Data Mining
Lecture 13: Outlier Detection

Jo Grundy

ECS Southampton

March 28, 2022

Outlier Detection

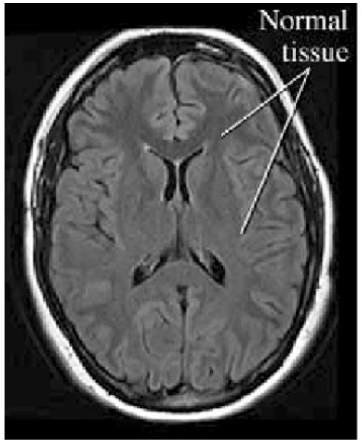
Bank statement:

- 2.50 Artemis Olive
- 9.99 NETFLIX.COM
- 1.50 THE BRIDGE
- 7.20 Sainsbury's
- 32.99 Amazon
- 4.00 THE BRIDGE
- 1.75 THE SHOP
- 50.00 CASH LONDON
- 5.10 BREWHOUSE AND KITC

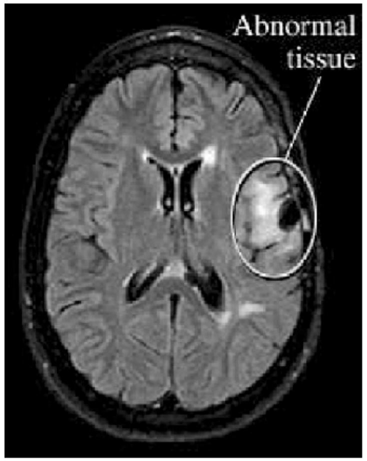
Do all of these look right?

Outlier Detection

If you see lots of scans that look like this:

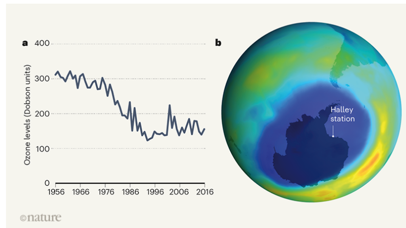


Then it is easier to see that there is something wrong here



Outlier Detection

Man with BMI of 28,000 gets offered COVID vaccine (In Jan 2021) .. listed as having height of 6.2 cm rather than 6'2". `https://www.bbc.co.uk/news/uk-england-merseyside-56111209`



Ozone Layer data - depletion was originally ignored by NASA as algorithms flagged it as bad data

Outlier Detection

A Data mining approach:

- Model the data
- What does not fit is outlier

Can use many different models
Need:

- a measure of fit

Outlier Detection

We can model data using a Gaussian distribution:
Univariate:

$$p(x) = \frac{1}{2\sqrt{2\pi}} \exp \frac{-\frac{1}{2}(x-\mu)^2}{\sigma^2}$$

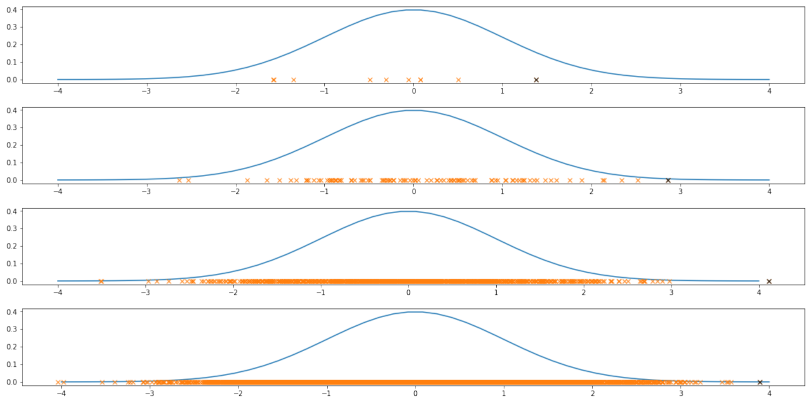
Estimate mean:

- $\mu = \frac{1}{N}\sum_{i=1}^{N} x_i$

Estimate standard deviation:

- $\sigma = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)$

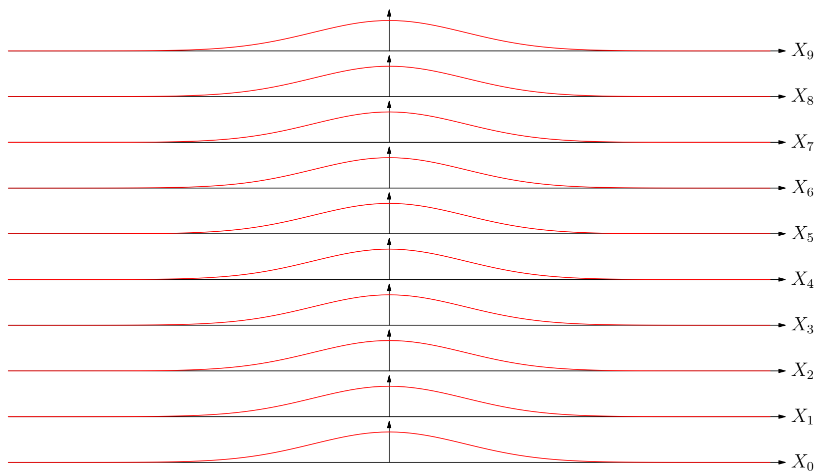
Outlier Detection



How 'outlier' a point looks depends on how many data points there are.

