Anomaly Detection

Anomaly Detection

- Goal: find objects that are different from most other objects
- The different objects are called **outliers** or **anomalies**

Applications

- **Fraud Detection**: purchasing behavior of identity thief could be different from that of original owner
- **Intrusion Detection**: attacks on computer systems have different patterns than normal use
- **Public Health**: occurrences of a disease after using a vaccine indicates problems with the vaccination program
- **Ecosystem Disturbances**: hurricanes, floods, droughts, fires, …
  - https://www.nature.com/news/satellite-alerts-track-deforestation-in-real-time-1.19427

Causes

- **Data from different classes:**
  - An object is anomalous because it belongs to a different class
  - Fraud cases, intrusions, outbreaks of diseases
- **Natural variation:**
  - The object is a rare occurrence from the distribution.
  - Exceptionally tall person, belongs to same class, but higher than average
- **Data measurement error**:
  - Recording a measurement incorrectly
  - Problem with the measuring device

Types

- **Global (point anomalies):**
  - If the object deviates significantly from the rest of the data objects

  e.g., spam emails

- **Contextual (conditional anomalies):**
  - If the object deviates significantly with respect to a specific context

  e.g., shopping behavior

- **Collective**:
  - If a subset of data objects deviates as a group from the rest of the data objects
  - Individual behavior does not deviate significantly from the normal range, but the combined anomaly indicated a bigger issue

  e.g., temperature sensors

Issues

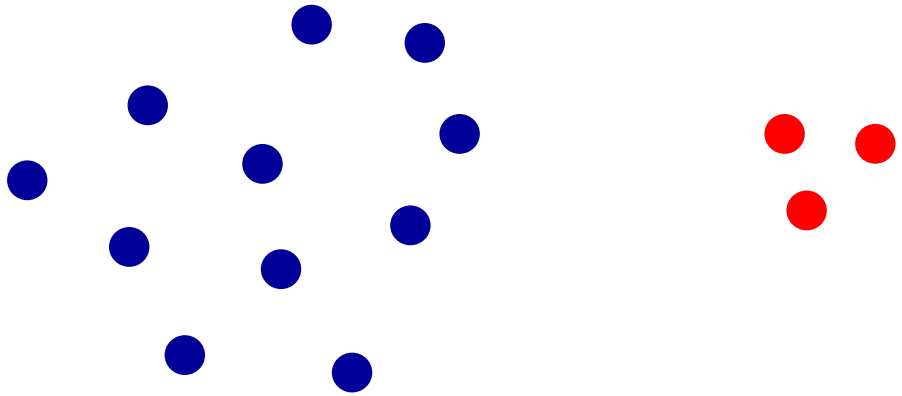
- What attributes should be used to define the anomaly:
  - A single attribute, a combination of attributes?
  - Normal values: a person who is 2 feet tall (a child), a person who weighs 220lbs
  - Anomaly: a person who is 2 feet tall who weighs 220lbs
  - Difficult with high dimensionality

- Global vs local perspective:
  - A person who is 6’5” is tall with respect to general population, but average with respect to basketball players

Issues

- Degree of anomaly:
  - Binary result: anomaly or not
  - Anomaly score: assessment of the extent to which the object is anomalous

- Finding one anomaly at a time vs many at once:
  - The presence of several anomalies may mask their presence



Issues

- Evaluation:
  - Anomalous class is much smaller than normal class
  - If labeled: measures: recall, precision, false positive are more appropriate than accuracy
  - If not labeled, judge in term of improvement to the model when anomalies are eliminated

- Efficiency:
  - The computational cost of different detection schemes

Issues

Concept Drift

- Normal behavior changes over time
  - Text mining: topic of conversation changes
  - Credit card: user may take up a new hobby
- Static or global model will trigger false alarms

Anomaly Detection Techniques

- **Statistical Approaches**
- **Proximity-based**
  - Anomalies are points far away from other points
  - Distance
  - Density
- **Clustering based**
  - Kmeans
- **Reconstruction Based (Dimension reduction based)**

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.

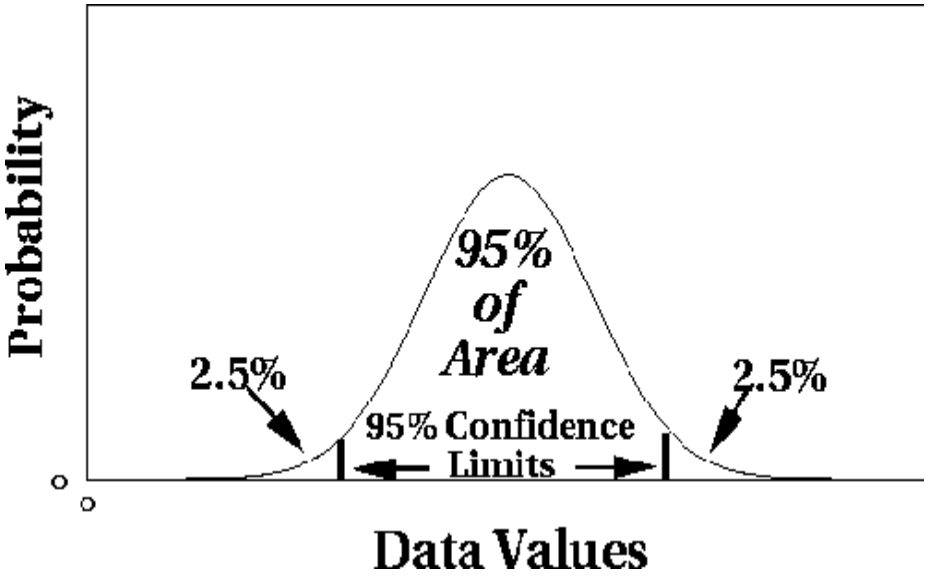
Usually assume a parametric model describing the distribution of the data (e.g., normal distribution)

Apply a statistical test that depends on

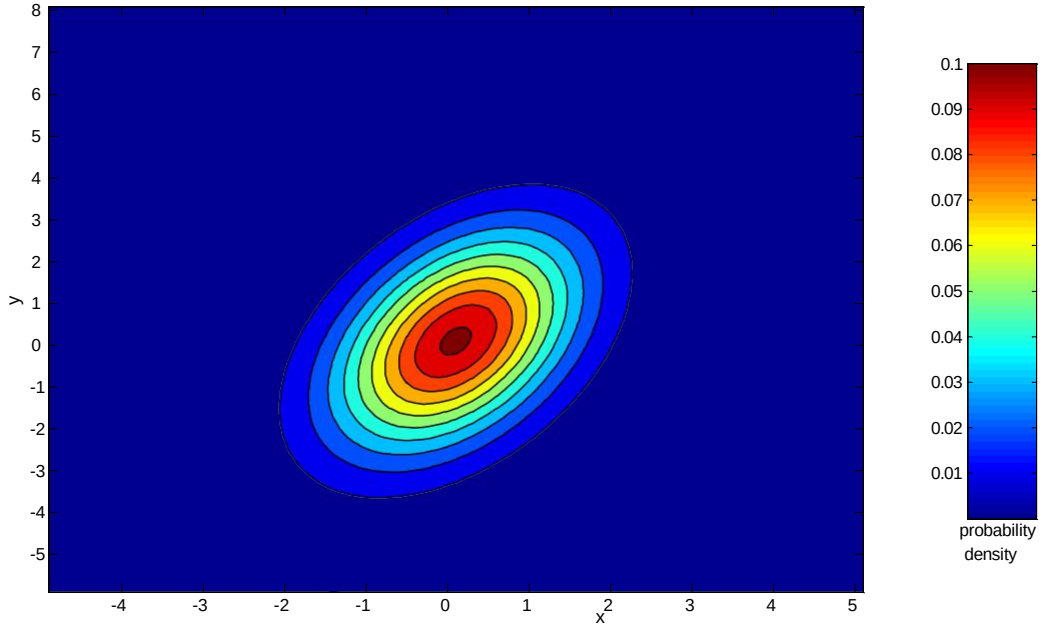
- Data distribution
- Parameters of distribution (e.g., mean, variance)
- Number of expected outliers (confidence limit)

Normal Distributions

One-dimensional Gaussian



Two-dimensional Gaussian



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:

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

- Reject H_0 if:

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

$\bar{X}$ is the sample mean; s is the standard deviation

α is the significance level

$t_{(\alpha / 2N, N-2)}$ denotes the upper critical value of the t-distribution with N-2 DOF and a significance level $\alpha / 2N$

Statistically-based – Likelihood Approach

Assume the data set D contains samples from a mixture of two probability distributions:

- M (majority distribution)
- A (anomalous distribution)

General Approach:

- Initially, assume all the data points belong to M
- Let $L_t(D)$ be the log likelihood of D at time t
- For each point x_t that belongs to M, move it to A
  - Let $L_{t+1}(D)$ be the new log likelihood.
  - Compute the difference, $\Delta = L_t(D) - L_{t+1}(D)$
  - If $\Delta > c$ (some threshold), then x_t is declared as an anomaly and moved permanently from M to A

Strengths/Weaknesses of 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

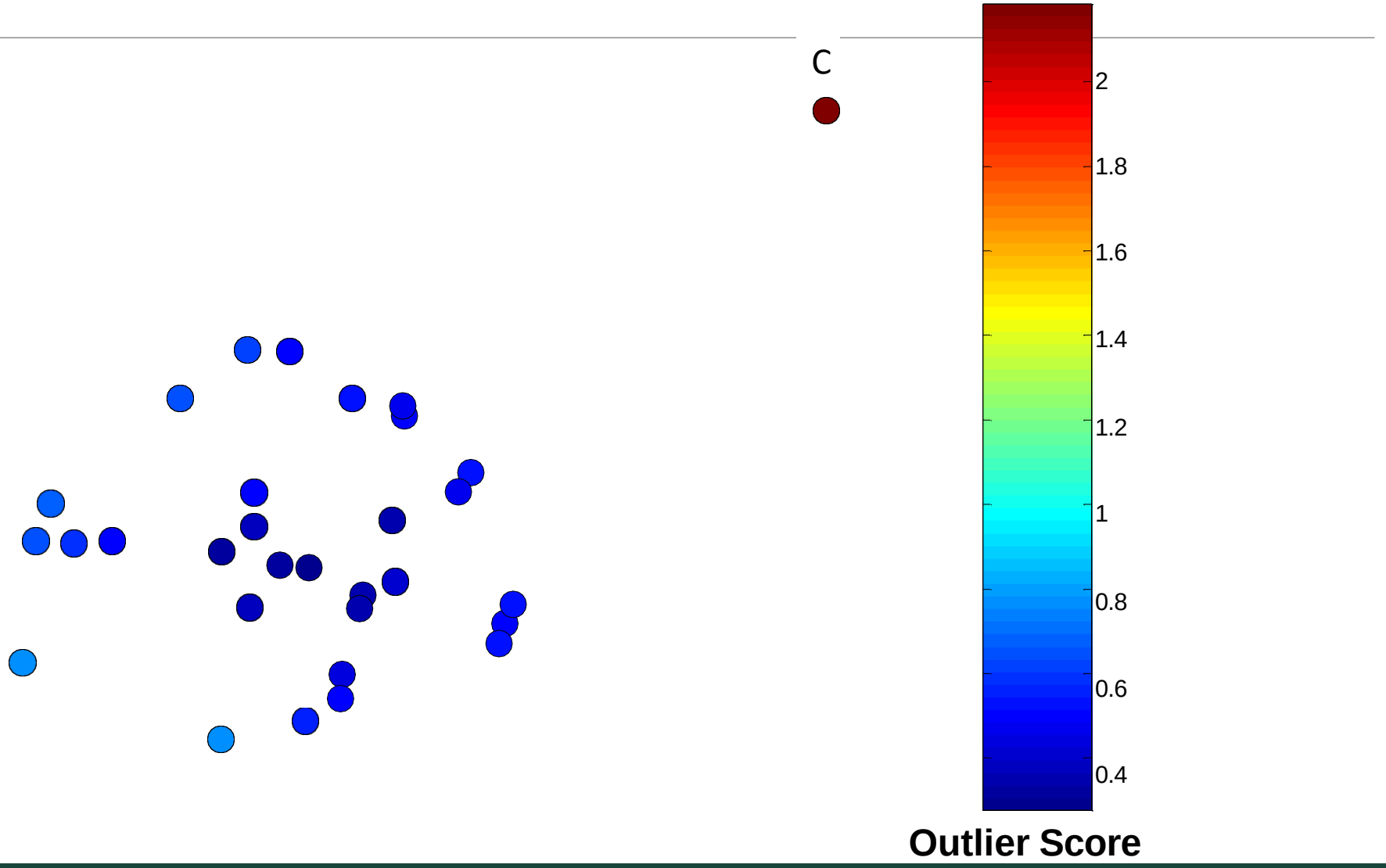
Anomalies can distort the parameters of the distribution

