A Discussion on Outlier Detection

Rei Sanchez-Arias, Ph.D.

Introduction to anomaly/outlier detection

Oct 21, 2021 (Thu)

- abnormal detection (clustering)
- R practice.

Oct 26, 2021 (TUE)

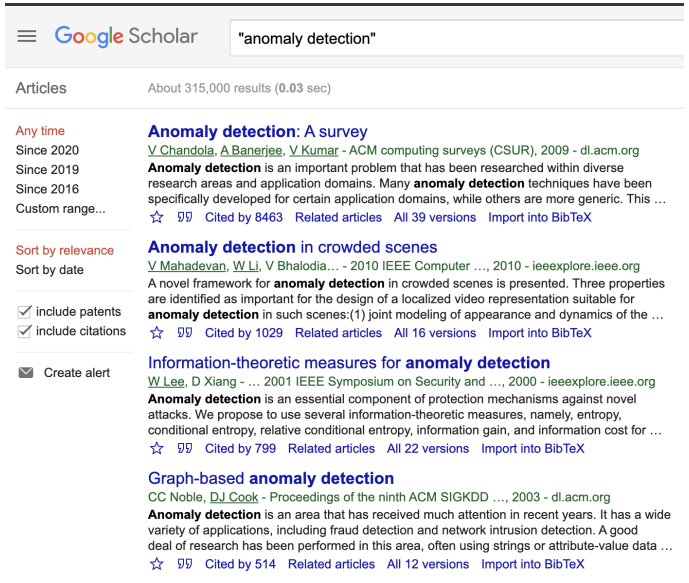
- abnormal detection (supervised Learning)
- abnormal detection example (research)
- project plan updated

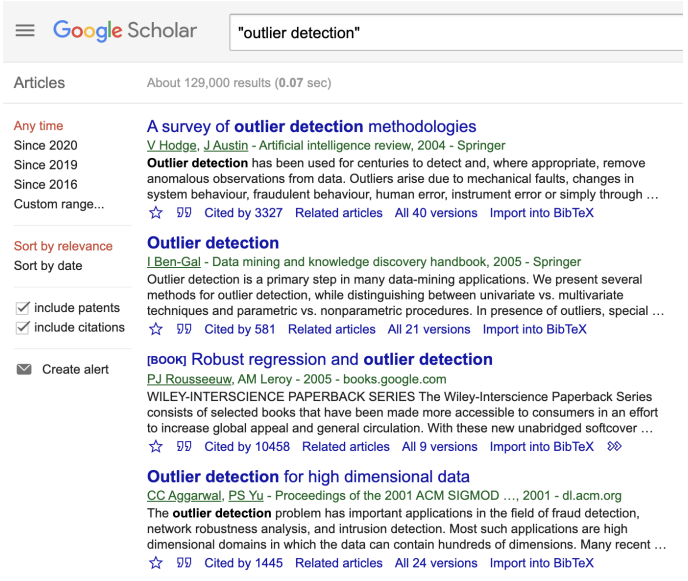
Oct 28, 2021 (Thu)

- Research Examples for project
- Review for Quiz
- Quiz in class [Canvas]

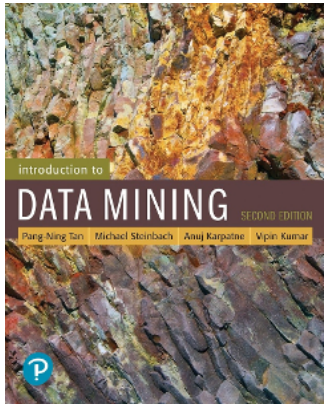
Some resources and motivation

Some resources





"Introduction to Data Mining" by Tan, Steinbach, Karpatne, Kumar

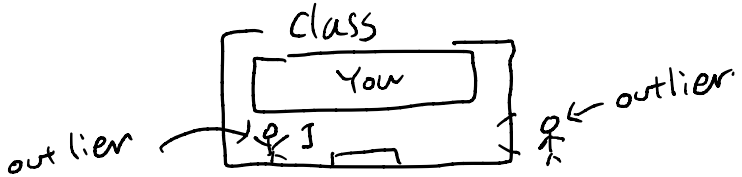


> Examples and materials in this set of slides are adapted from the books mentioned above.

Discussion

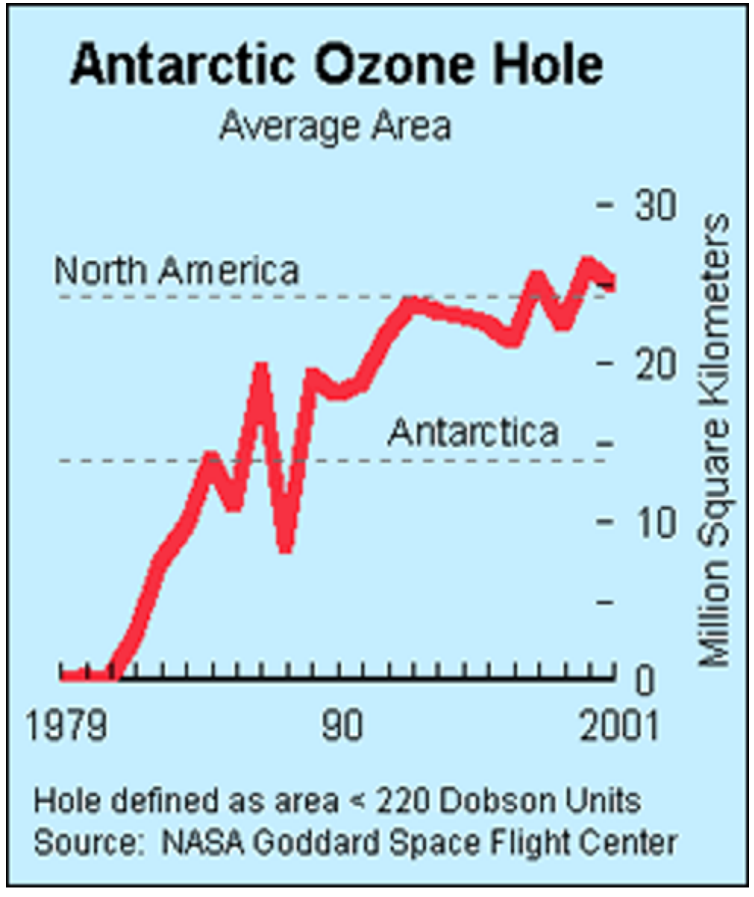
What are anomalies/outliers? The set of data points that are **considerably different** than the remainder of the data

An outlier is an observation that deviates from the (model fit suggested by the) majority of the observations



- 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 year old
  - Unusually high blood pressure

Importance of anomaly detection



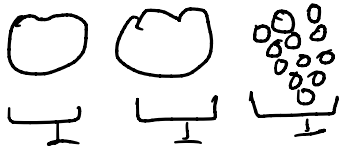
https://www.epa.gov/ozone-layer-protection

- 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**!

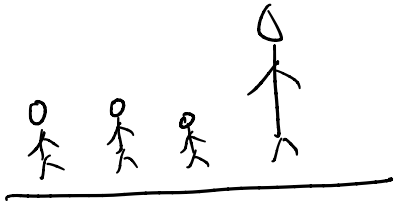
Some causes and issues

Causes of anomalies

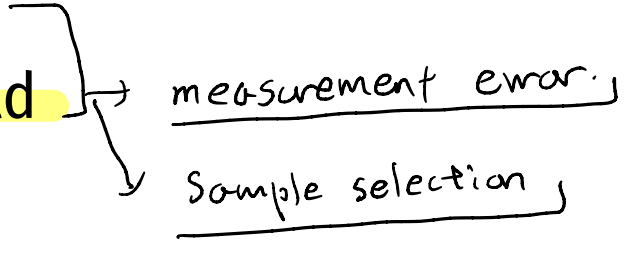
- **Data from different classes**
  - Measuring the weights of oranges, but a few grapefruit are mixed in