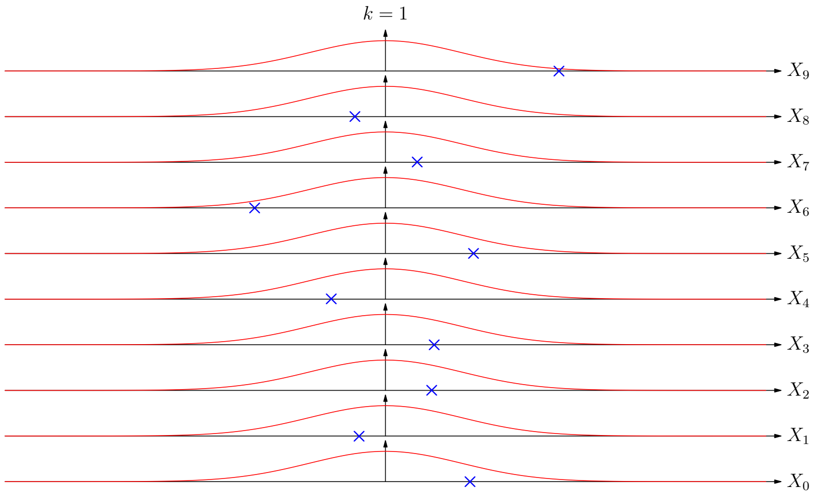
Outlier Detection - Extreme Values

How do we separate values that are just randomly different, due to noise?



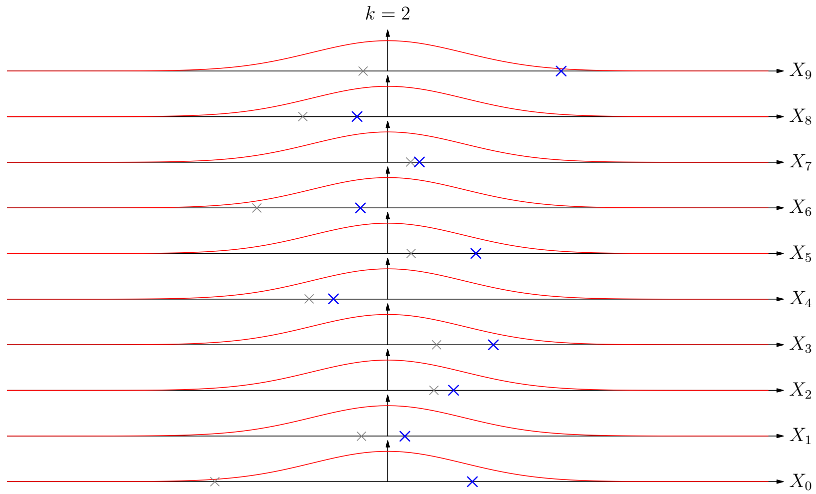
Outlier Detection - Extreme Values

How do we separate values that are just randomly different, due to noise?



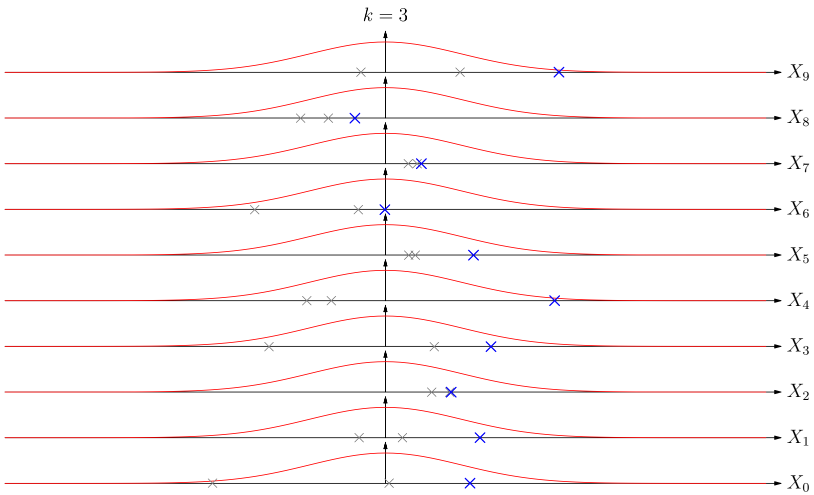
Outlier Detection - Extreme Values

How do we separate values that are just randomly different, due to noise?



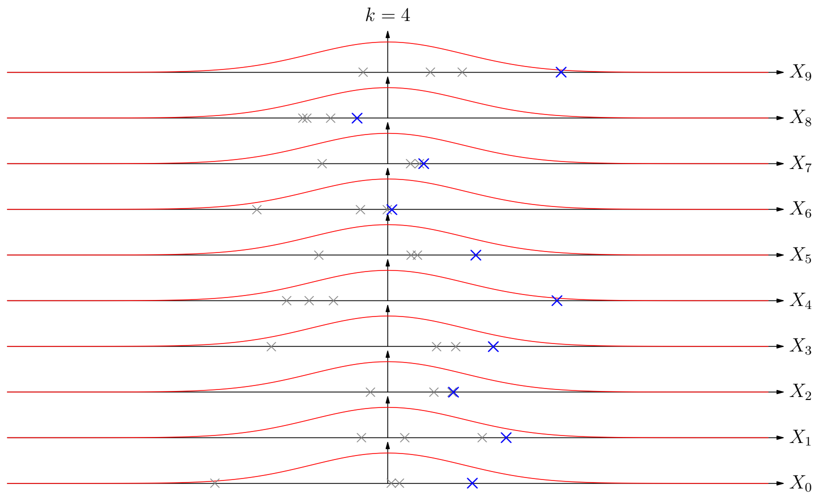
Outlier Detection - Extreme Values

How do we separate values that are just randomly different, due to noise?