Proximity Based

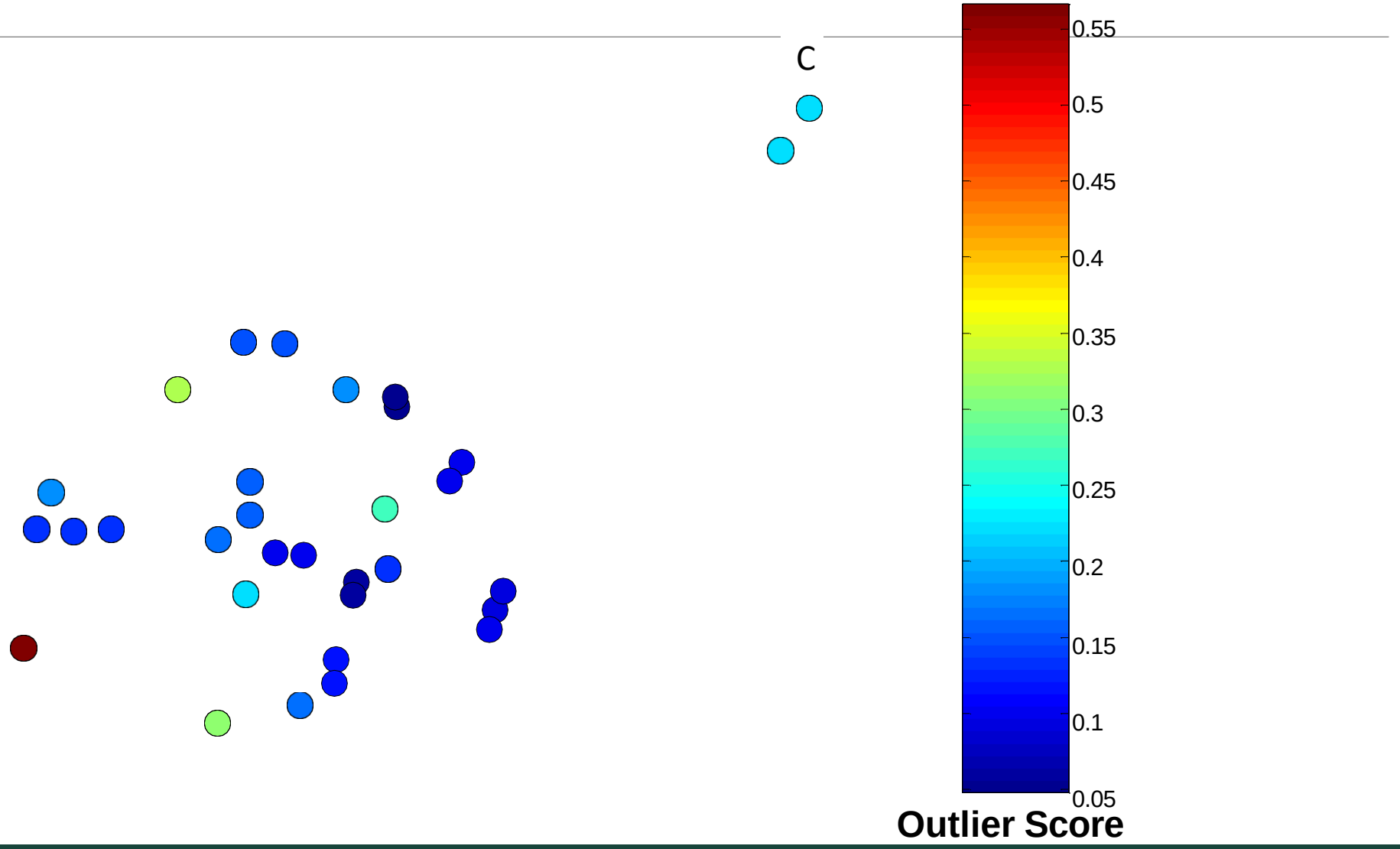
- An object is anomalous if it is distant from most points

Metric: Distance to its kth nearest neighbor gives the outlier score of an object

One Nearest Neighbor - One Outlier



One Nearest Neighbor - Two Outliers

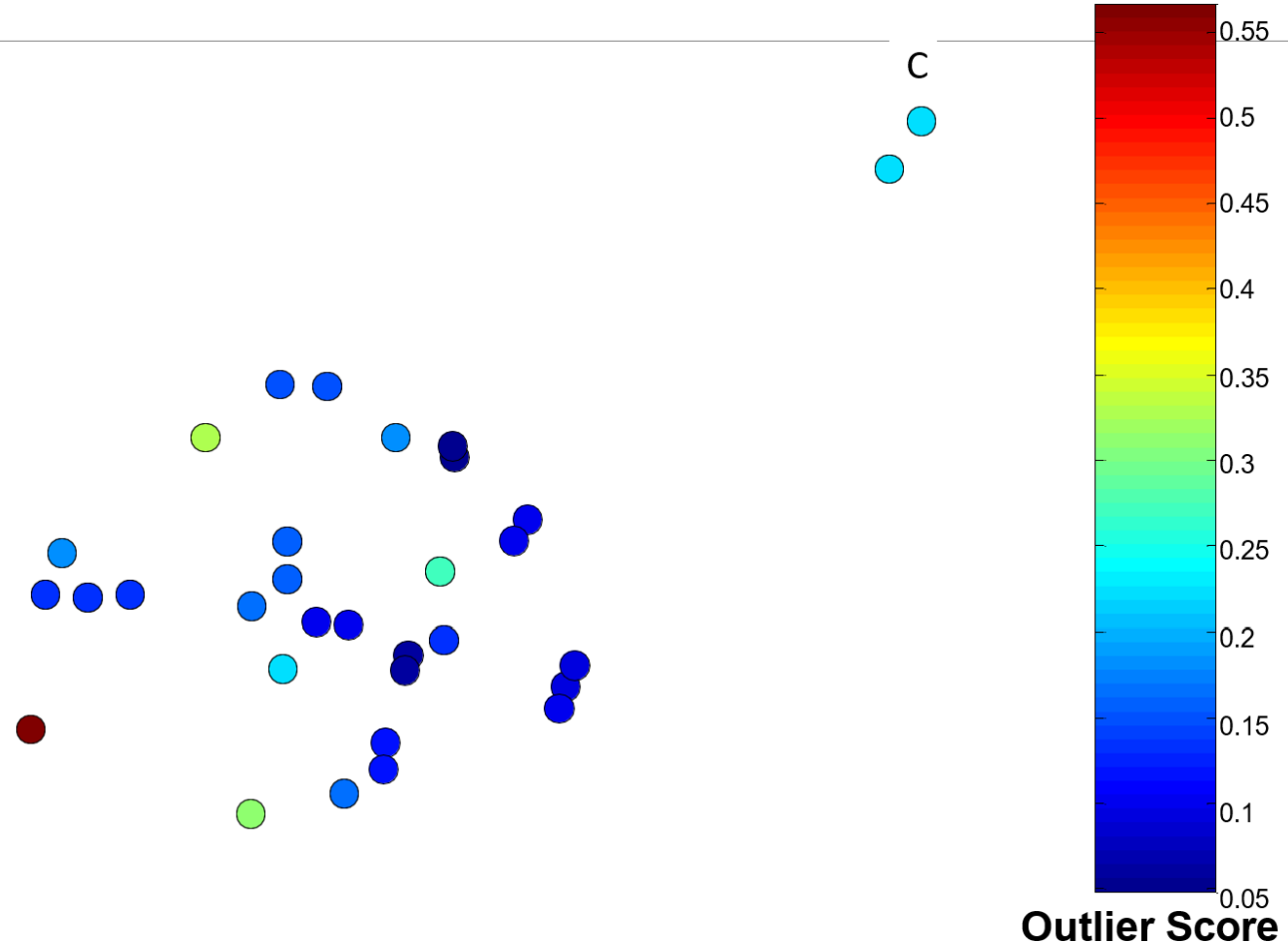


Choice of k

If k is too small for example k=1, then a small number of outliers close to each other can show a low anomaly score