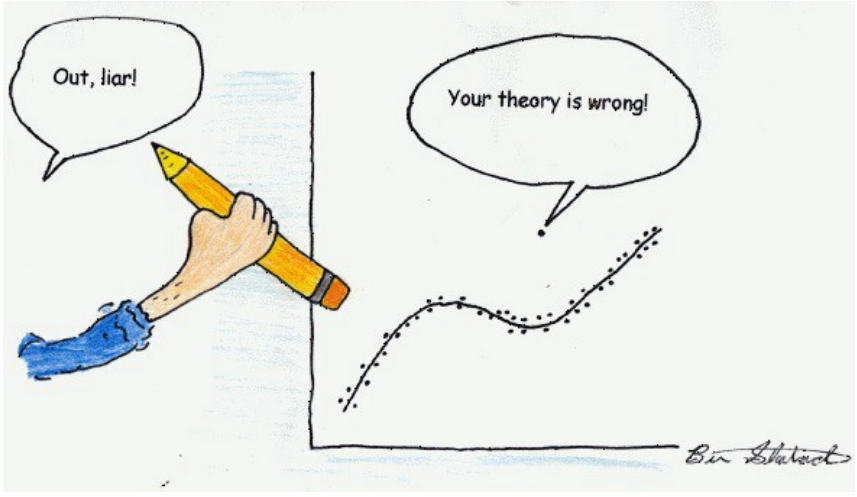


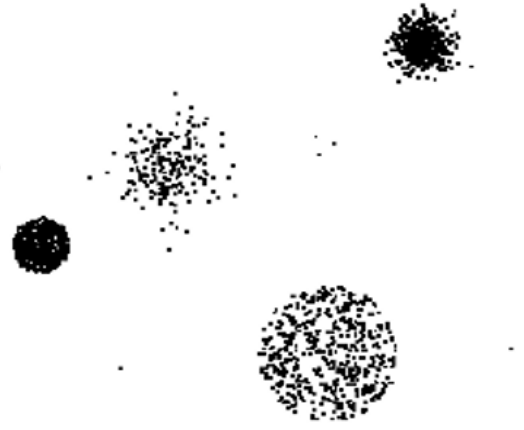
- **Natural variation**
  - Unusually tall people



- **Data errors**
  - 200 pound 2 year old

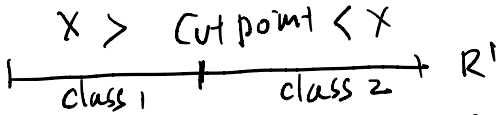






General issues

- 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

High dimensional problem

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

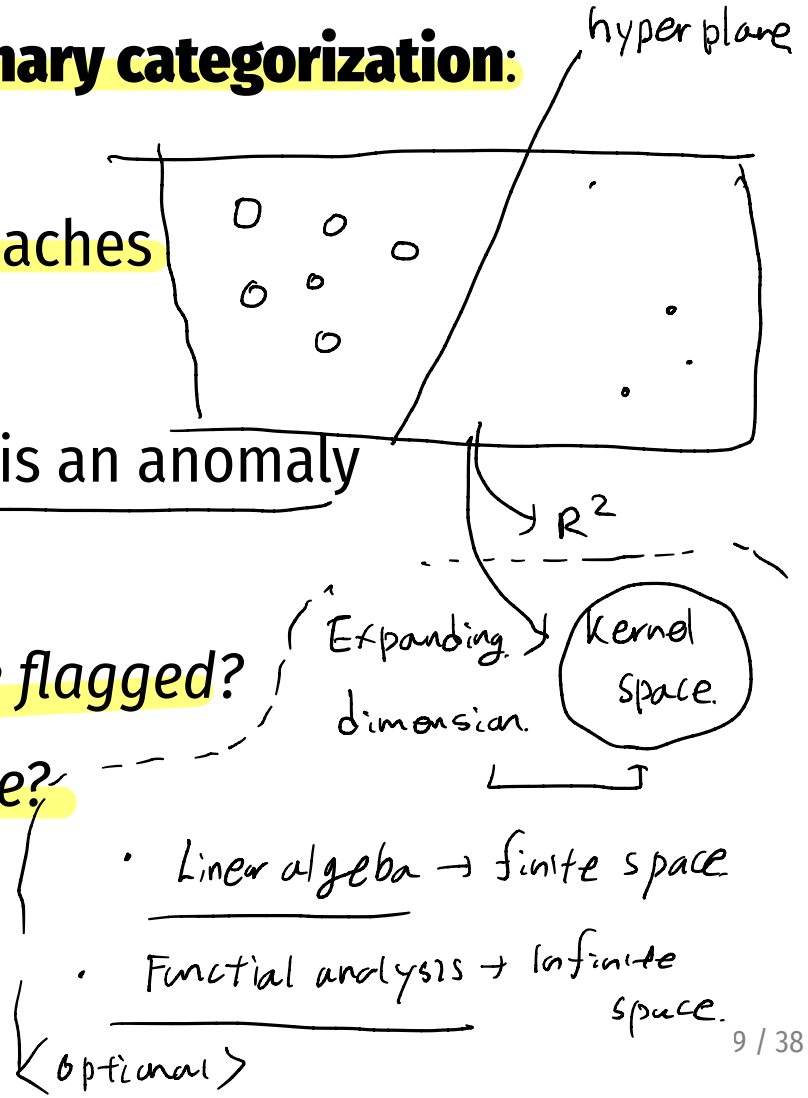
- **Find** all anomalies at once or one at a time
- **Evaluation**:
  - How do you measure performance?
  - Supervised vs. unsupervised situations
- **Efficiency**
- **Context**

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

Should this credit card transaction be flagged?

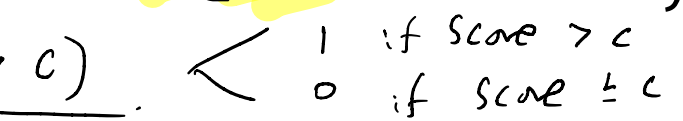
How many anomalies are there?



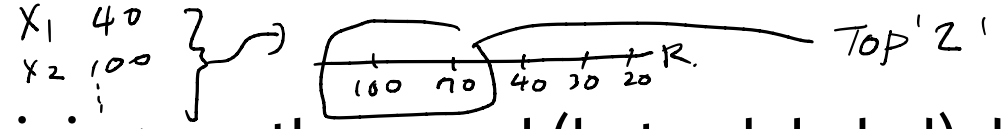
Variants of anomaly detection problems

Problem formulations

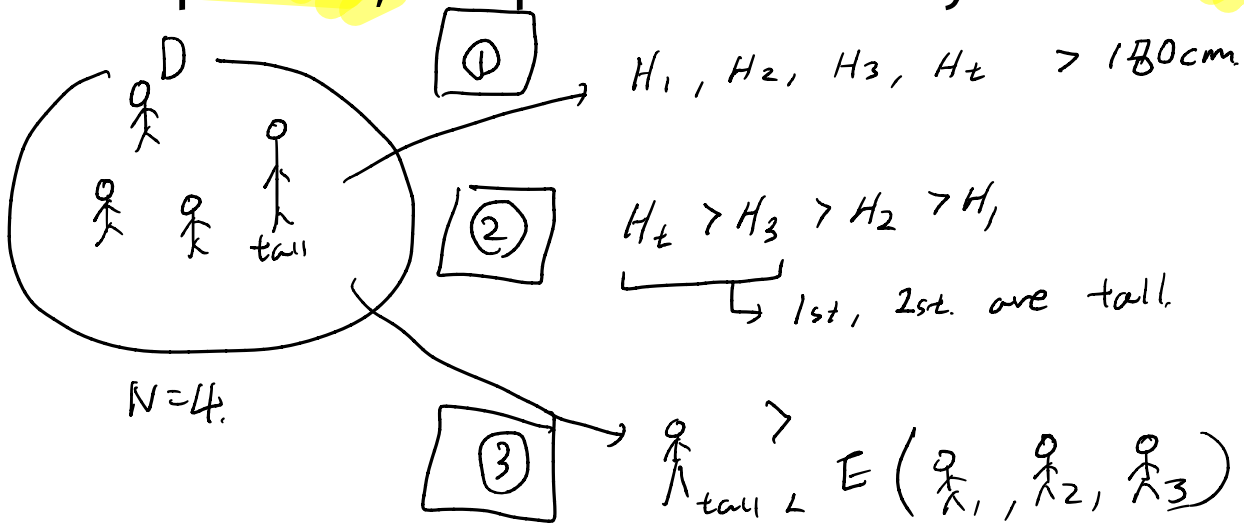
1. Given a data set $\mathbf{D}$, find all data points $\mathbf{x} \in \mathbf{D}$ with anomaly scores greater than some threshold t



2. Given a data set $\mathbf{D}$, find all data points $\mathbf{x} \in \mathbf{D}$ having the top-n largest anomaly scores