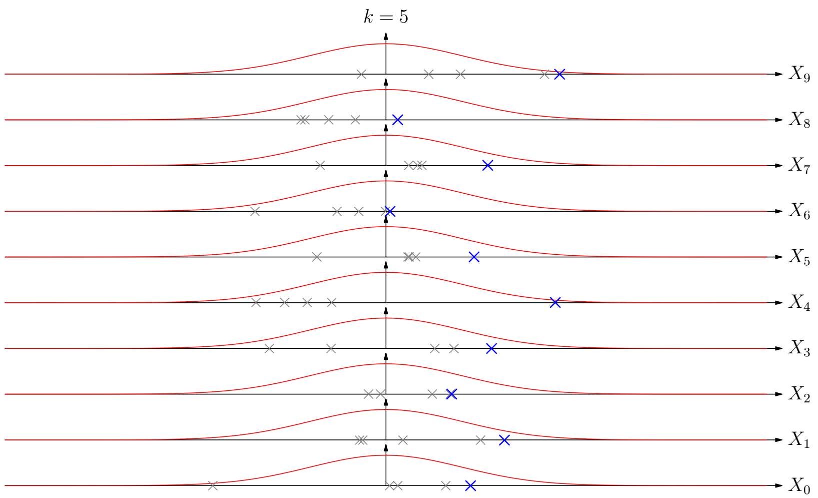
Outlier Detection - Extreme Values

How do we separate values that are just randomly different, due to noise?



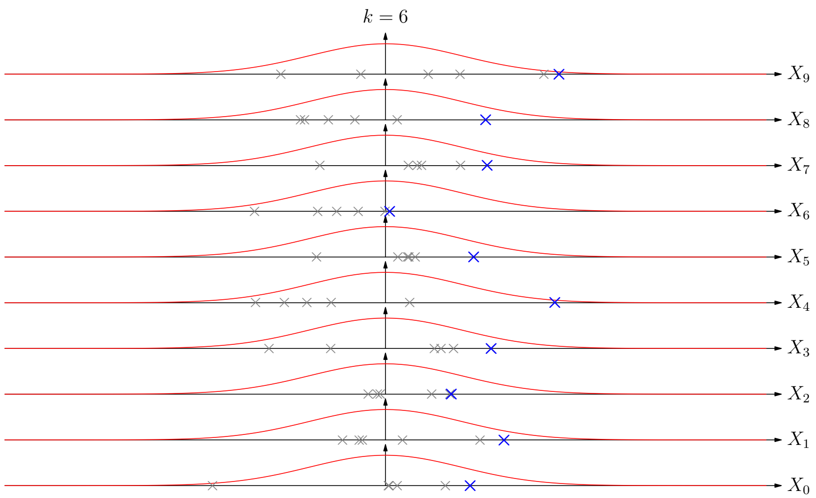
Outlier Detection - Extreme Values

How do we separate values that are just randomly different, due to noise?



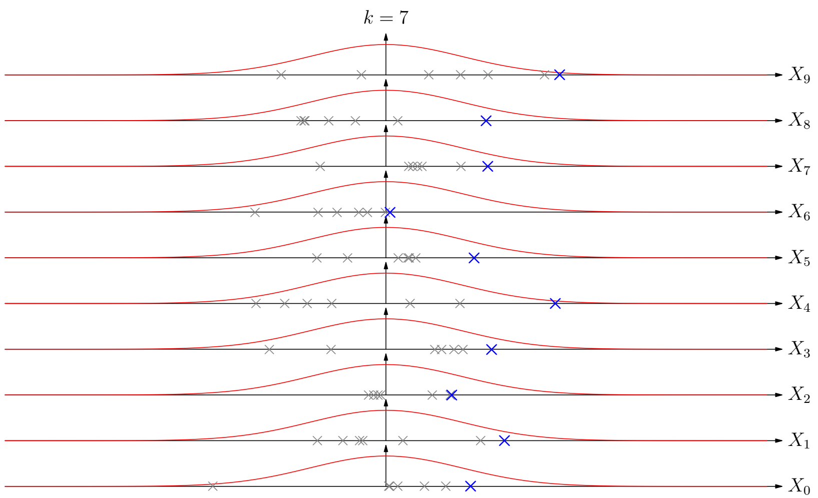
Outlier Detection - Extreme Values

How do we separate values that are just randomly different, due to noise?



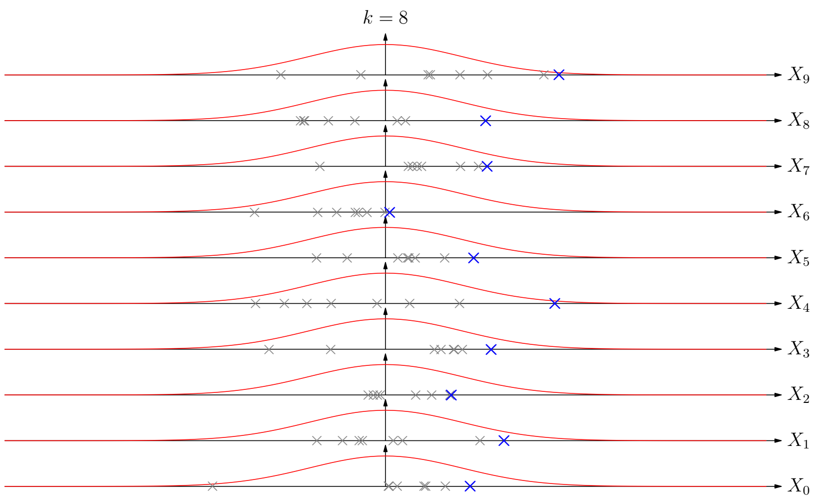
Outlier Detection - Extreme Values

How do we separate values that are just randomly different, due to noise?