Both C and its neighbor have a low outlier score

Points in the loose cluster have a higher score and one is identified as outlier

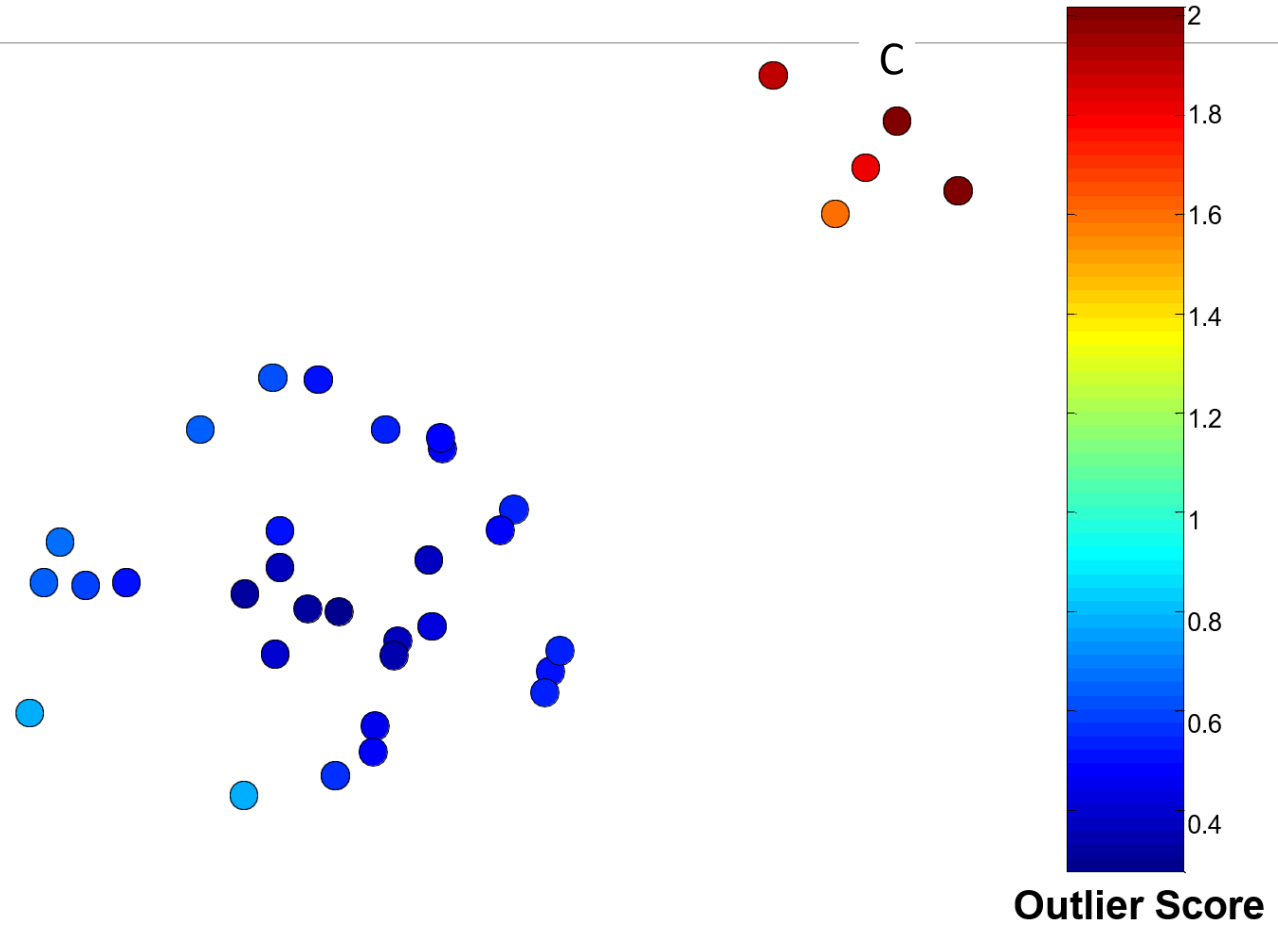


Outliers score based on 1-NN

Choice of k

If k is too large , then it is possible for all objects in a cluster that has fewer than k objects to become anomalies

A small cluster may become an outlier cluster

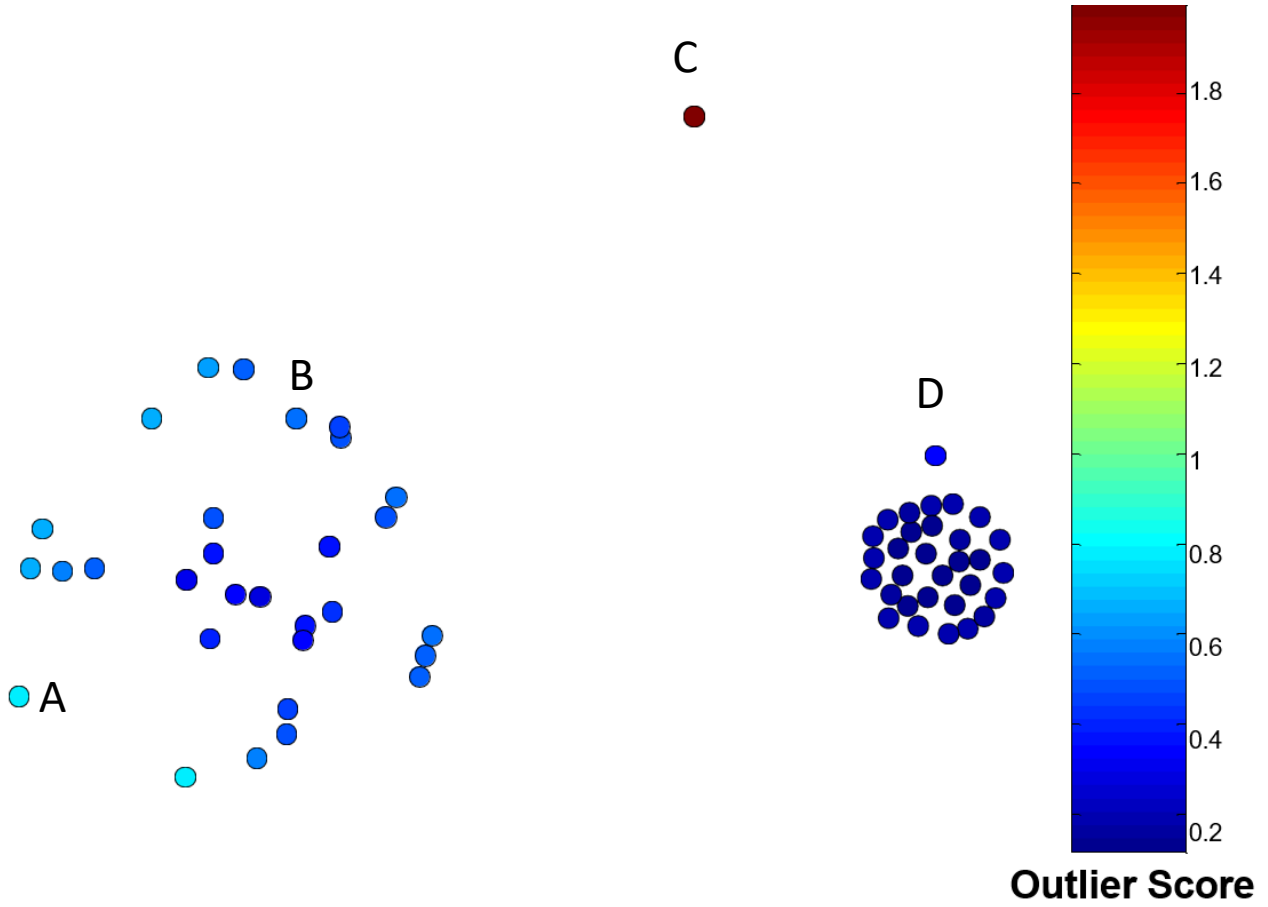


Outliers score based on 5-NN

Choice of k

Clusters with different densities:

- *C is correctly identified as an anomaly*
- *D is missed although it is an anomaly*
- *D has lower score than A*



Outliers score based on 5-NN

Issues

- Sensitive to choice of parameters
- Computation time: m^2 but could be improved
- Cannot handle regions of different densities because global thresholds are used instead of different thresholds for different regions

- Improvement: average of the distances to the first k-nearest neighbors
  - widely used as a reliable proximity-based anomaly score

Density-Based

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

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

Relative Density

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

Let $y_1, \ldots, y_k$ be the k nearest neighbors of $\boldsymbol{x}$

$$density(\boldsymbol{x}, k) = \frac{1}{dist(\boldsymbol{x}, k)} = \frac{1}{dist(\boldsymbol{x}, \boldsymbol{y}_k)}$$

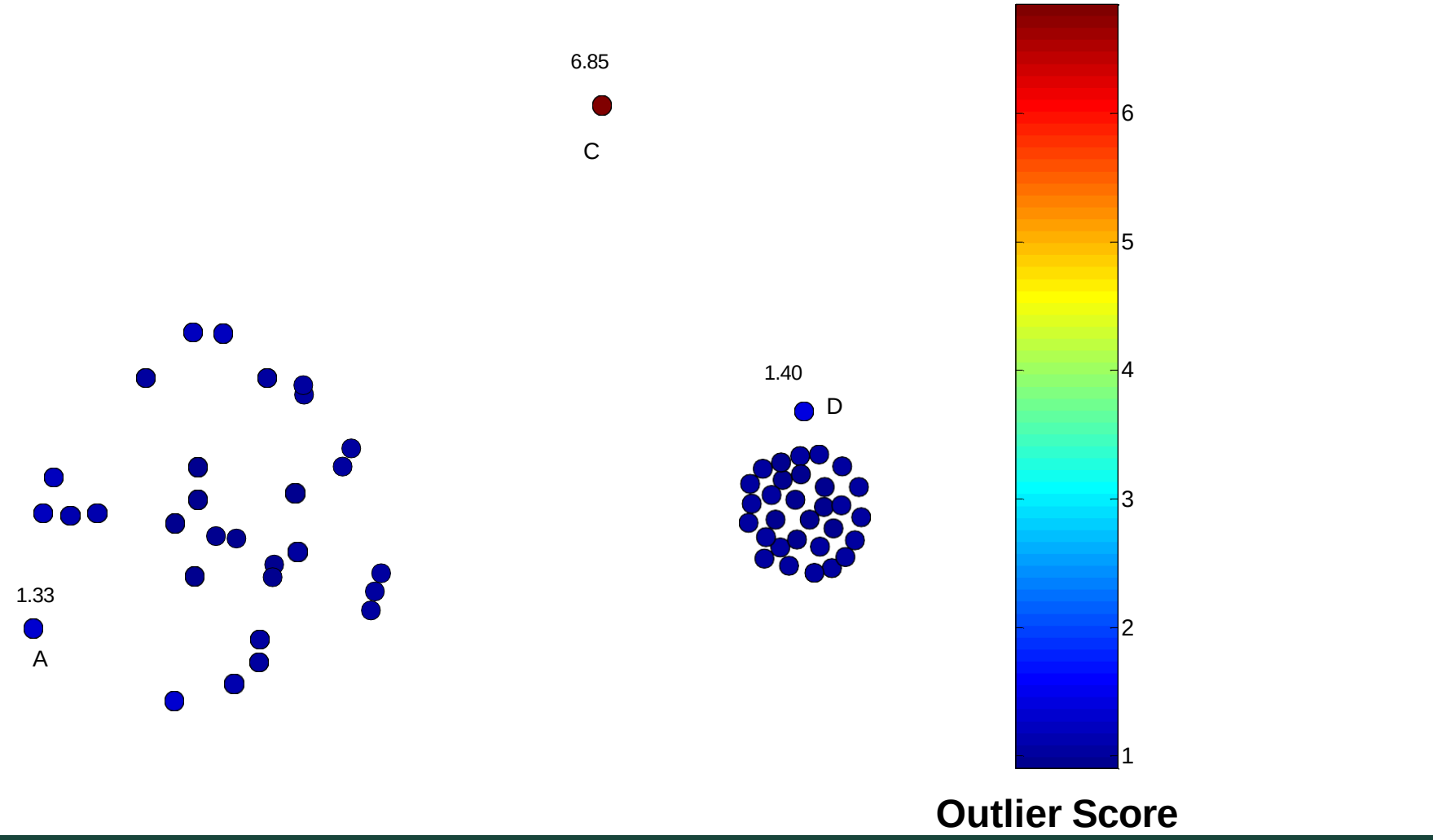
$$relative\ density(\boldsymbol{x}, k) = \frac{\sum_{i=1}^{k} density(\boldsymbol{y}_i, k)/k}{density(\boldsymbol{x}, k)}$$

$$\frac{\text{average density of neighbors}}{\text{density of } x}$$

$$\rightarrow \frac{dist(\boldsymbol{x}, k)}{\sum_{i=1}^{k} dist(\boldsymbol{y}_i, k)/k}$$