3. Given a data set $\mathbf{D}$, containing mostly normal (but unlabeled) data points, and a test point $\mathbf{x}$, compute the anomaly score of $\mathbf{x}$ with respect to $\mathbf{D}$



Model-based anomaly detection

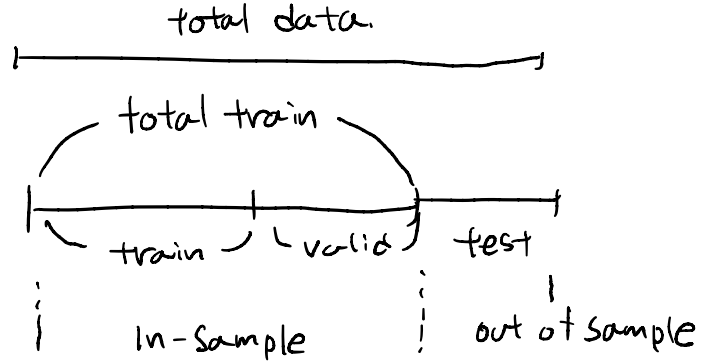
Build a model for the data and study it

Unsupervised

- Anomalies are those points that do not fit well
- Anomalies are those points that distort the model
- Examples: statistical distribution, clusters, regression, graph

Supervised → classification → abnormal / normal

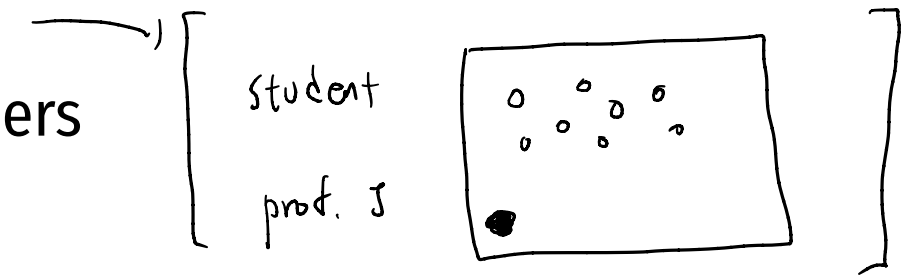
- Anomalies are regarded as a rare class
- Need to have training data



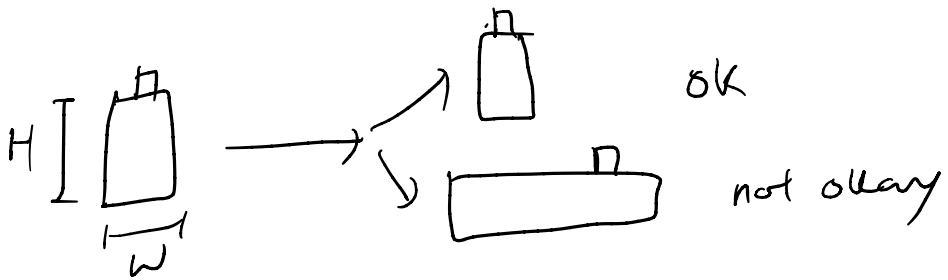
Other anomaly dteection techniques

- **Proximity-based**
  - Anomalies are points far away from other points
  - Can detect this graphically in some cases

- **Density-based**
  - Low density points are outliers



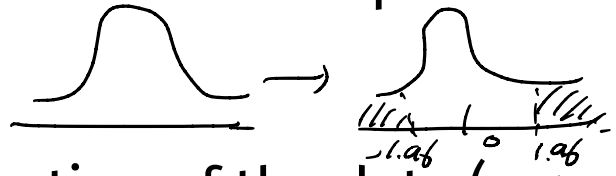
- **Pattern matching**
  - Create profiles or templates of atypical but important events or objects
  - Algorithms to detect these patterns are usually simple and efficient



Visual and statistical approaches

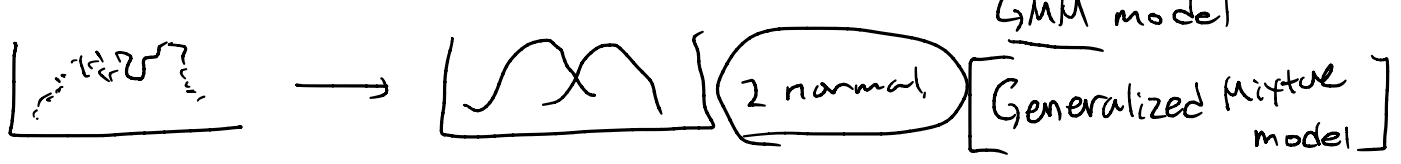
Boxplots and/or scatterplots for exploration. A limitation is that this is not automatic and can be subjective

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.



compared the mean of two groups, Hypothesis test

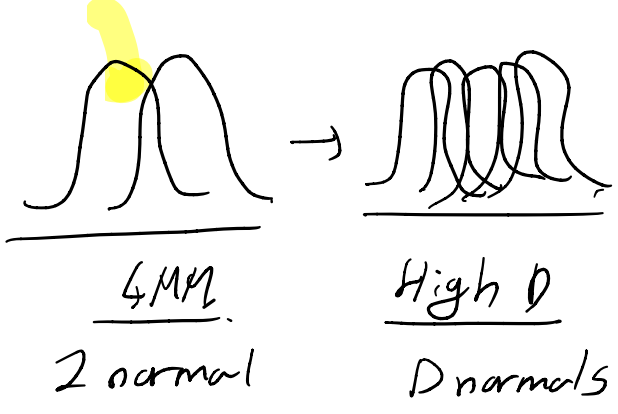
- Usually assume a **parametric model** describing the distribution of the data (e.g., normal distribution)
- Apply a **statistical test** that depends on: data distribution, parameters of distribution (e.g., mean, variance), number of expected outliers (confidence limit)
- Some issues include: identifying the distribution of a data set, number of attributes to considered, is the data a **mixture of distributions**?



About statistical approaches

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

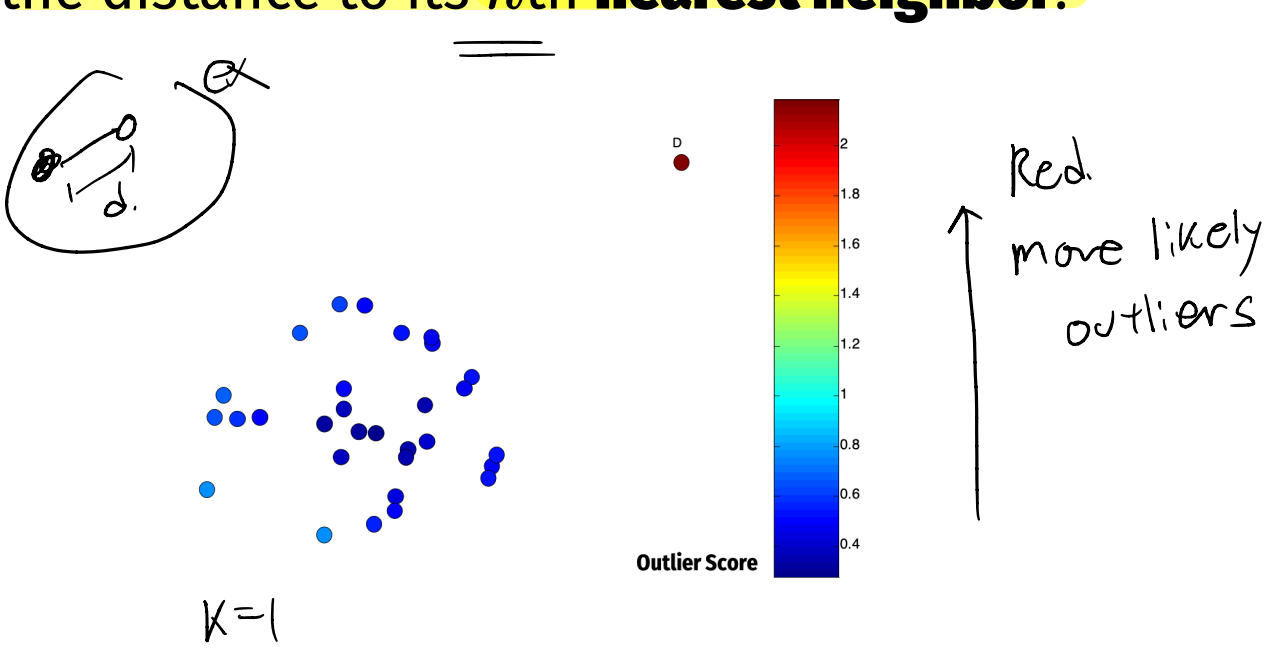
Examples include: Grubbs' Test, and the likelihood approach



