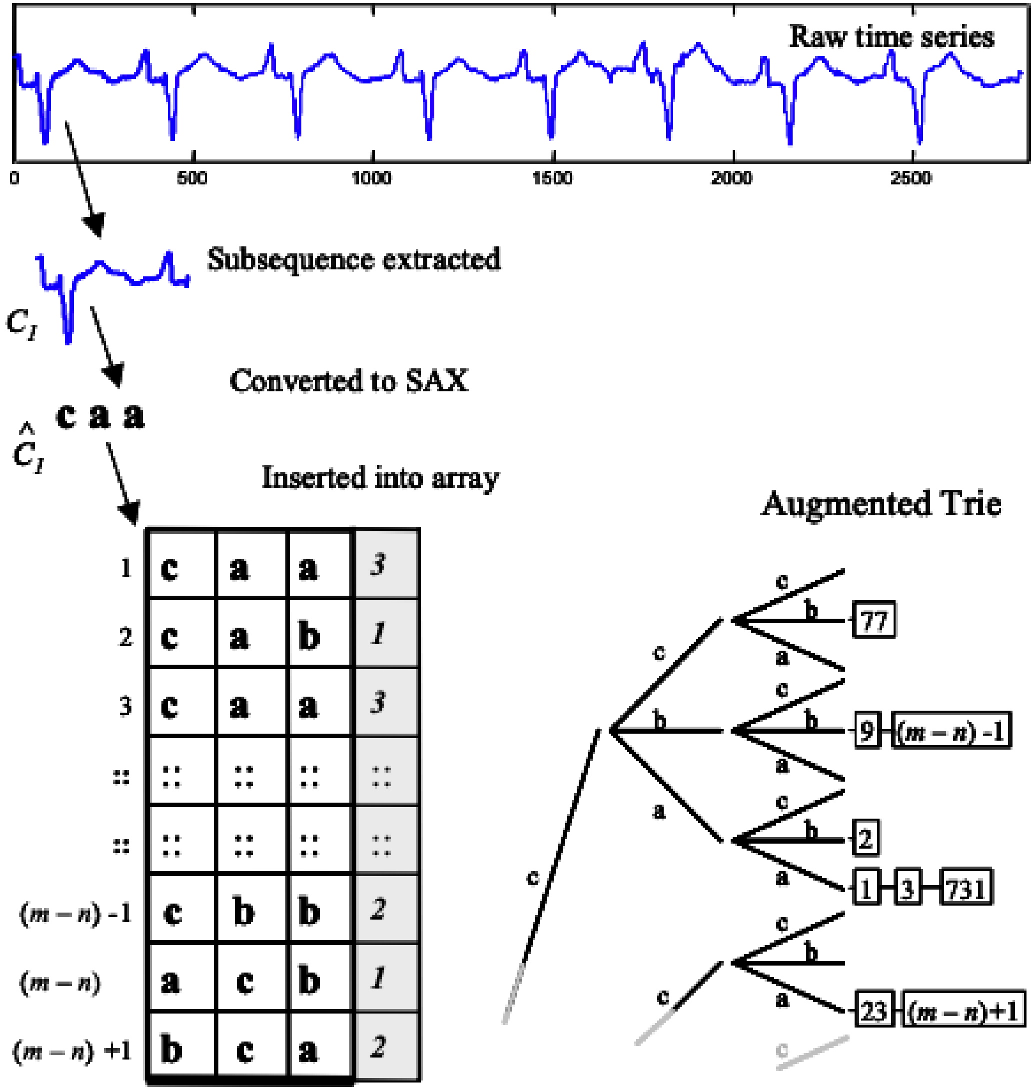
Inner loop

The idea: subsequence with the same encoding are similar. In addition, we just need to find one subsequence that is similar enough, i.e. the distance to the candidate needs to be smaller than the distance found so far).



Inner loop

1. We trave the tire and discover the subsequence with the same encoding
2. We first evaluate the distance from them, if we are lucky we exit
3. If not, random search on the other encodings.



References

- Anomaly Detection. Chapter 10. Introduction to Data Mining.
- Survey for time series anomaly detection: https://dl.acm.org/doi/pdf/10.14778/3538598.3538602
- HOT SAX: https://www.cs.ucr.edu/~eamonn/HOT%20SAX%20%20long-ver.pdf