Can use average distance instead

Relative Density Outlier Scores

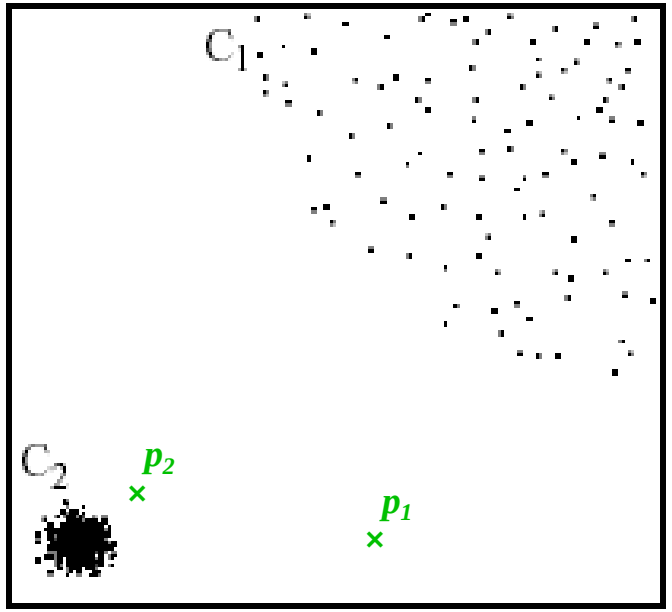


Relative Density-based: LOF approach

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



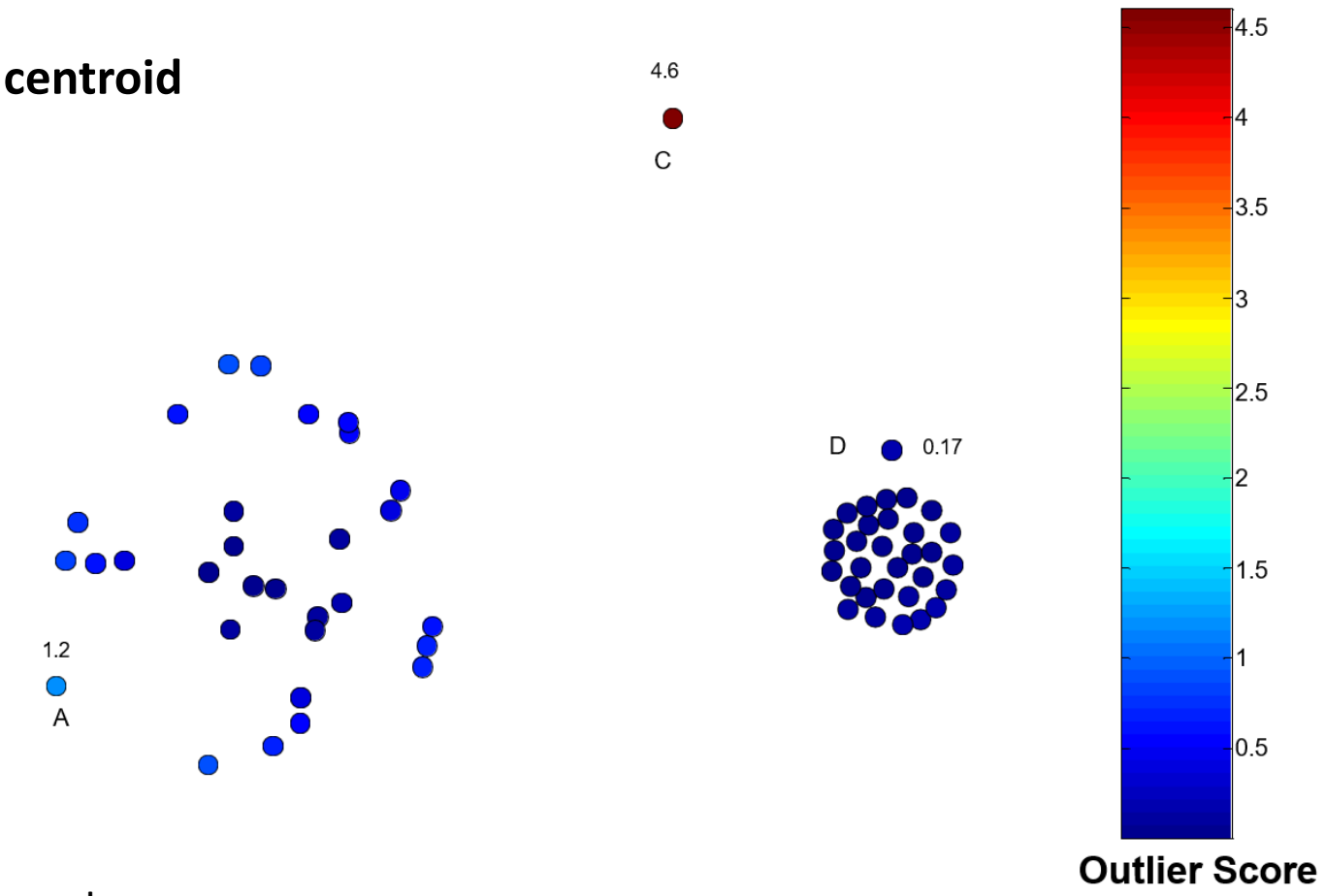
LOF approach find both p_1 and p_2 as outliers

Clustering Based

- From clustering perspective, a way to describe an anomaly is an instance that cannot be explained well by any of the normal clusters
- So the basic approach is to:
  - Cluster all objects
  - Assess the degree to which an object belongs to a cluster
- How to assess:
  - For prototype-based clusters, an object is an outlier if it is not close enough to a cluster center
    - Outliers can impact the clustering produced
  - For density based: if density is too low
    - Can’t distinguish between noise and outliers
  - For connectivity based: if connectivity is too low

K-Means example

Assess using the point's distance to its closest **centroid** as their anomaly score



This approach has problem when clusters has different densities. For example, D is not considered an outlier.

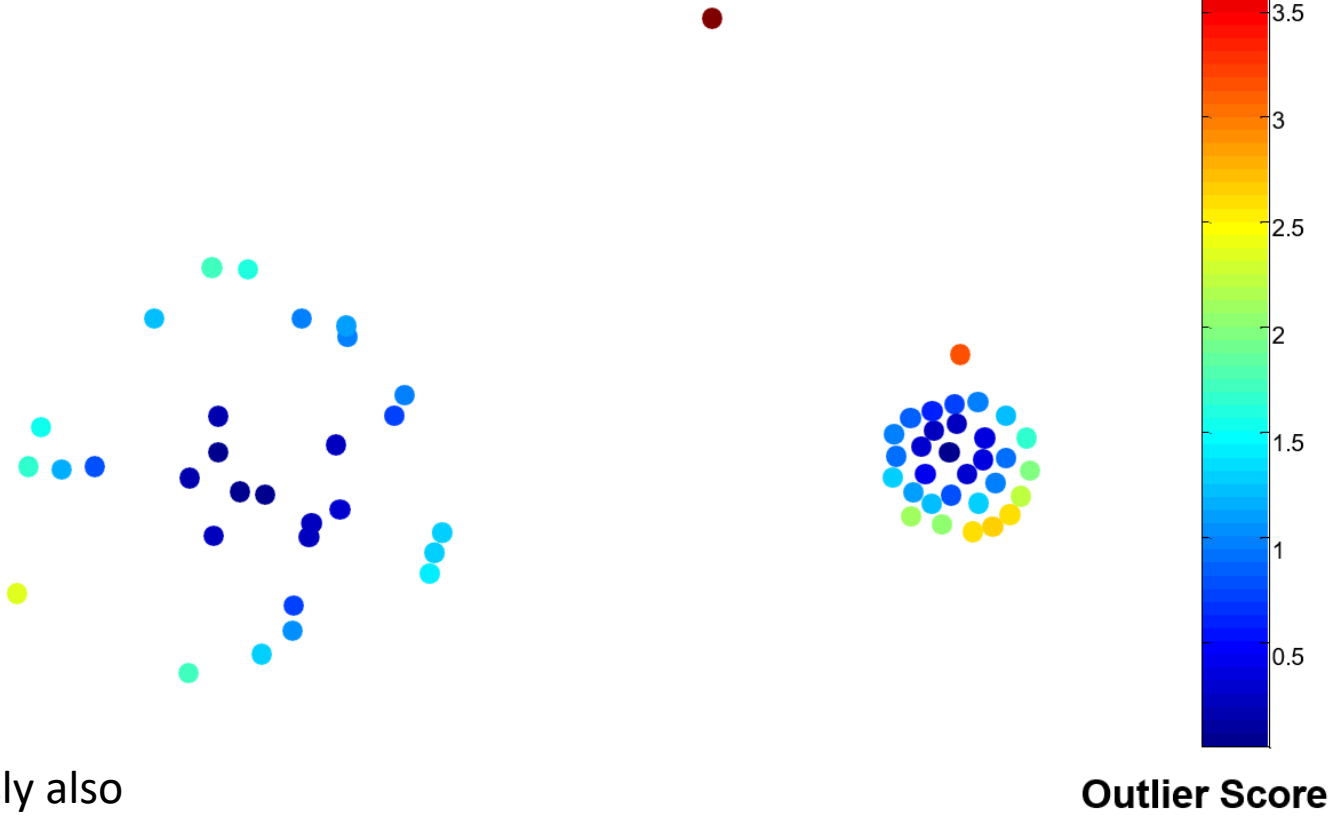
K-Means example

Assess using relative distance from closest centroid

Relative distance of a point p to a centroid c:

$$d(p, c)/D_c$$

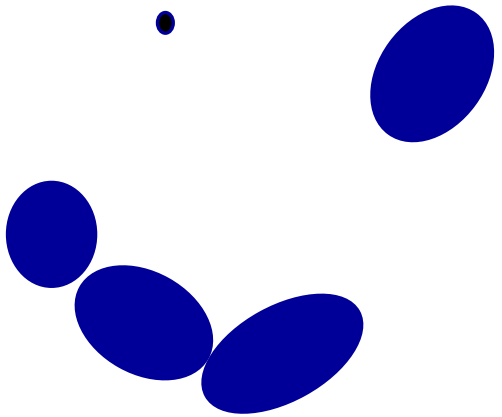
where D_c is the sum of the distances of all points in the cluster of c to c



The points that has been identified as outliers previously also show up as anomalies here.