Distance-based approaches

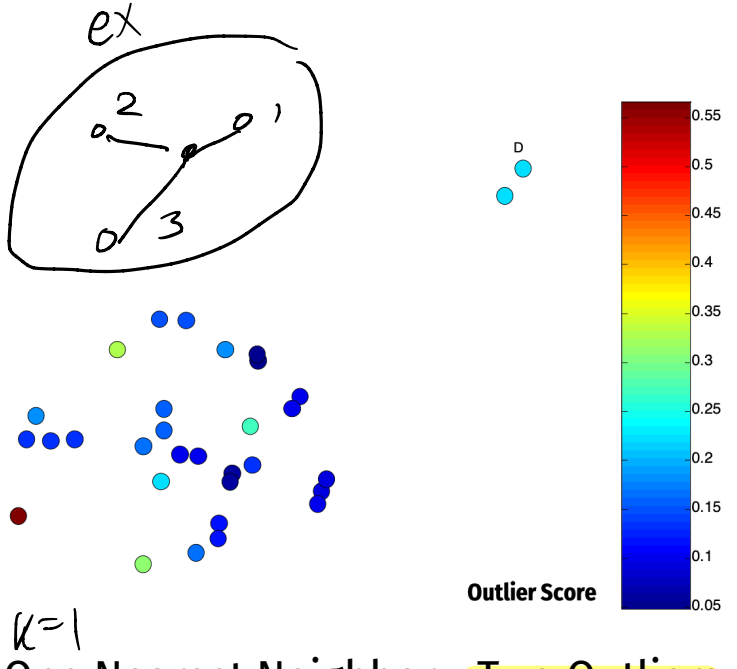
There are several different techniques. An object is an **outlier if a specified fraction of the objects is more than a specified distance away** (Knorr, Ng 1998)

Some statistical definitions are special cases of this. The **outlier score** of an object is the distance to its kth **nearest neighbor**.



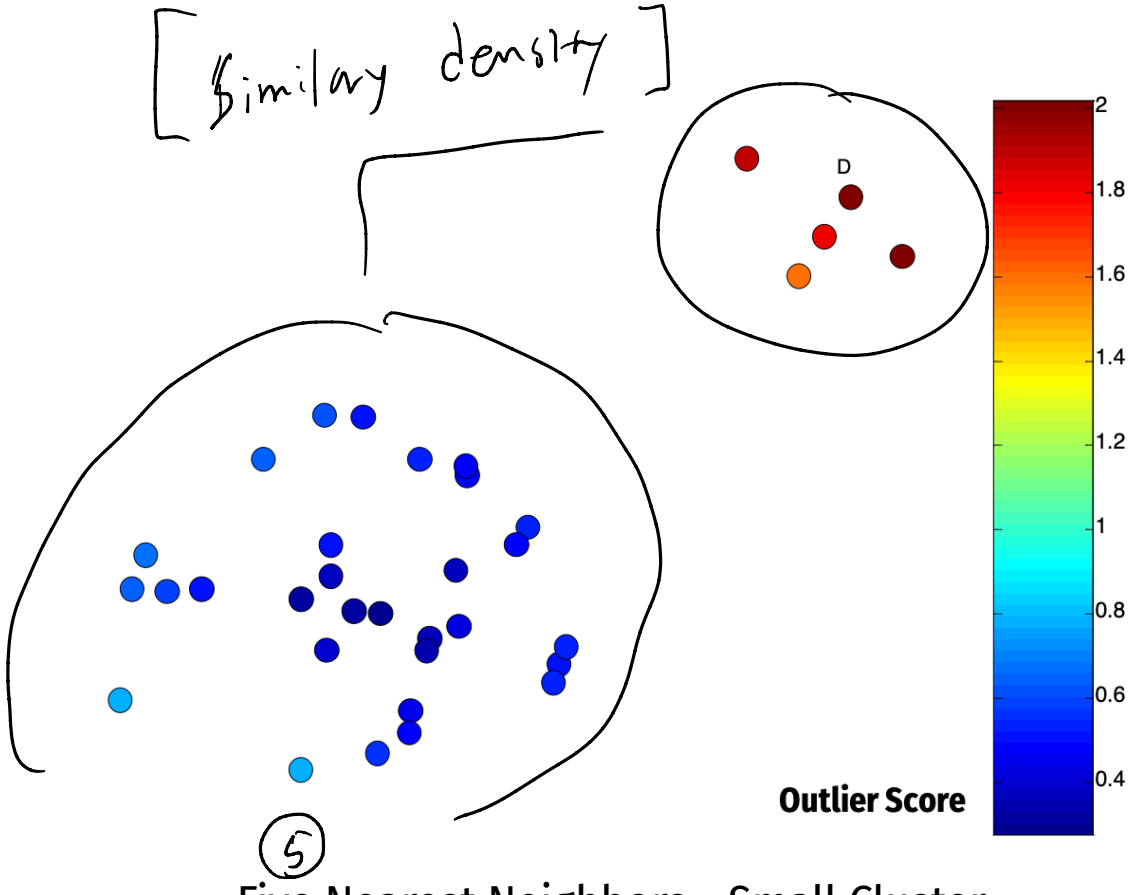
Red
more likely
outliers

One Nearest Neighbor - One Outlier

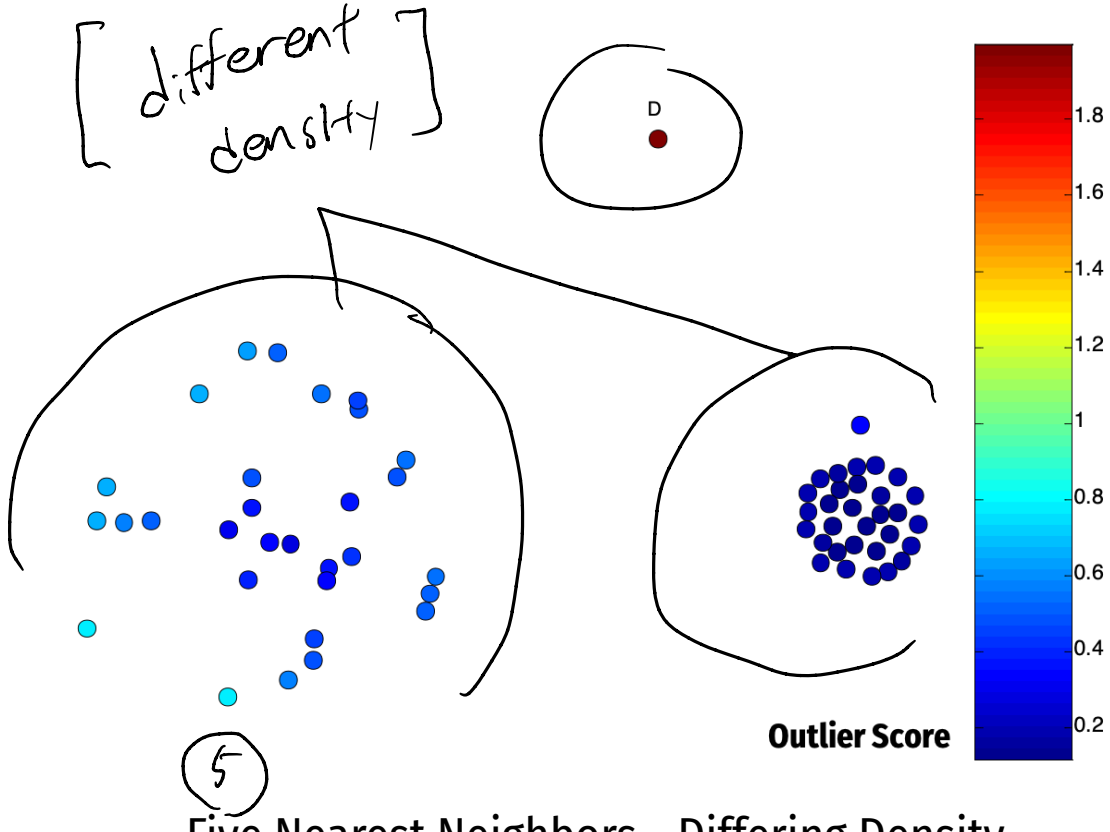


One Nearest Neighbor - Two Outliers

Distance-based approaches (cont.)