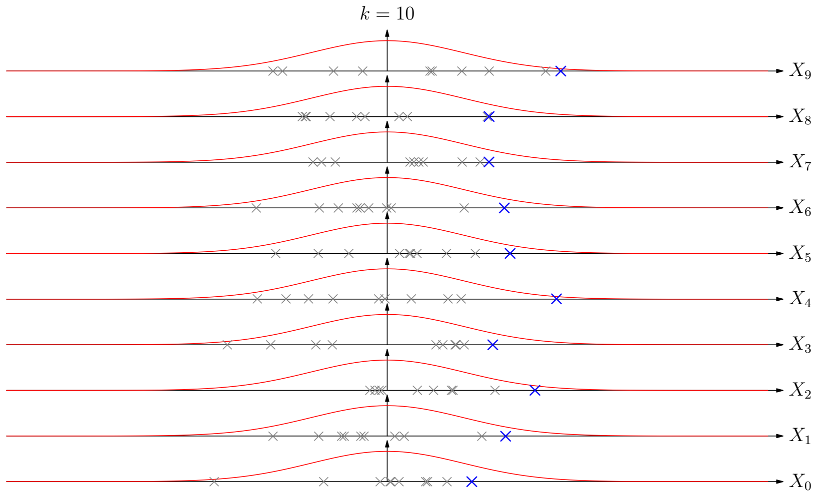
Outlier Detection - Extreme Values

How do we separate values that are just randomly different, due to noise?



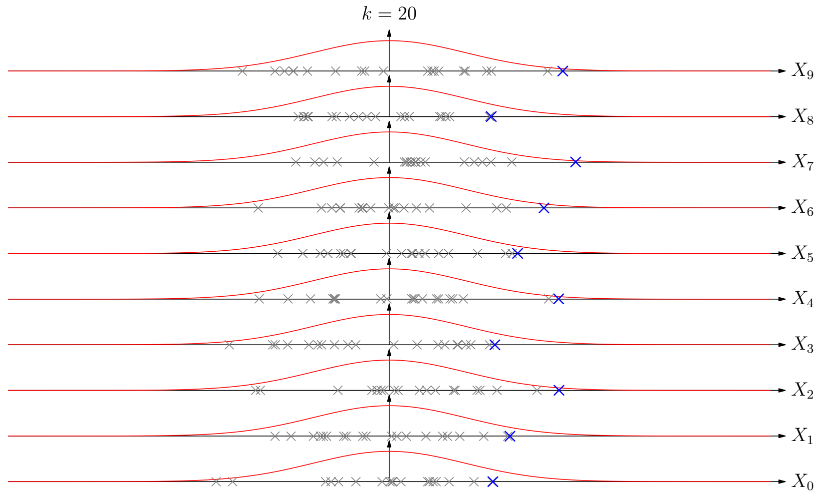
Outlier Detection - Extreme Values

How do we separate values that are just randomly different, due to noise?



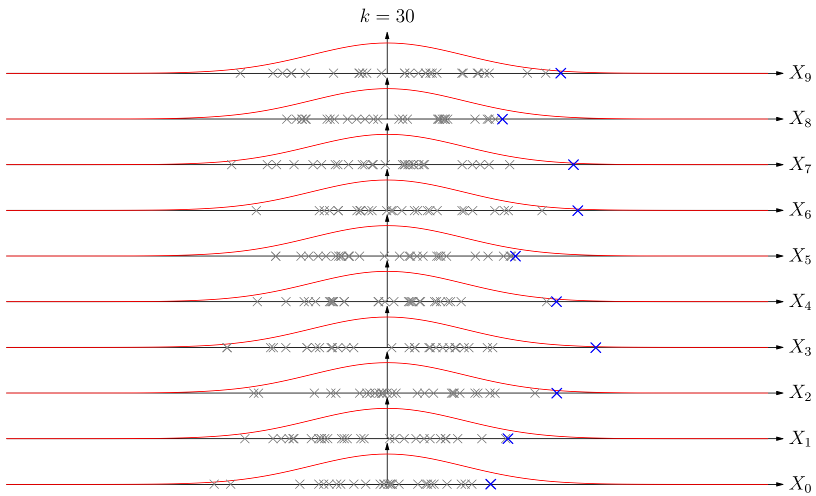
Outlier Detection - Extreme Values

How do we separate values that are just randomly different, due to noise?