Issues

- Do outliers affect initial clustering?
  - Yes because clustering are often sensitive to the presence of outliers in the data
- Solution:
  - Cluster points
  - Remove outliers
  - Cluster again



- What number of clusters to use for outlier detection?
  - Option 1: Repeat for different k
  - Option 2: Find a large number of small clusters
    - If an object does not belong to any group, then it is likely it is a true outlier
    - A small group of outliers could be identified as a normal cluster

Dimension Reduction based

Reconstruction-based techniques

- assume that the normal class resides in a space of lower dimensionality than the original space of attributes
- there exists patterns in the distribution of the normal class that can be captured using lower-dimensional representation

Such lower-dimensional representation is obtained using dimensionality reduction techniques

Dimension Reduction based

Transform data using Principal Component Analysis (PCA)

$$X^\top X w = S w = \Lambda \mathrm{w}$$

- X is the data matrix, with column means = 0
- S is the covariance matrix of X, $S = X^\top X$
- w is the eigenvalues, Λ is a diagonal matrix of eigenvalues

Project original data X into a reduced dimension using the top k principal components

$$\hat{X} = Xw$$

Reconstruct the original data from top k principal components

$$X_{new} = \hat{X} w^\top$$

Points with large reconstruction error are anomalous

$$\|X - X_{new}\|$$

Dimension Reduction based

Use Singular Value Decomposition (SVD) to work with original data matrix

$$X = U\Sigma V^{\top}$$

- U is the left singular vectors (columns of U)
- V is the right singular vectors (columns of V)
- U and V are orthogonal ($U^{-1} = U^{\top},\ V^{-1} = V^{\top}$)

$$UU^{\top} = I \qquad VV^{\top} = I$$

- I is the identity matrix

Transform the data

- $\hat{X} = XV$

Reconstruct the data

- $X_{new} = \hat{X}V^{\top}$

Points with large decomposition error are anomalies

Relationship between PCA and SVD

PCA on covariance matrix of X is equivalent to SVD on X when mean subtracted from columns

Many implementation of PCA use SVD

SVD can be applied to sparse data

- Uses less memory

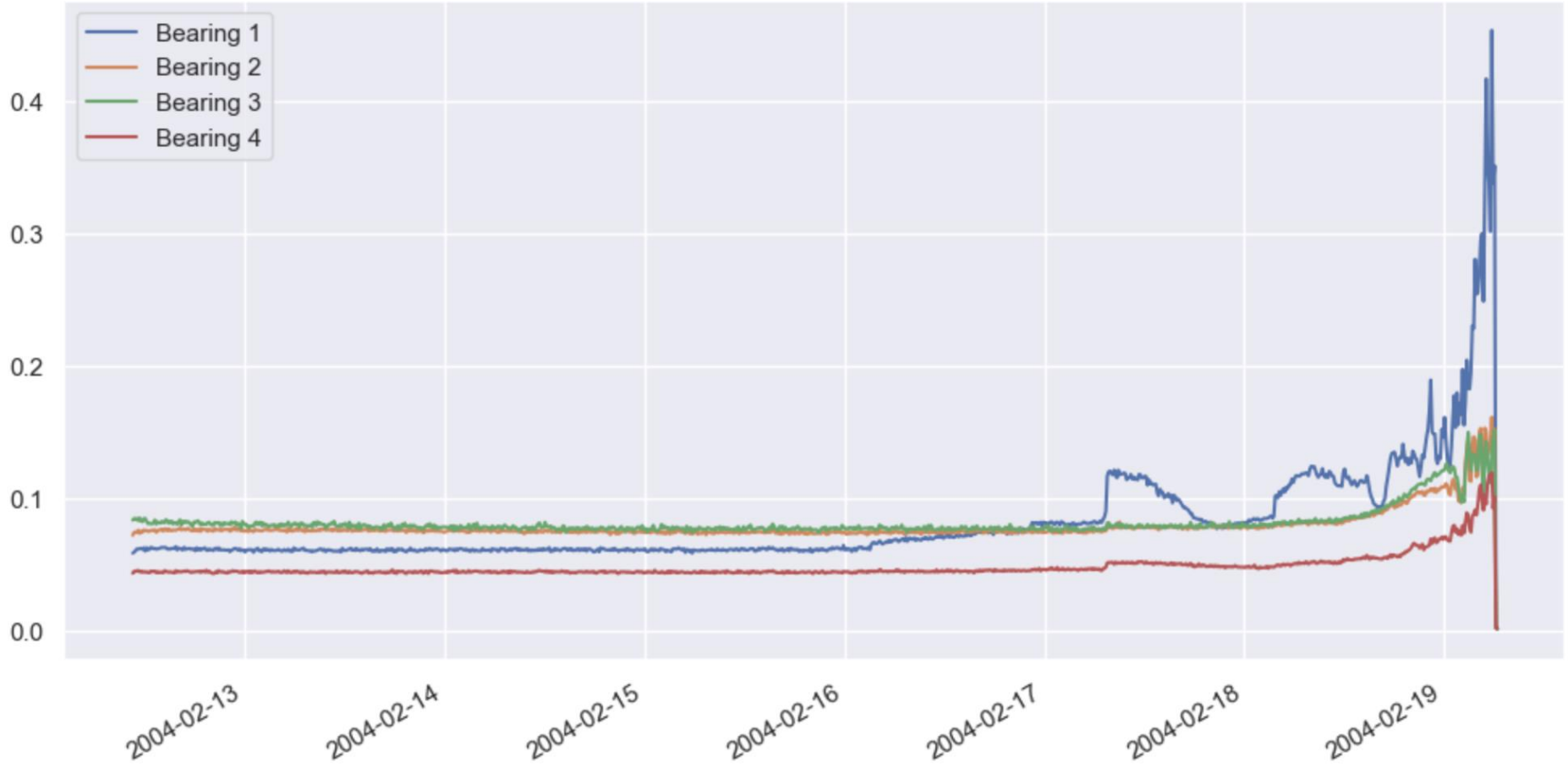
Class Exercise

PCA based Anomaly Detection

```
# Load Data

import pandas as pd

df = pd.read_csv('./bearings.csv', index_col=0, parse_dates=[0])


df.head()

# Each sample contains vibration data for four bearings
# the samples were taken 10 minutes apart

import matplotlib.pyplot as plt
import seaborn as sns
sns.set()

# this dataset contains 984 samples.

df.plot(figsize=(12,6))
```

Results

about four days into the test, vibrations in bearing #1 began increasing.
They spiked a day later,
and about two days after that, bearing #1 suffered a faliure.
the goal is to recognize increased vibration as a sign of impending failure



Class Exercise

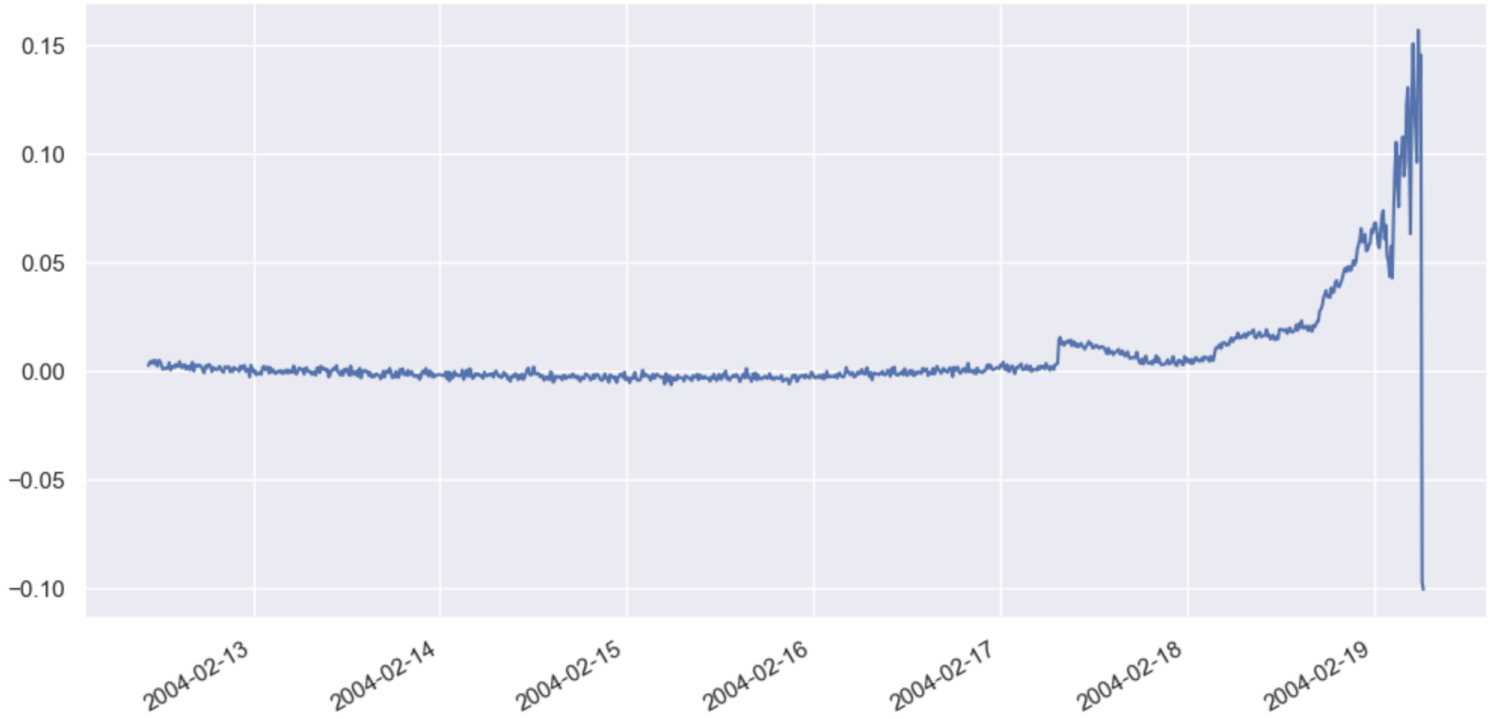
PCA based Anomaly Detection

```
# Fit the model
from sklearn.decomposition import PCA
x_train = df['2004-02-12 10:32:39':'2004-02-13 23:42:39']
x_test = df['2004-02-13 23:52:39':]
```

Class Exercise

PCA based Anomaly Detection

```
df_pca = pd.concat([x_train_pca, x_test_pca])
df_pca.plot(figsize=(12,6))
plt.legend().remove()
```



Class Exercise

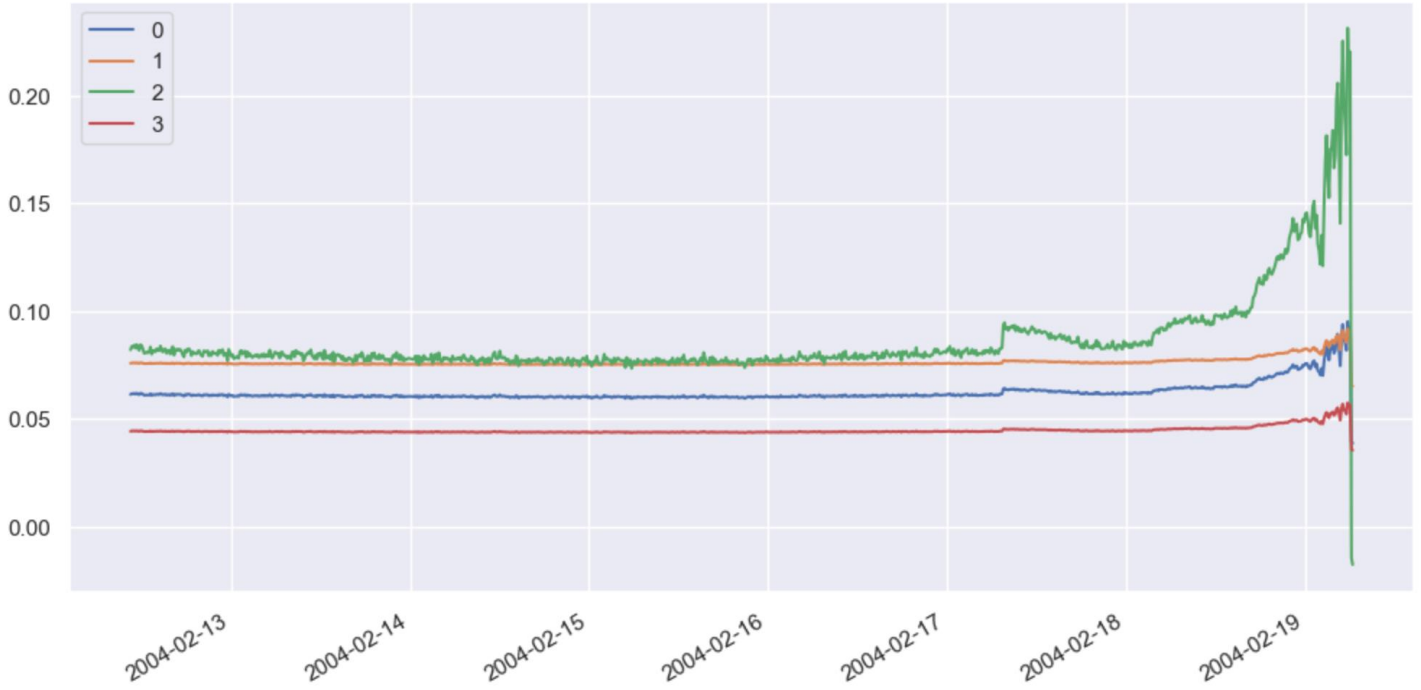
PCA based Anomaly Detection

now invert the PCA transform and plot the restored dataset
df_restored = pd.DataFrame(pca.inverse_transform(df_pca), index=df_pca.index)
df_restored.plot(figsize=(12,6))