Five Nearest Neighbors - Small Cluster



Five Nearest Neighbors - Differing Density

Density-based approaches

Density-based Outlier: The outlier score of an object is the inverse of the density around the object.

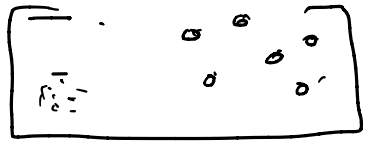
- Can be defined in terms of the k nearest neighbors
- One definition: Inverse of distance to kth neighbor

The outlier score $= \frac{1}{\text{density score.}}$

$= \frac{1}{\text{distance to kth neighbor}}$

_ Another definition: Inverse of the average distance to k neighbors

- **DBSCAN** definition



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

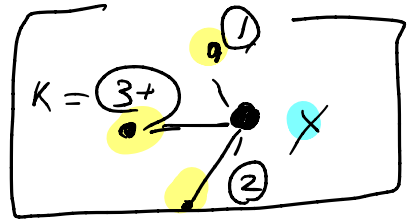
- Simple, may be expensive. It can be sensitive to parameters, and desnsity becomes less meaningful in high-dimensional space.

Relative density outlier scores

Relative density: density of a point relative to that of its k nearest neighbors. Average relative density (ARD):

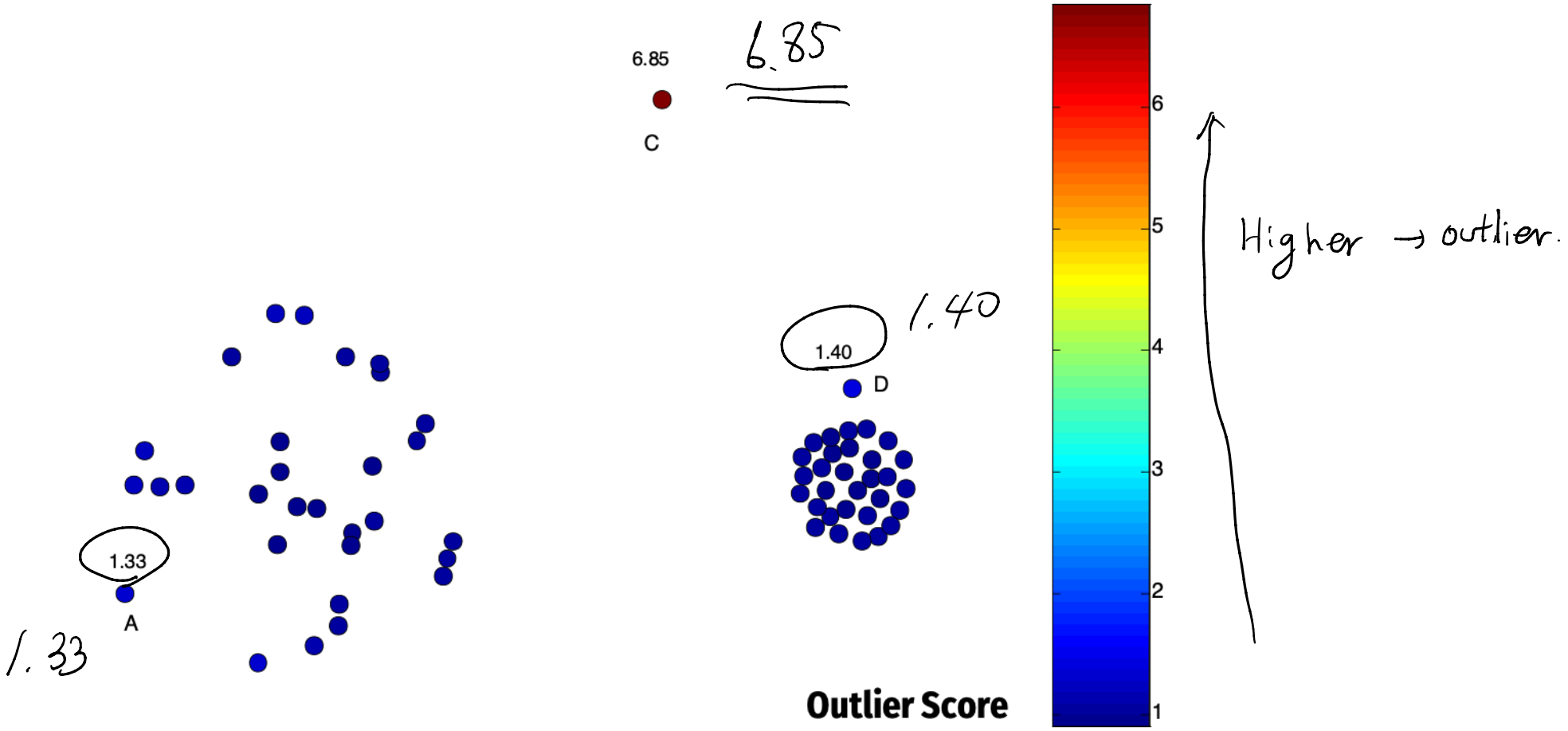
Average relative density $= ARD(\mathbf{x}, k) = \dfrac{\text{density}(\mathbf{x}, k)}{\sum_{\mathbf{y} \in \mathcal{N}(\mathbf{x},k)} \text{density}(\mathbf{y}, k) / |\mathcal{N}(\mathbf{x}, k)|} = \dfrac{|N(x,k)| \cdot den(x,k)}{\sum_{y \in N(x,k)} den(y,k)}$

1. **for all** objects $\mathbf{x}$ **do**
2. Determine $\mathcal{N}(\mathbf{x}, k)$ (get the k-nearest neighbors of $\mathbf{x}$)
3. Determine $\text{density}(\mathbf{x}, k)$ (compute density of $\mathbf{x}$ using its nearest neighbors)
4. **end for**
5. **for all** objects $\mathbf{x}$ **do**
6. Set the **outlier score** $\text{score}(\mathbf{x}, k) = ARD(\mathbf{x}, k)$
7. **end for**



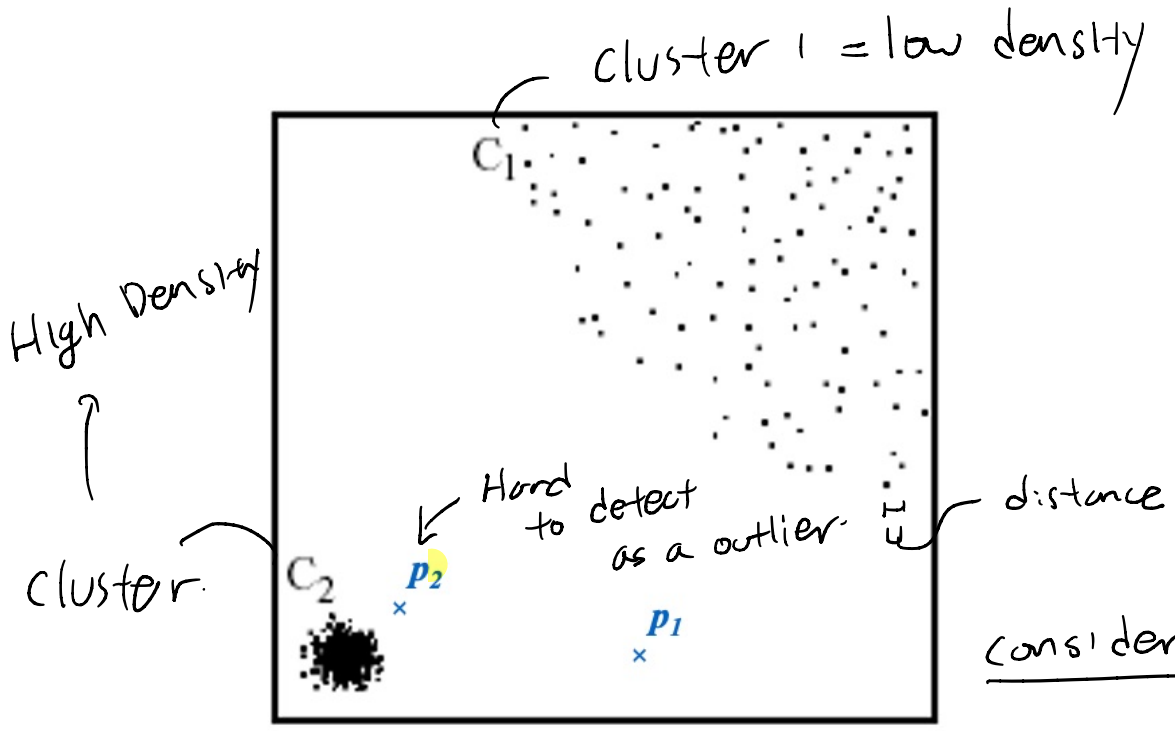
$= \text{relative score} = \dfrac{kd(x)}{\frac{kd(1)}{3} + \frac{kd(2)}{3} + \frac{kd(3)}{3}} = \dfrac{3 \cdot kd(x)}{kd(1) + kd(2) + kd(3)}$

Relative density outlier scores (cont.)