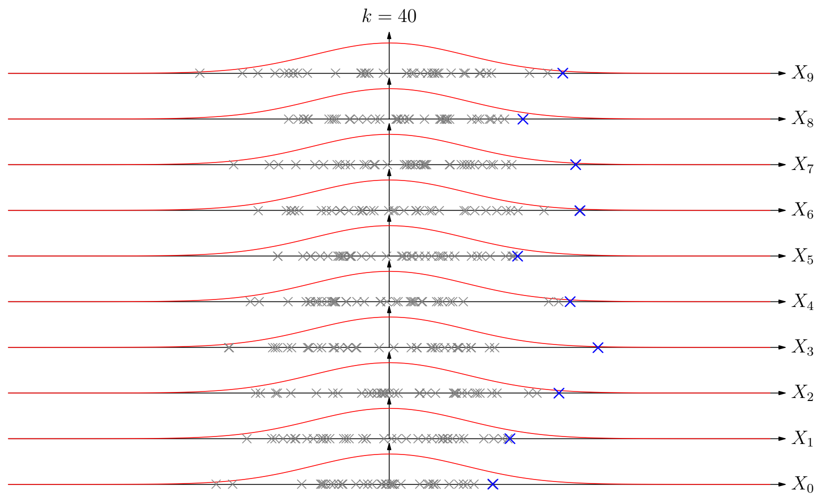
Outlier Detection - Extreme Values

How do we separate values that are just randomly different, due to noise?



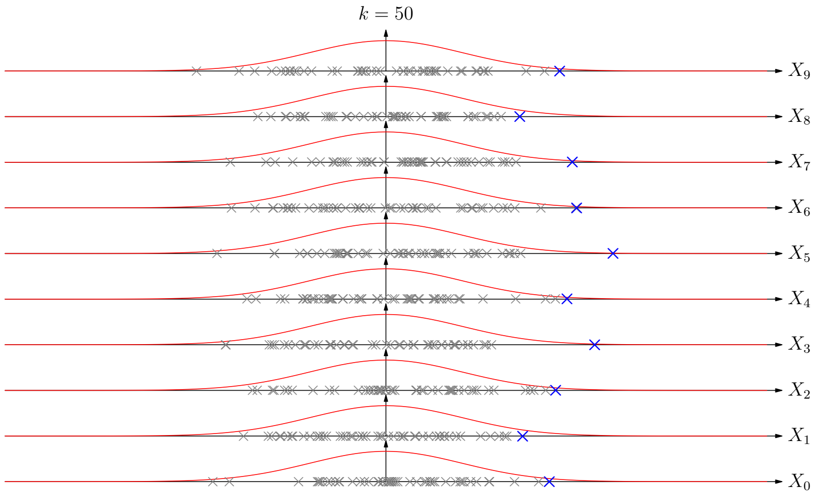
Outlier Detection - Extreme Values

How do we separate values that are just randomly different, due to noise?



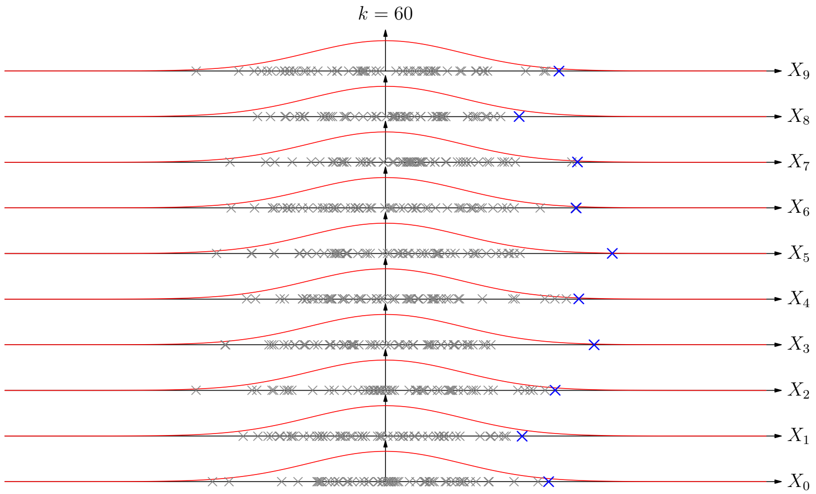
Outlier Detection - Extreme Values

How do we separate values that are just randomly different, due to noise?



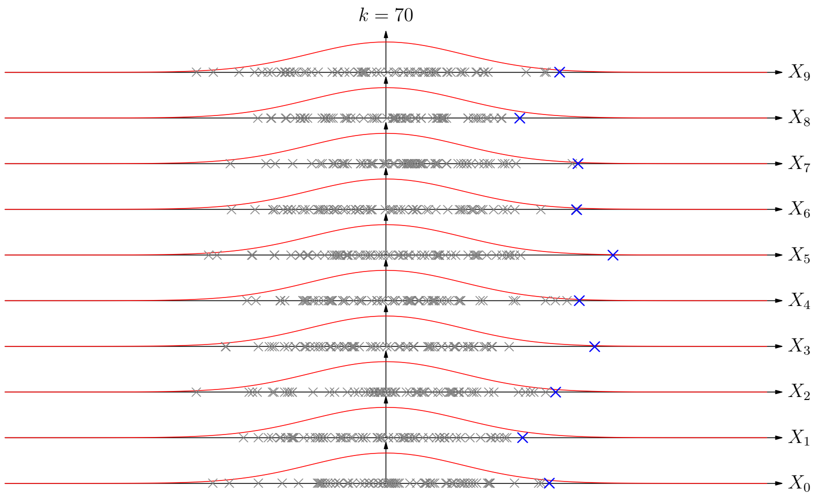
Outlier Detection - Extreme Values

How do we separate values that are just randomly different, due to noise?