Class Exercise

PCA based Anomaly Detection

now invert the PCA transform and plot the restored dataset
df_restored = pd.DataFrame(pca.inverse_transform(df_pca), index=df_pca.index)
df_restored.plot(figsize=(12,6))



Class Exercise

```
df.head()
```

	Bearing 1	Bearing 2	Bearing 3	Bearing 4
2004-02-12 10:32:39	0.058333	0.071832	0.083242	0.043067
2004-02-12 10:42:39	0.058995	0.074006	0.084435	0.044541
2004-02-12 10:52:39	0.060236	0.074227	0.083926	0.044443
2004-02-12 11:02:39	0.061455	0.073844	0.084457	0.045081
2004-02-12 11:12:39	0.061361	0.075609	0.082837	0.045118

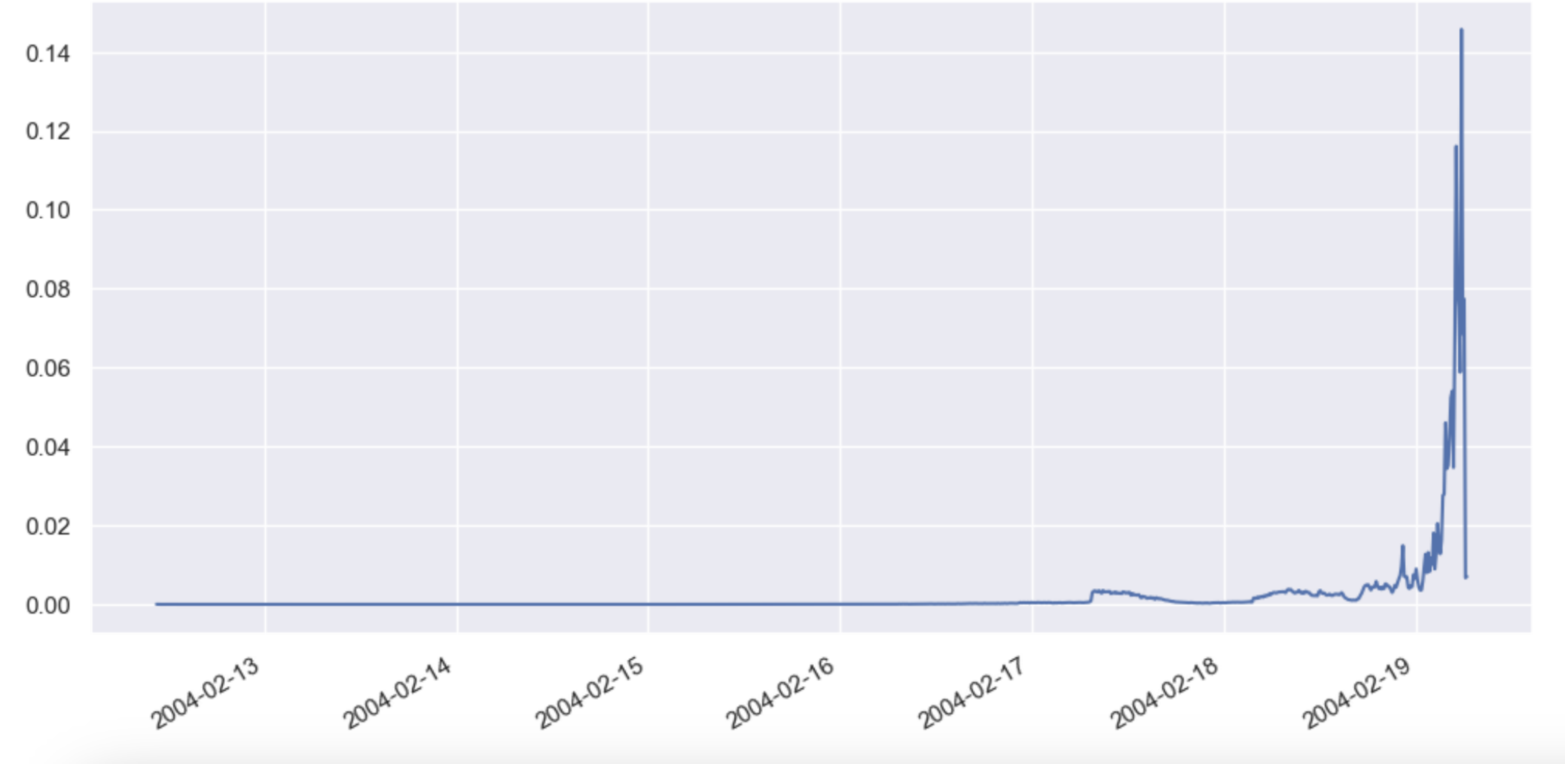
```
df_restored.head()
```

	0	1	2	3
2004-02-12 10:32:39	0.061336	0.075812	0.082033	0.044257
2004-02-12 10:42:39	0.061697	0.075981	0.083626	0.044397
2004-02-12 10:52:39	0.061652	0.075960	0.083426	0.044380
2004-02-12 11:02:39	0.061826	0.076042	0.084195	0.044447
2004-02-12 11:12:39	0.061519	0.075898	0.082840	0.044328

Class Exercise

```
# let's define a function that computs the loss in a range of samples

import numpy as np


def get_anomaly_scores(df_original, df_restored):

?

    loss = pd.Series(data=loss, index=df_original.index)

    return loss



scores = get_anomaly_scores(df, df_restored)
scores.plot(figsize=(12,6))
```

Results


0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
2004-02-13
2004-02-14
2004-02-15
2004-02-16
2004-02-17
2004-02-18
2004-02-19

Class Exercise

```
# detect anomaly
def is_anomaly(data, pca, threshold):
    pca_data = pca.transform(data)
    restored_data = pca.inverse_transform(pca_data)
    loss = np.sum((data - restored_data)**2)
? return
```

```
x = [df.loc['2004-2-16 22:52:39']]
is_anomaly(x, pca, 0.002)
```

False

```
x = [df.loc['2004-2-18 22:52:39']]
is_anomaly(x, pca, 0.002)
```

True

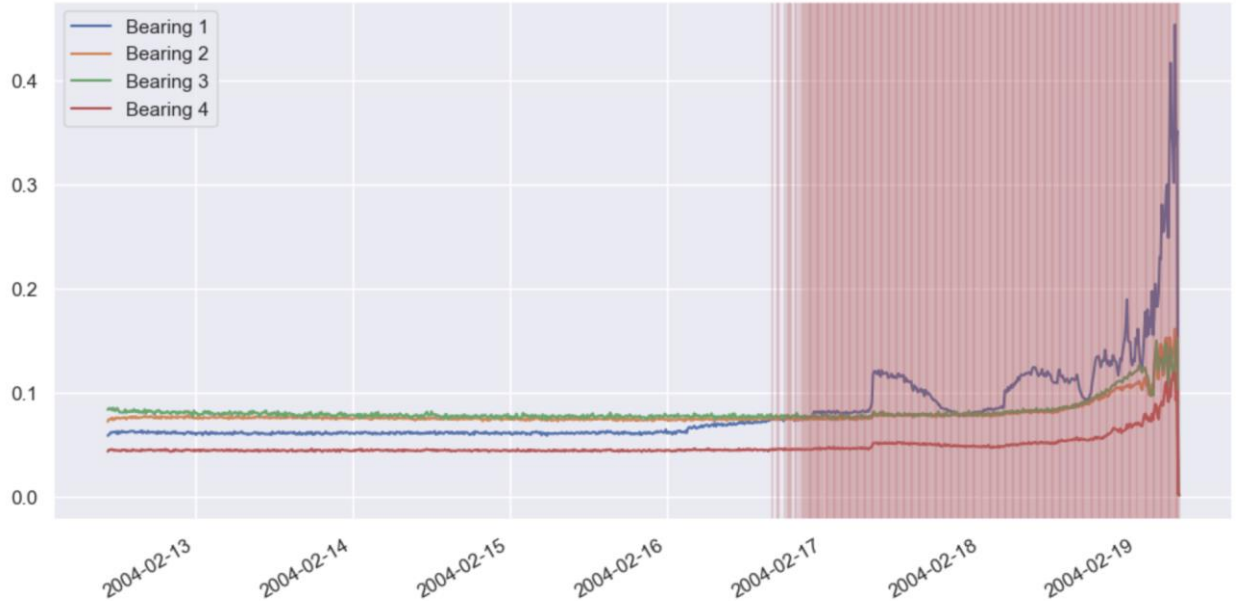
Class Exercise

```
df.plot(figsize=(12,6))

for index, row in df.iterrows():

    if is_anomaly([row], pca, 0.002):

        plt.axvline(row.name, color='r', alpha=0.2)
```