Density-based: LOF approach

- 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 the LOF approach finds both p_1 and p_2 as outliers

consider local information → LOF

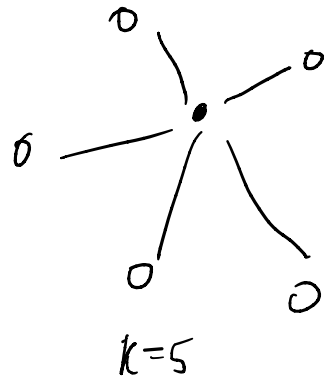
detail skip

close to 1 Normal

bigger than 1 outlier.

$$LOF = \frac{\sum_{o \in N_k(p)} \frac{lrd(o)}{lrd(p)}}{N_k(p)}$$

$$= E\left(\frac{lrd(o)}{lrd(p)}\right)$$



o where LRD means

$$lrd_k(p) = \frac{\sum_{o \in N_k(p)} (\text{reach-dist}_k(p,o))^{-1}}{N_k(p)}$$

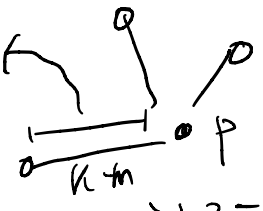
o where reach-dist means

reach-distance(p, o)

= max (k-distance(o), dist(p, o))

o where

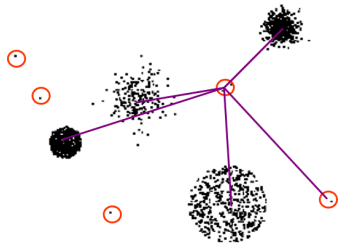
k distance



Skip,
I will post additional resource.

Breunig, M. M., Kriegel, H. P., Ng, R. T., & Sander, J. (2000, May). LOF: identifying density-based local outliers. In Proceedings of the 2000 ACM SIGMOD international conference on Management of data (pp. 93-104).

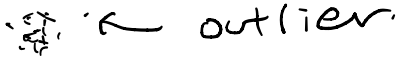
Cluster-based approaches



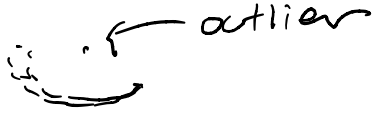
Clustering-based outlier: An object is a cluster-based outlier if it **does not strongly belong to any cluster**

For **prototype-based clusters**, an object is an outlier if it is **not close enough to a cluster center**

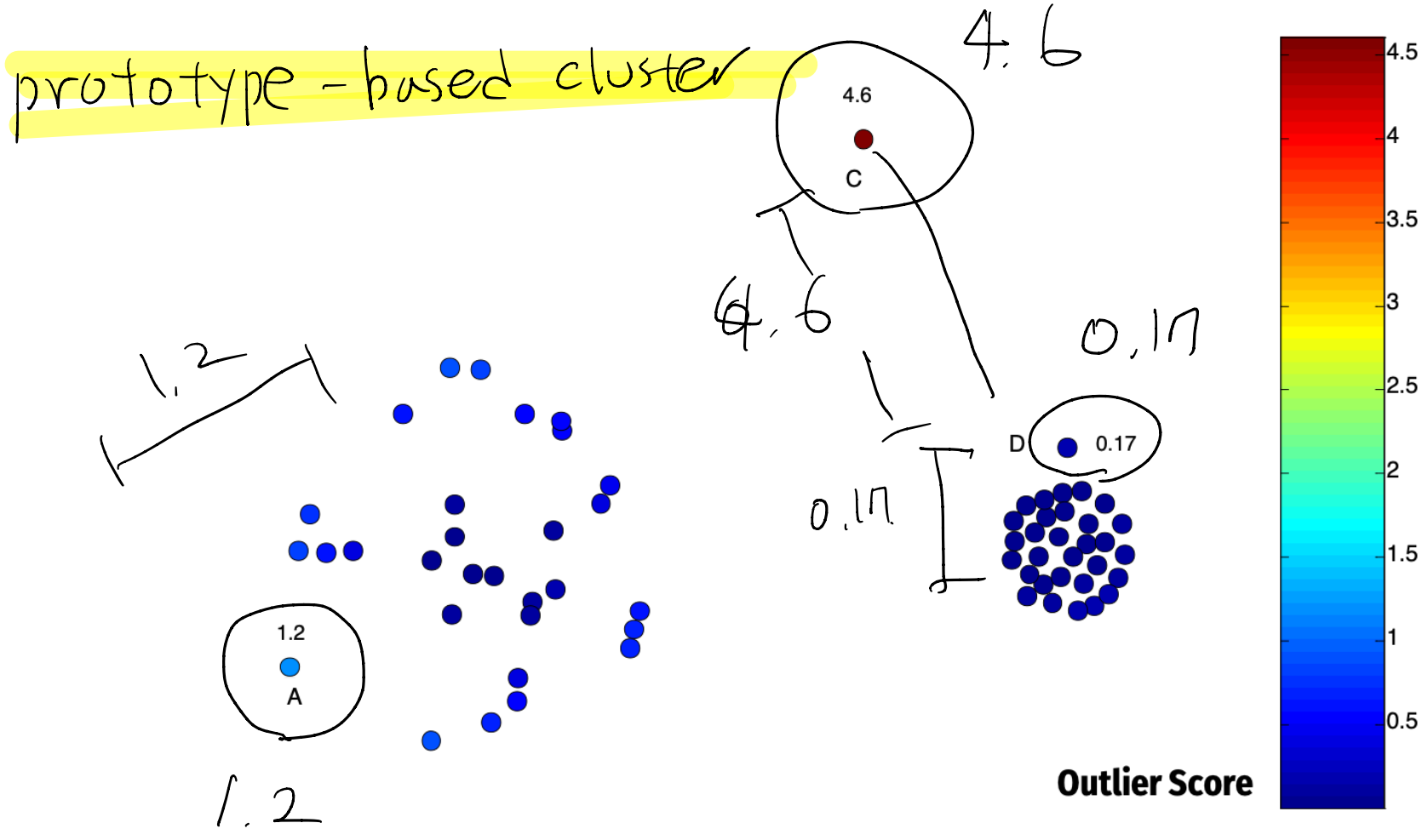
For **density-based clusters**, an object is an outlier if its **density is too low**



For **graph-based clusters**, an object is an outlier if it is **not well connected**