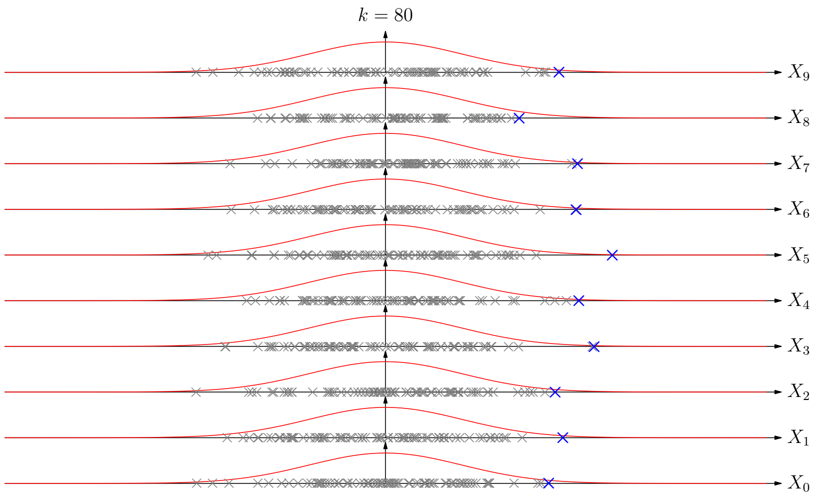
Outlier Detection - Extreme Values

How do we separate values that are just randomly different, due to noise?



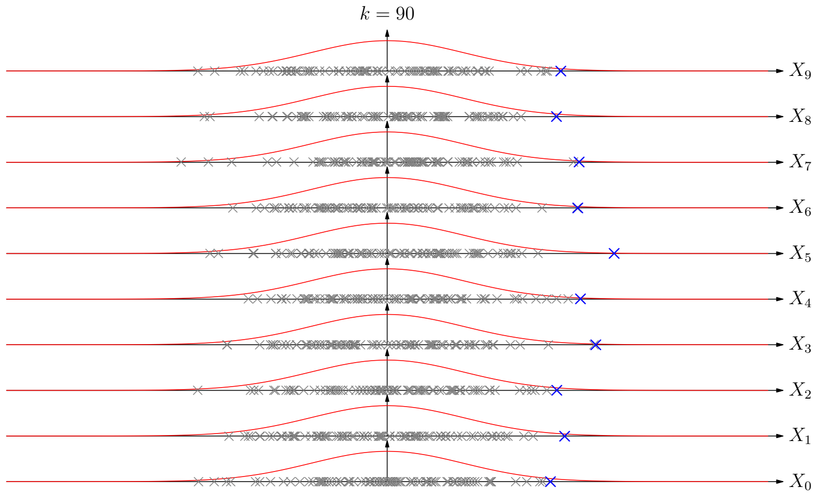
Outlier Detection - Extreme Values

How do we separate values that are just randomly different, due to noise?



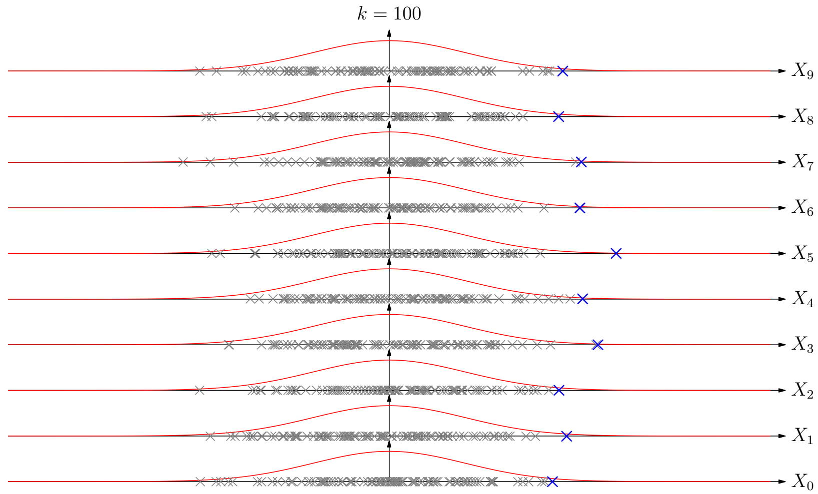
Outlier Detection - Extreme Values

How do we separate values that are just randomly different, due to noise?



Outlier Detection - Extreme Values

How do we separate values that are just randomly different, due to noise?

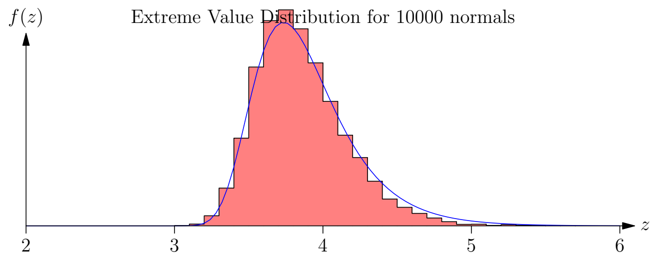


Outlier Detection - Extreme Value statistics

Extreme Value Statistics
A way to characterise extreme values using a rule similar to the central limit theorem.
Also known as the Fisher-Tippet theorem

$$f(x) \approx \frac{1}{\beta} e^{\frac{x-\mu}{\beta} - e^{\frac{x-\mu}{\beta}}}$$

Outlier Detection - Extreme Value statistics



The Weibull distribution is used here to give a probability that a value is an maximal value from a normal distribution. With more samples, the distribution is more clearly defined.