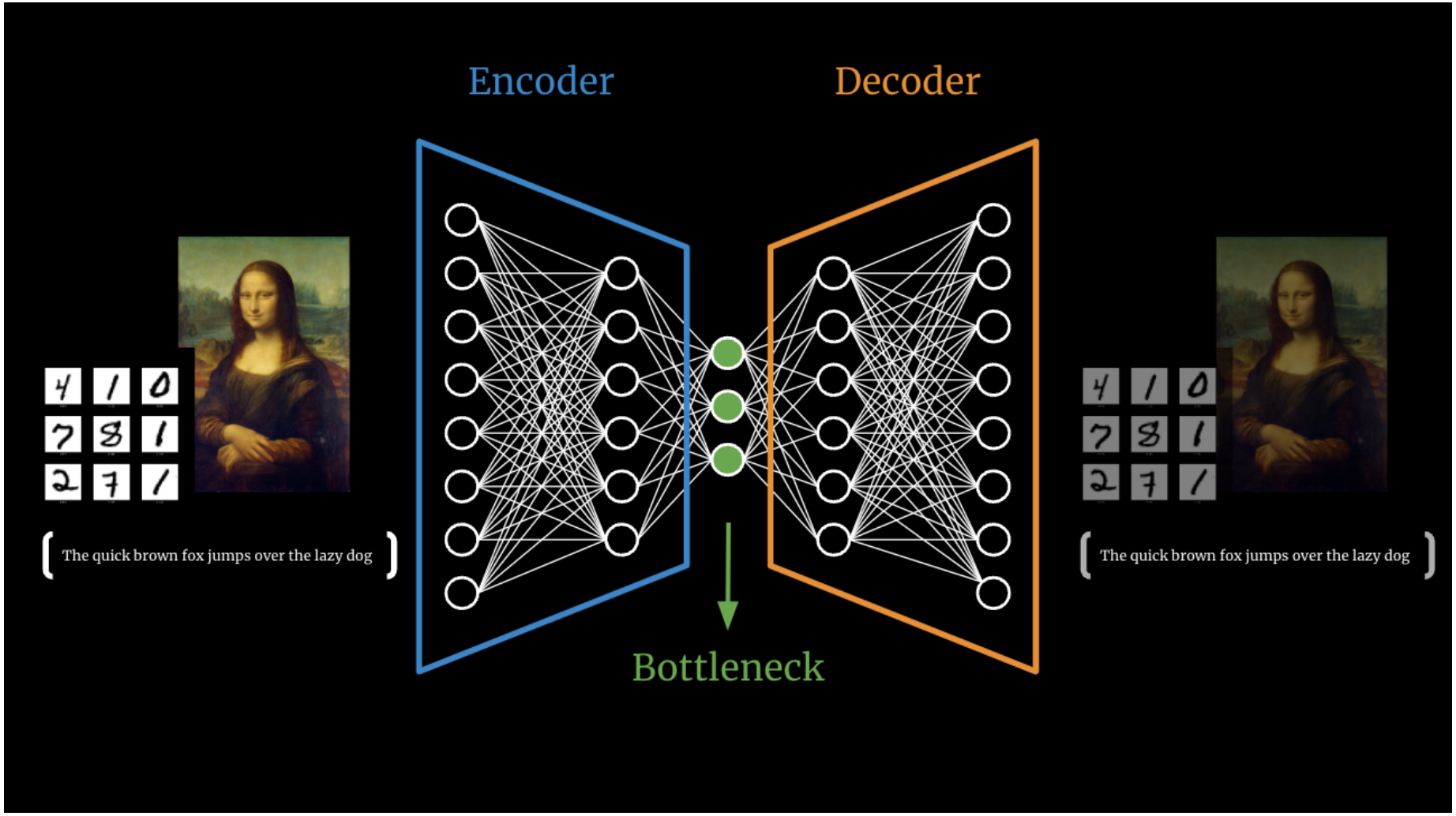
```
# lower threshold
df.plot(figsize=(12,6))
for index, row in df.iterrows():
    if is_anomaly([row], pca, 0.0002):
        plt.axvline(row.name, color='r', alpha=0.2)
```

Results


Bearing 1
Bearing 2
Bearing 3
Bearing 4
0.4
0.3
0.2
0.1
0.0
2004-02-13
2004-02-14
2004-02-15
2004-02-16
2004-02-17
2004-02-18
2004-02-19


Bearing 1
Bearing 2
Bearing 3
Bearing 4
0.4
0.3
0.2
0.1
0.0
2004-02-13
2004-02-14
2004-02-15
2004-02-16
2004-02-17
2004-02-18
2004-02-19

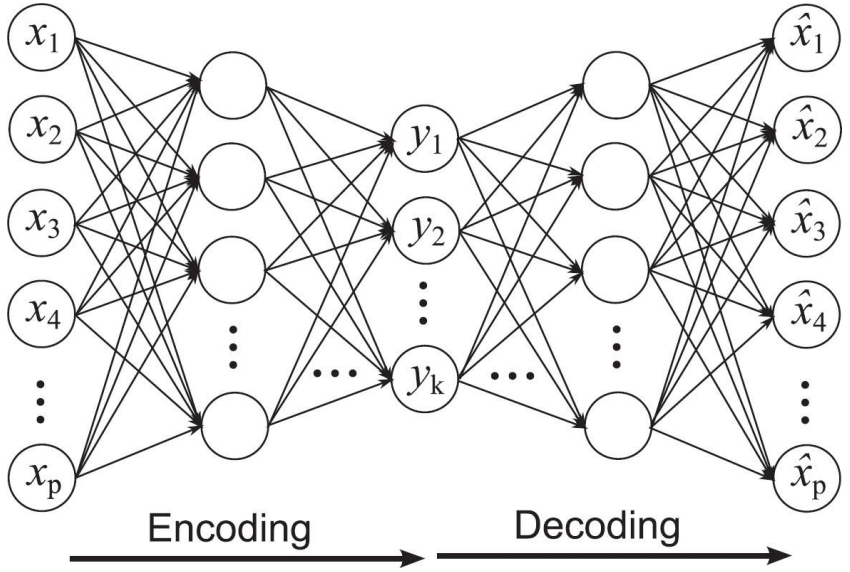
Deep Learning based Reconstruction



Basic Architecture of an Autoencoder

An autoencoder is a multi-layer neural network

The number of input and output neurons is equal to the number of original attributes.



One Class SVM

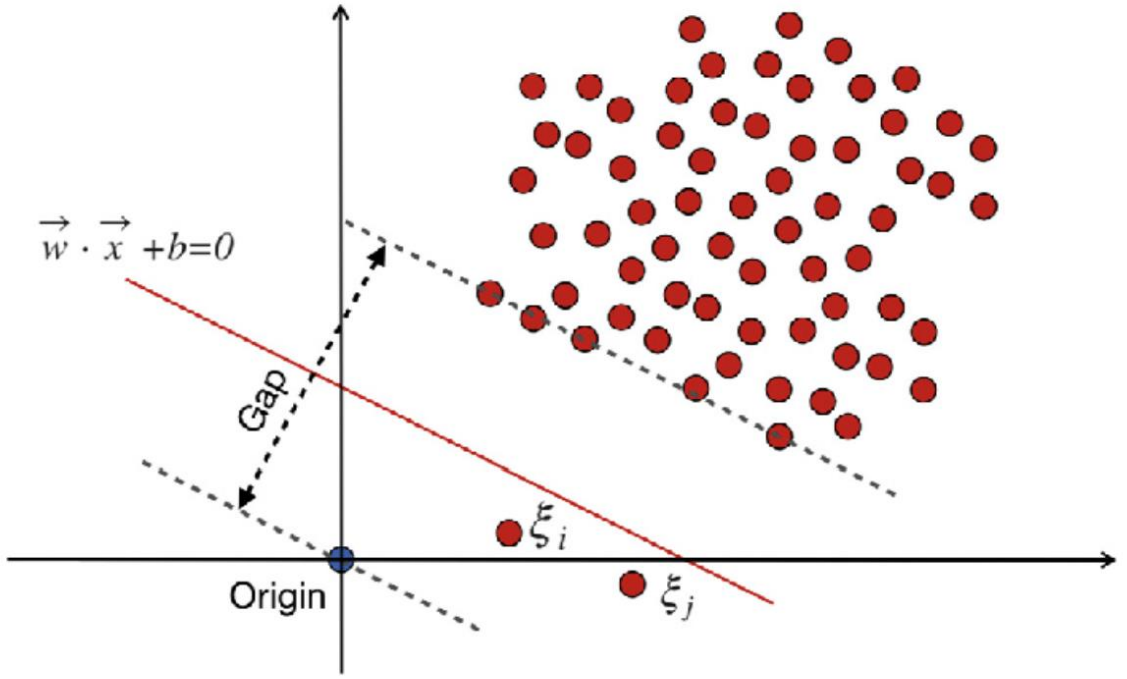
Uses an SVM approach to classify normal objects

Uses the given data to construct such a model

This data may contain outliers

But the data does not contain class labels

How to build a classifier given one class?



How Does One-Class SVM Work?

Uses the “origin” trick

Use a Gaussian kernel

$$\kappa(\mathbf{x}, \mathbf{y}) = \exp(-\frac{||\mathbf{x} - \mathbf{y}||^2}{2\sigma^2})$$

- Every point mapped to a unit hypersphere

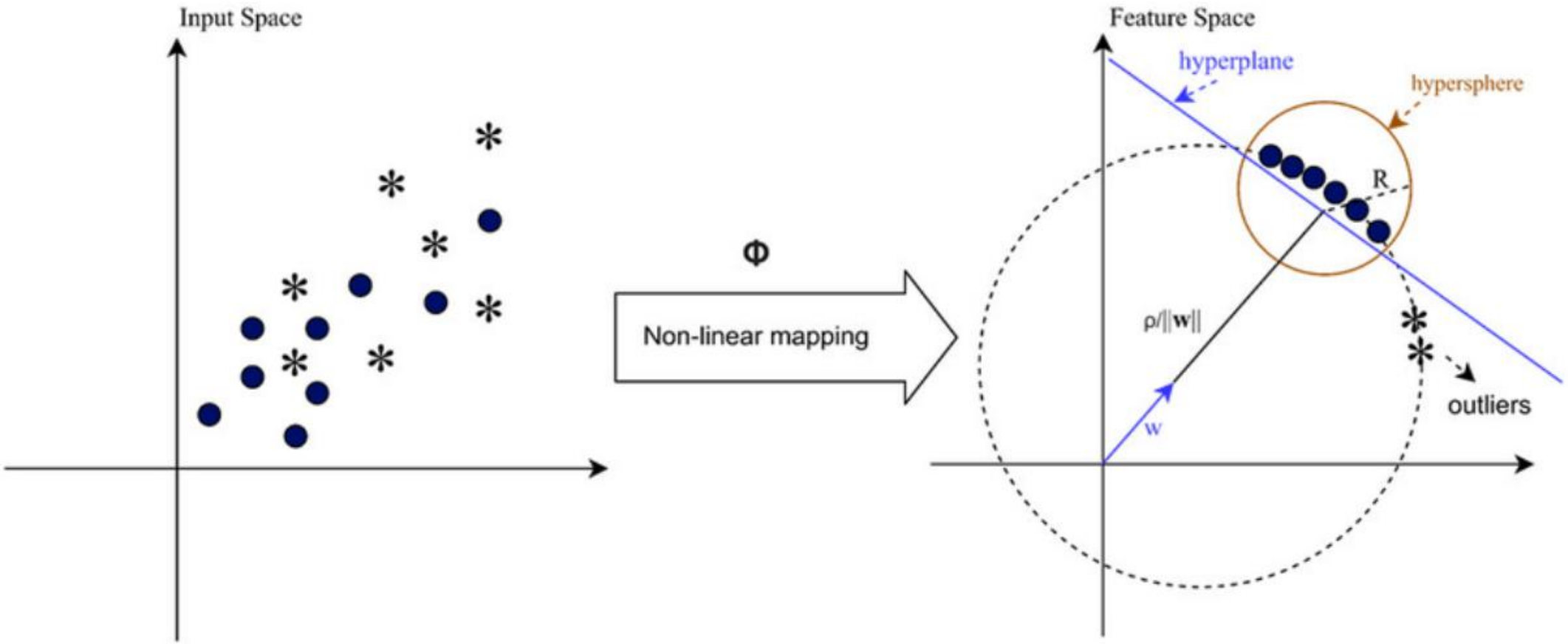
$$\kappa(\mathbf{x}, \mathbf{x}) = \langle \phi(\mathbf{x}), \phi(\mathbf{x}) \rangle = ||\phi(\mathbf{x})||^2 = 1$$