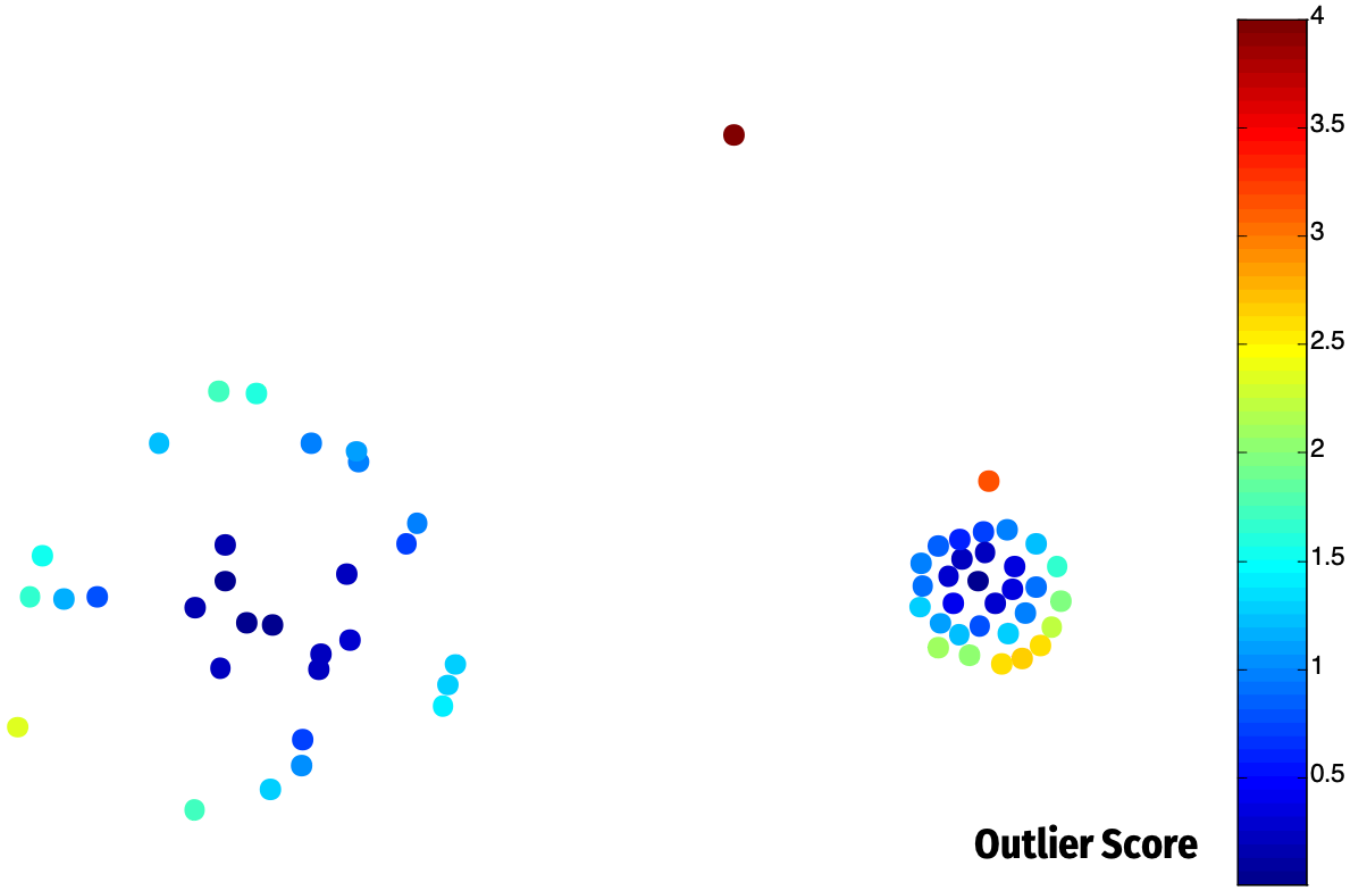


Illustration



Distance of Points from Closest Centroids

Illustration (cont.)



Relative Distance of Points from Closest Centroid

Thinks to keep in mind

For clustering-based approaches:

- Implementation is simple
- There are many clustering techniques that can be used
- It can be difficult to decide on a clustering technique
- It can be difficult to decide on number of clusters
- Outliers can distort the clusters

Example: Univariate Outlier Detection

Simulated data

Create a vector `x` with 100 entries sampled from the standard normal distribution

$N(0,1)$

```
set.seed(3147)      # set a seed for reproduc
x <- rnorm(100)    # use rnorm() to sample
```

Obtain a summary of `x`:

```
summary(x)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.
## -3.3154 -0.4837  0.1867  0.1098  0.7120  1
```

We can get outliers using the `boxplot()` function

```
boxplot.stats(x)$out
```

```
## [1] -3.315391  2.685922 -3.055717  2.5712(
```

```
boxplot(x)
```

Emsemble approach

The above univariate outlier detection can be used to find outliers in multivariate data in a simple ensemble way. As an example, we first generate a dataframe `df`, which has two columns, `x` and `y`. Then, outliers are detected separately from `x` and `y`. We then take outliers as those data which are outliers for **both** columns.

```
# create vector with samples from the norma
y <- rnorm(100)
# create data frame with x and y
df <- data.frame(x, y)
head(df)   # print first 6 rows
```

```
##             x          y
## 1 -3.31539150  0.7619774
## 2 -0.04765067 -0.6404403
## 3  0.69720806  0.7645655
## 4  0.35979073  0.3131930
## 5  0.18644193  0.1709528
## 6  0.27493834 -0.8441813
```

```
# find the index of outliers from x
(a <- which(x %in% boxplot.stats(x)$out))
```

```
## [1]  1 33 64 74
```

```
# find the index of outliers from y
(b <- which(y %in% boxplot.stats(y)$out))
```

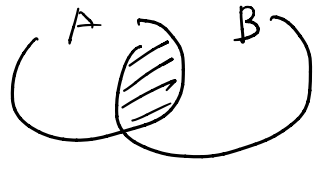
```
## [1] 24 25 49 64 74
```

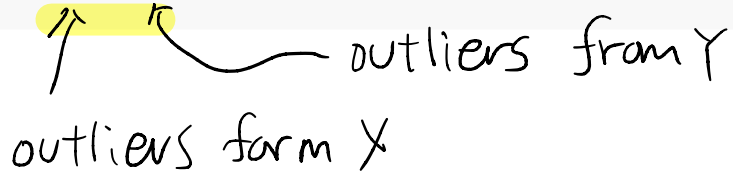
Both (`intersect()`), either (`union()`)

Get **outliers in both** with the `intersect()` function:

```
(outlier_list_1 <- intersect(a,b))
```

```
## [1] 64 74
```

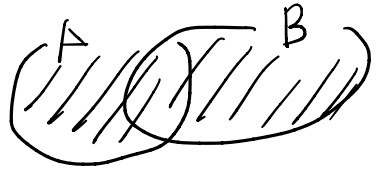




We can also take outliers as those data which are **outliers in either** `x` or `y`, using the `union()` function:

```
(outlier_list_2 <- union(a,b))
```

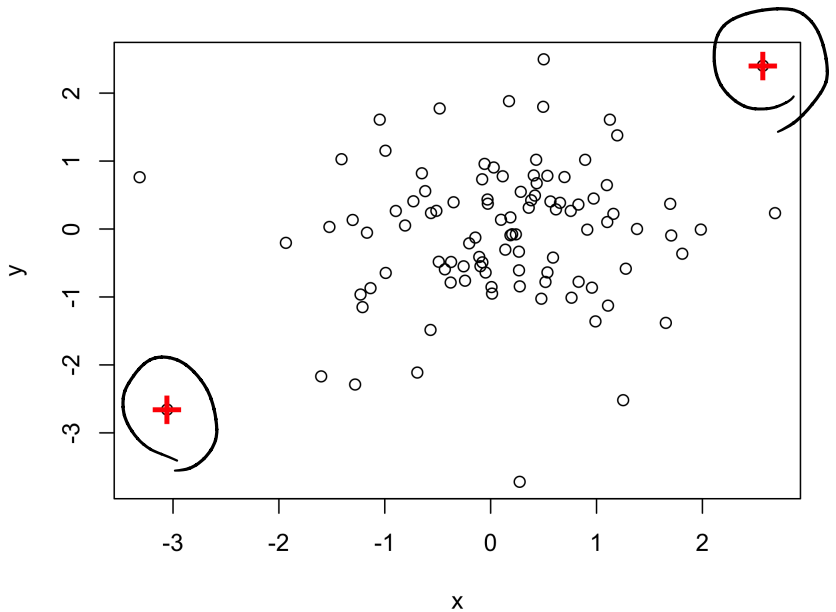
```
## [1]  1 33 64 74 24 25 49
```



Visualization: univariate outlier

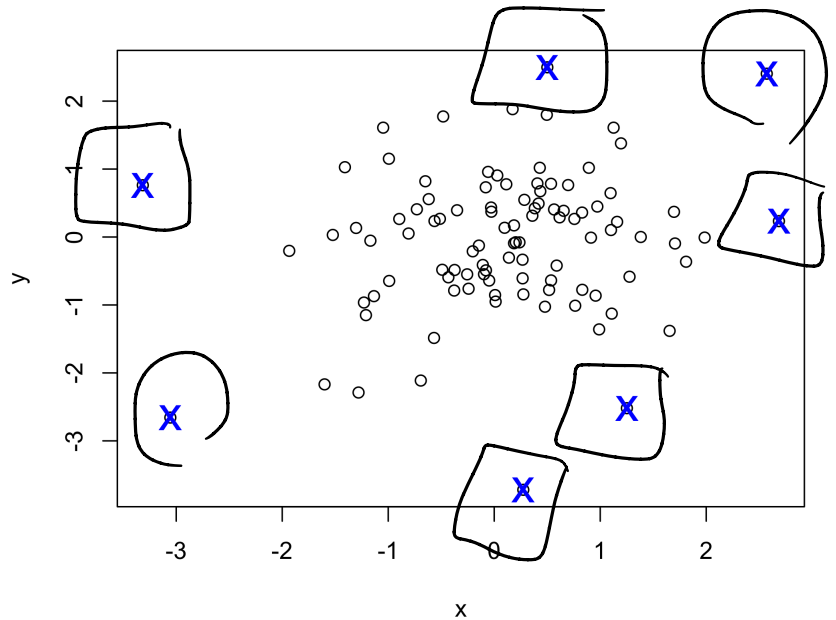
outliers in both ∩

```
plot(df)  #simple plot
points(df[outlier_list_1, ],
       col = "red", pch = "+", cex = 2.5)
```



outliers in either ∪

```
plot(df)
points(df[outlier_list_2, ],
       col = "blue", pch = "x", cex = 2)
```