See e.g. S.J.Roberts IEE Proceedings 2000, 147,6,363-367

Outlier Detection - Gaussian Distribution

We can model the data using a multivariate Gaussian distribution:

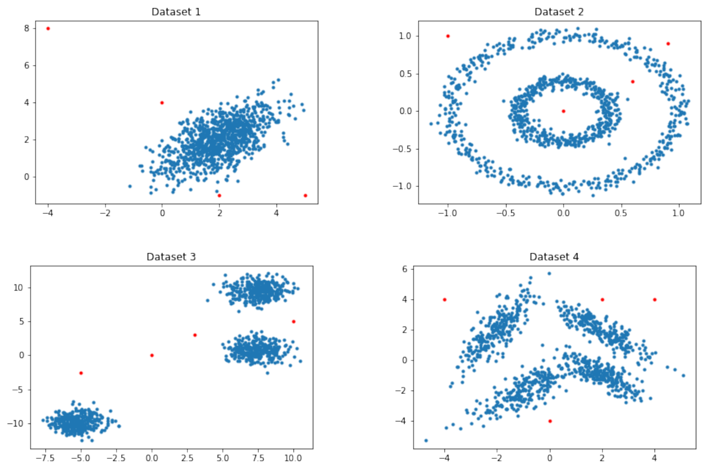
$$p(x) = \frac{1}{2\pi^{\frac{p}{2}}\sqrt{|C|}} \exp\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{m})^T C^{-1}(\boldsymbol{x}-\boldsymbol{m})\}$$

Covariance and mean can be estimated from the data.. how?
mean $= \boldsymbol{m} = \frac{1}{N}\sum_i^N x_i$
covariance is proportional to the inner product of the mean centred data
or

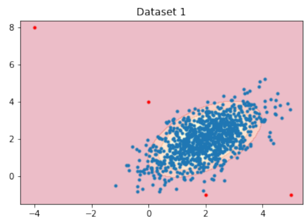
$$C = \frac{1}{N}\sum_i^N(\boldsymbol{x_i}-\boldsymbol{m})(\boldsymbol{x_i}-\boldsymbol{m})^T$$

Outlier Detection - Gaussian Distribution

For example:



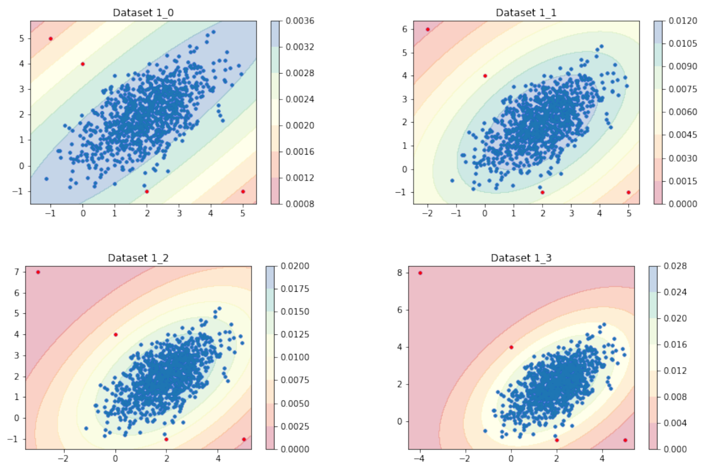
Outlier Detection - Gaussian Distribution



Fits a Gaussian distribution reasonably well.
however sensitive to outliers..

Outlier Detection - Gaussian Distribution

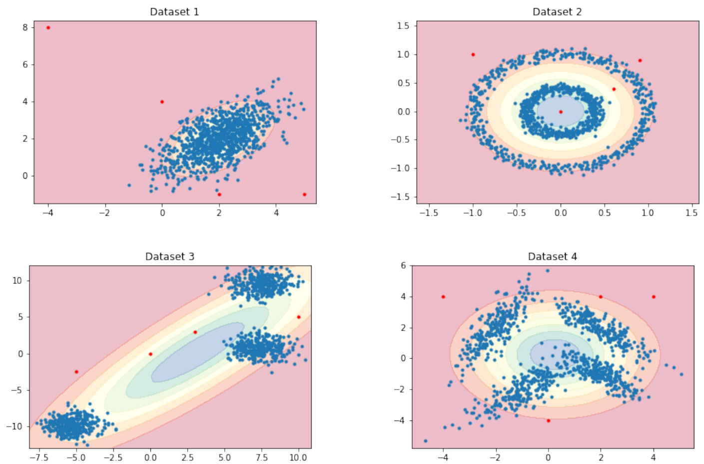
For example:



One of the outliers is made more outlier each time, increasing the covariance of the fitted distribution

Outlier Detection - Gaussian Distribution

Also.. Does not fit multimodal or oddly shaped distributions



Outlier Detection - Gaussian Mixture Model

Try using more than one Gaussian: **Gaussian Mixture Model**

$$\sum_{k}^{K} \pi_k p(x|\mu, C)$$

Estimate weighting π, mean μ and covariance C?
If we knew the weights, mean and covariance, we could calculate the probability
if we knew the probabilities, we could calculate the weights, mean and covariance
Expectation maximisation: generalisation of K Means