- Every point in the same orthant (quadrant)

$$\kappa(\mathbf{x}, \mathbf{y}) = \langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle \geq 0$$

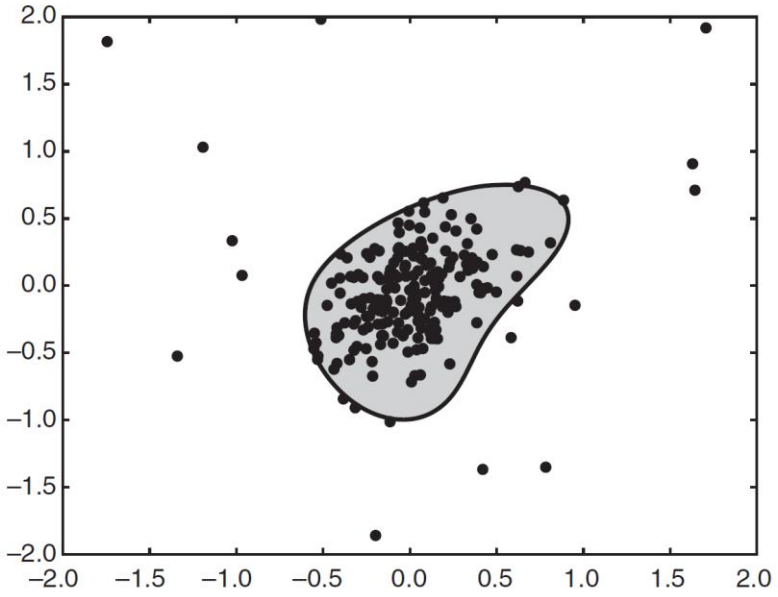
Aim to maximize the distance of the separating plane from the origin

How Does One-Class SVM Work?

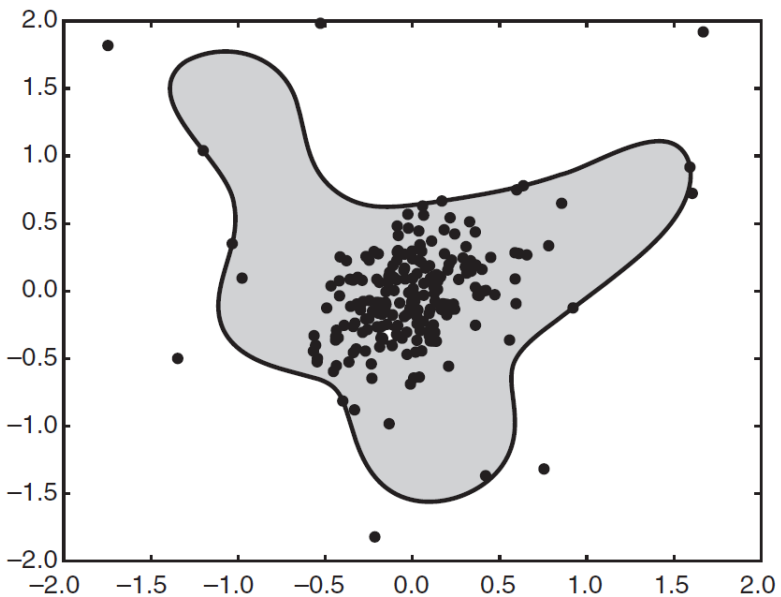


Finding Outliers with a One-Class SVM

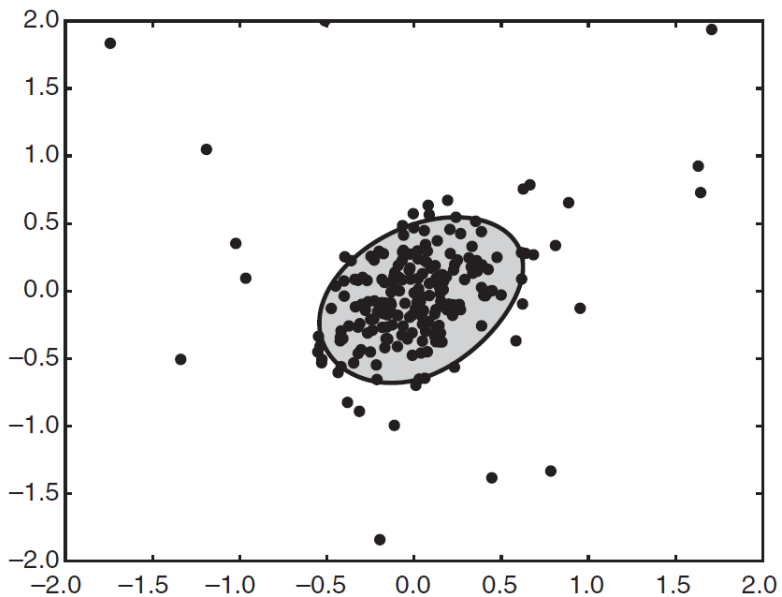
Decision boundary with $\nu = 0.1$



Decision boundary with $\nu = 0.05$ and $\nu = 0.2$



(a) $\nu = 0.05$.



(b) $\nu = 0.2$.

Strengths and Weaknesses

Strong theoretical foundation

Choice of ν is difficult

Computationally expensive

Sklearn for Anomaly Detection

- One class SVM

  sklearn.svm.OneClassSVM

  Alternative: **sklearn.linear_model.SGDOneClassSVM**

  * decision boundaries from these two functions are very similar. The SGD version is more computationally efficient.

- Local Outlier Factor (LOF)

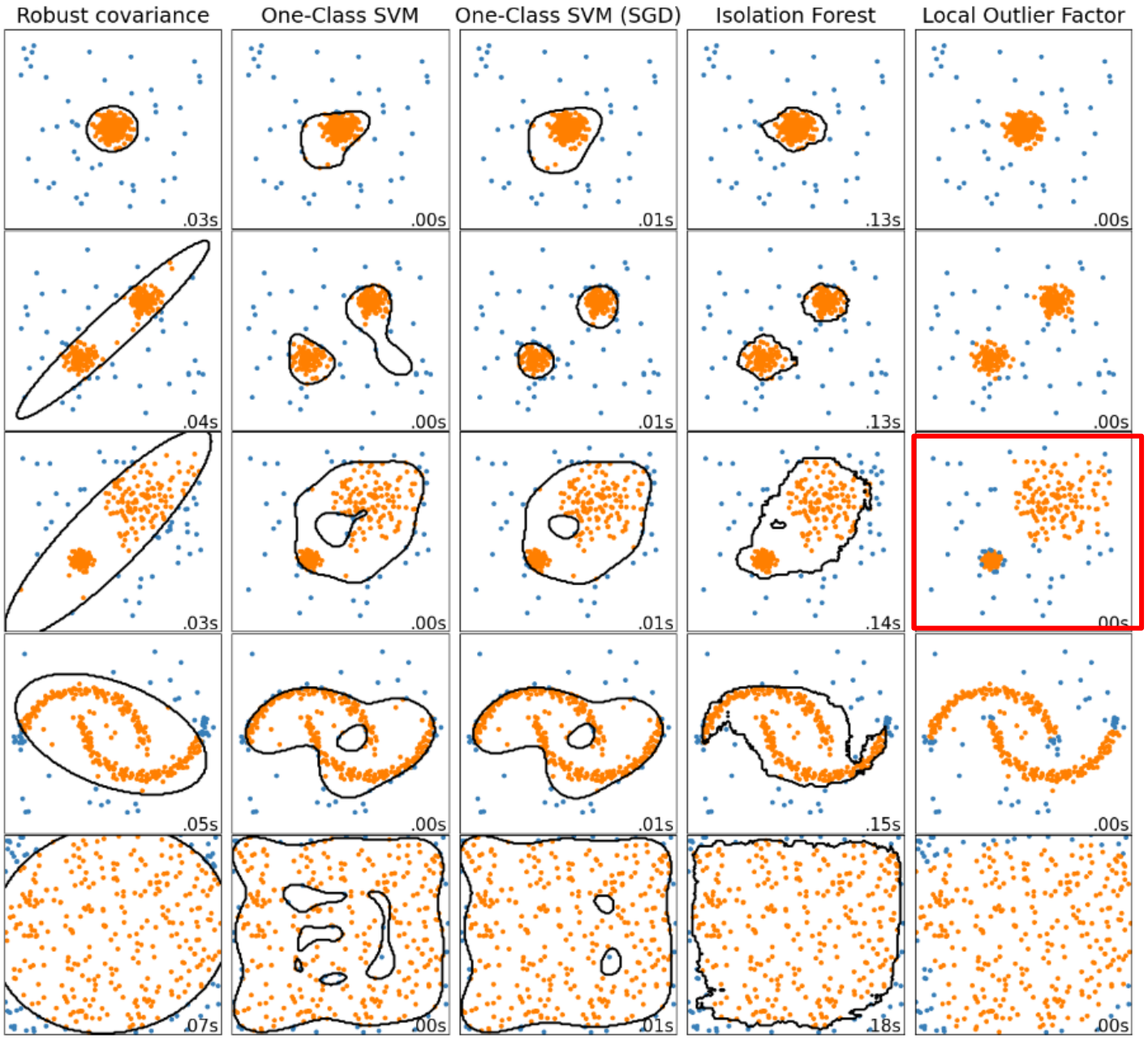
  from sklearn.neighbors import LocalOutlierFactor

- Isolation Forest (for large scale dataset)

  from sklearn.ensemble import IsolationForest

- Robust covariance (for Gaussian distributed dataset)

  from sklearn.covariance import EllipticEnvelope



Different Density

Evaluation of Anomaly Detection

If class labels are present, then use standard evaluation approaches for rare class such as precision, recall, or false positive rate

- FPR is also know as false alarm rate

For unsupervised anomaly detection use measures provided by the anomaly method

- E.g. reconstruction error or gain

Can also look at histograms of anomaly scores.

Distribution of Anomaly Scores

Anomaly scores should show a tail

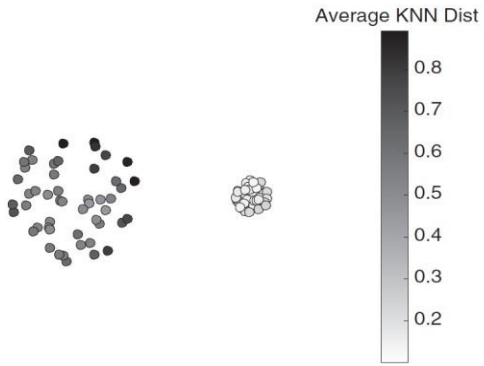


Figure 10.17. Anomaly score based on average distance to fifth nearest neighbor.

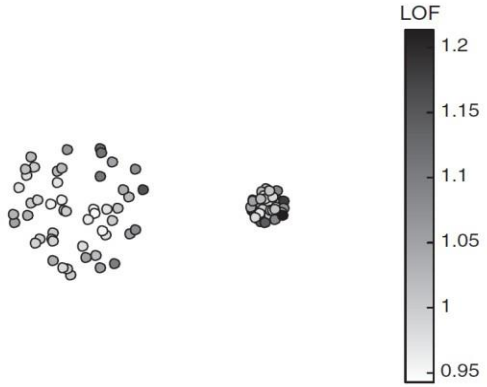


Figure 10.18. Anomaly score based on LOF using five nearest neighbors.