Example: Outlier Detection with LOF

Local outlier factor

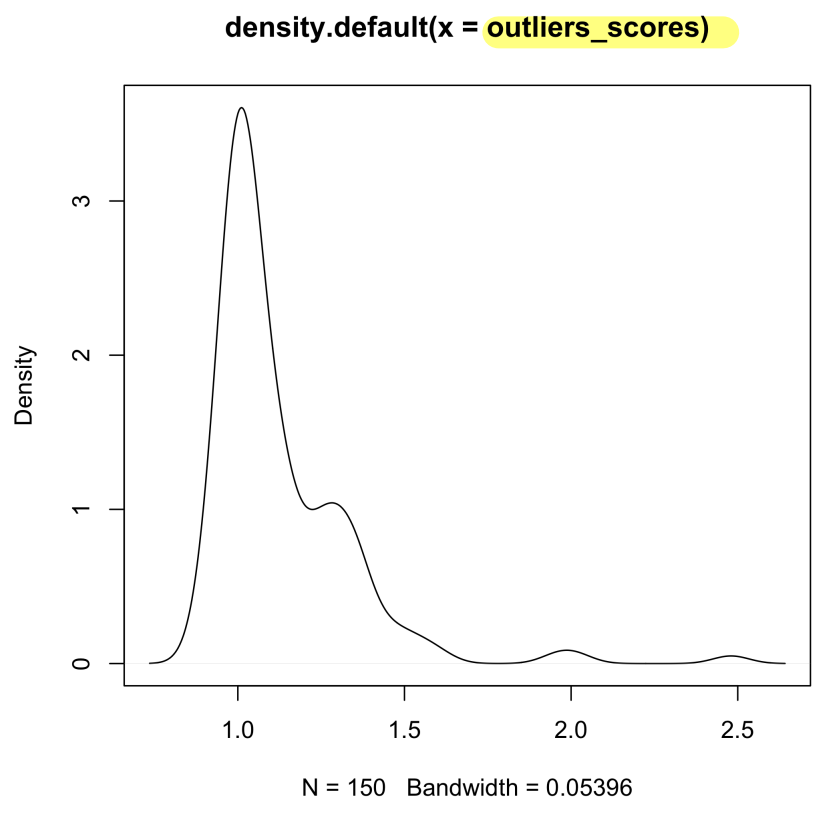
LOF (Local Outlier Factor) is an algorithm for identifying density-based local outliers (Breunig et al., 2000). With LOF, the local density of a point is compared with that of its neighbors. If the former is significantly lower than the latter (with an LOF value greater than one), the point is in a sparser region than its neighbors, which suggests it be an outlier.

```
library(dbscan)
# remove "Species", which is a categorical column
iris_2 <- iris[ , 1:4]
```

We use the `dbscan::lof()` function, which calculates the Local Outlier Factor (LOF) score for each data point using a kd-tree to speed up kNN search. A LOF score of approximately 1 indicates that density around the point is comparable to its neighbors. Scores significantly larger than 1 indicate outliers.

Density plot of LOF scores

```
outliers_scores <- lof(iris_2, k = 5)
# density plot of the different scores
plot(density(outliers_scores))
```



Find out the *identity* of the outliers:

```
outliers <- order(outliers_scores, # pick to
                  decreasing = T)[1:5]
print(outliers)
```

```
## [1]  42 107  23 110  63
```

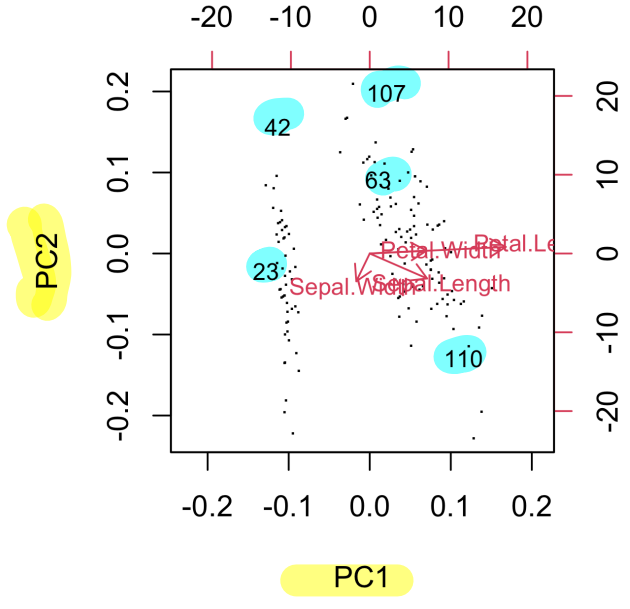
```
# attributes for each outliers
iris_2[outliers, ]
```

```
##     Sepal.Length Sepal.Width Petal.Length
## 42           4.5         2.3          1.3
## 107          4.9         2.5          4.5
## 23           4.6         3.6          1.0
## 110          7.2         3.6          6.1
## 63           6.0         2.2          4.0
```

Visualization: LOF

We show outliers with a **biplot** of the first two *principal components*

```
n <- nrow(iris_2) # number of observations
labels <- 1:n
# label points EXCEPT outliers with a "."
labels[-outliers] <- "."
# generate biplot
biplot(prcomp(iris_2), cex = .8,
       xlabs = labels)
```



`prcomp()` performs PCA, and `biplot()` plots the data with its first two principal components. The `x`- and `y`-axis are respectively the first and second principal components.

The arrows show the original columns (variables), and the five outliers are labeled with their row numbers.

Example: Outlier Detection by Clustering

Outlier Detection by Clustering

[DBSCAN]

Another way to detect outliers is clustering. By grouping data into clusters, those data not assigned to any clusters are taken as outliers. For example, with density-based clustering such as DBSCAN (Ester et al., 1996), objects are grouped into one cluster if they are connected to one another by **densely populated areas**. Therefore, objects not assigned to any clusters are isolated from other objects and are taken as outliers.

[K-mean]

We can also detect outliers with the K-means algorithm. With K-means, the data are partitioned into K groups by assigning them to the closest cluster *centers*. After that, we can calculate the distance (or dissimilarity) between each object and its cluster center, and pick those with largest distances as outliers.

> We perform K-means with 3 groups in the `iris_2` dataset

[outlier far from a cluster]

K-means for outlier detection

```
kmeans_result <- kmeans(iris_2, centers = 3)
# get cluster centers
kmeans_result$centers
```

3 clusters

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width
## 1     5.901613    2.748387     4.393548    1.433871
## 2     6.850000    3.073684     5.742105    2.071053
## 3     5.006000    3.428000     1.462000    0.246000
```

cluster centers

To see the assignment to each cluster, we can check the `.$cluster` component of the K-means object. Next, we identify the outliers by first **computing the distance from every point** to its corresponding center (as found by the K-means algorithm), and then **sorting the distance vector** to select the top 5 observations that fell far away from the center of their corresponding cluster.

Visualization: clusters and outliers

```
# calculate distances between objects and c
centers <- kmeans_result$centers[kmeans_resu
# compute Euclidean distance
km_distances <- sqrt(
          rowSums((iris_2 - centers) ^ 2))
# pick top 5 largest distances
km_outliers <- order(km_distances, decreasi
```

```
library(tidyverse)
ggplot()+
  geom_point(data=iris_2,
             aes(x = Sepal.Length, y = Sepa
                 color = factor(kmeans_resu
  geom_point(aes(x = iris_2[km_outliers, "Se
                 y = iris_2[km_outliers, "Se
             color = "black", size = 3) +
  labs(title = "Outlier Detection with K-mea
       subtitle = "Outliers in black and lar
       color = "Groups") + theme_minimal()
```