Outlier Detection - Gaussian Mixture Model

Algorithm 1: GMM

Data: X ($n \times p$ data), k Gaussians to use
Initialise π_k, μ_k and C_k ;
while *not converged* **do**
    for $x_i \in X$ **do**
        for $j \in 1, ..., k$ **do**
            responsibilities $r_{i,j} = p(x_i|\mu_j, C_j)$;
        end
    end
    for $j \in 1, ..., k$ **do**
        $N_j = \sum_{i=0}^{n} r_{i,j}$;
        $\pi_j = \frac{N_j}{N}$;
        $\mu_j = \frac{1}{N_j} \sum_{i=0}^{n} r_{i,j} x_i$;
        $C_j = \frac{1}{N_j} \sum_{i=0}^{n} r_{i,j}(x_i - \mu_j)(x_i - \mu_j)^T$;
    end
end

Outlier Detection - Gaussian Mixture Model

Initialisation:

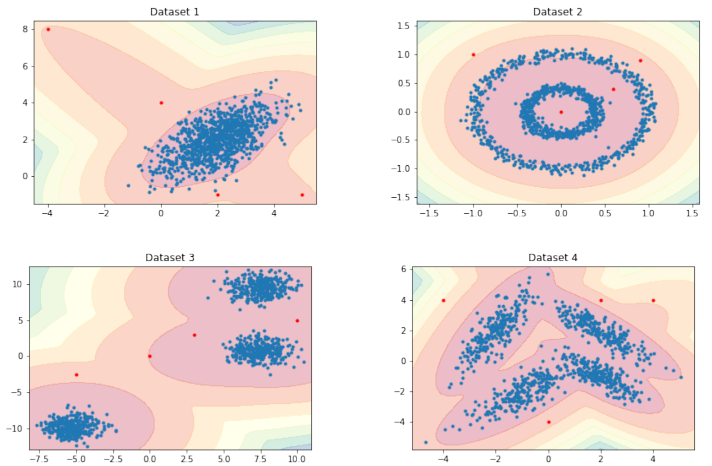
- randomly - can cause issues
- use K Means - works quite well

Convergence:

- Can check for an increase in the total probability
- $\sum_{i=0}^{k} \sum_{j=1}^{n} r_{i,j}$
- best to use logs

Outlier Detection - Gaussian Mixture Model

Test on datasets:



Works reasonably well for the three Gaussian distributions. Note sensitivity to outliers. What about the circular data set?

Outlier Detection - DBSCAN

DBSCAN - good for outlier detection as well as clustering
Recap: Density Based Spatial Clustering and Noise
Needs:

- maximum radius
- minimum number

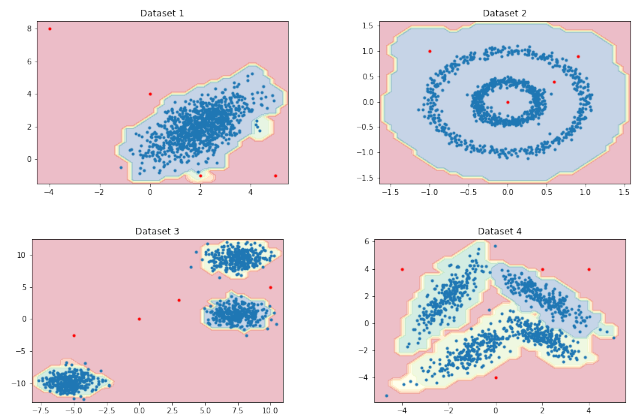
Max radius is the limit on which to look for neighbours
Min number is the lower limit on what can be in a cluster

Outlier Detection - DBSCAN

Algorithm 2: DBSCAN

Data: X, *eps*, *min_pts*
initialse *labels* list as zeros, *count* list, *core* list;
Find neighbours for each point, Find core points;
class = 1;
for *each core point p* **do**
  add neighbours(p) to queue;
  while *queue not empty* **do**
    neighbours = next(queue);
    for *q in neighbours* **do**
      set label(q = *class*);
      if *label(q) is 'core'* **then**
        add neighbours(q) to queue
      end
    end
  end
  class = *class* + 1
end
return labels;

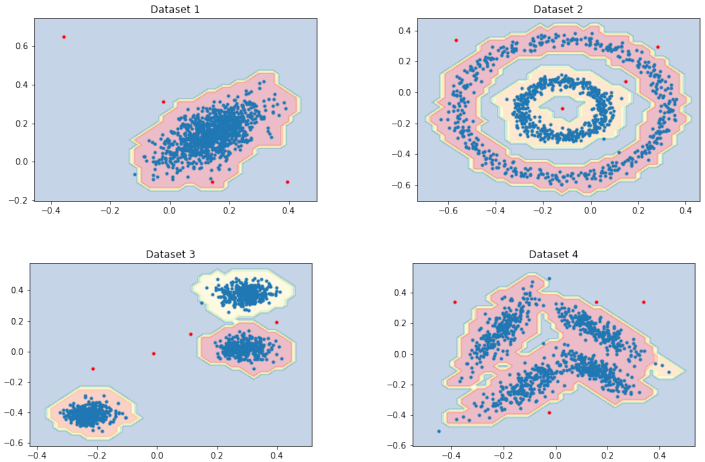
Outlier Detection - DBSCAN



What is going on here? works well (ish) on the Gaussian datasets, but not on the oddly shaped one..

Outlier Detection - DBSCAN

Normalisation! - and adjusting *eps*



Outlier Detection - Summary

Outlier detection is explored as a data mining problem:.
Extreme value statistics:

- to help tell the difference between an anomaly and an extreme member of a distribution

Gaussian Mixture Models:

- Models the system as a mixture of Gaussian distributions
- uses Expectation Maximisation to find parameters
- can be distorted by outliers

DBSCAN:

- Used for outlier detection
- Robust to outliers
- can have issues with parameters eps