Clustering - Evaluation

Introduction

- For classification, evaluation is an integral part when developing a classification model:
  - Well accepted measures: accuracy, cross-validation
- Cluster evaluation is not commonly used part of cluster analysis. Why not?
  - Cluster analysis conducted as part of an exploratory data ----> complicated addition
  - Different type of clusters ----> seems to require different measures

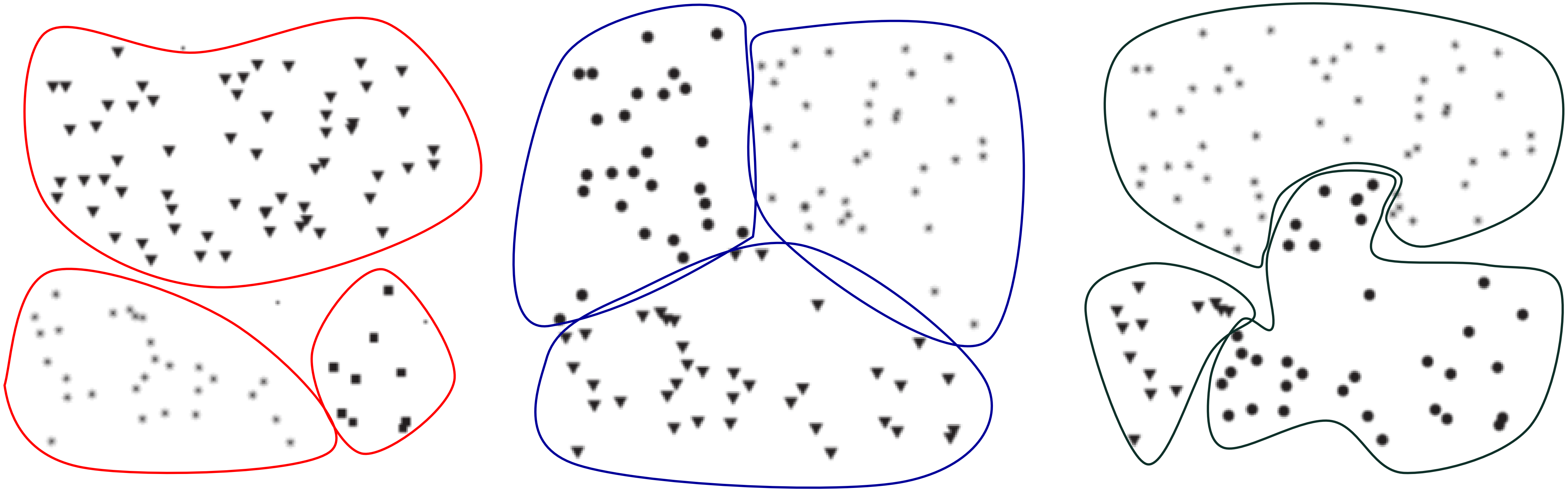
    e.g., K-means clusters might be evaluated using SSE which will not work well for density-based clusters
- However, cluster evaluation should be part of any cluster analysis

Good clustering?

Original points: Randomly (uniformly) distributed

Good clustering?

The clusters do not look compelling for any of the three methods.



DBSCAN K-Means Complete Link

Clustering algorithms will always find clusters in a data set even if the data has no natural cluster structure

Issues

- Since any clustering algorithms will always find clusters in a data set even if the data is random, we need to evaluate the cluster tendency.

- How to evaluate the resulting clusters?
  - Without using external data
  - Using external labels
- How to determine the correct number of clusters?
- How to compare two sets of clusters to determine which is better?
- Is there a non-random structure in the data: cluster tendency

Cluster Evaluation – Validity Measures

- The evaluation measures that are applied to judge various aspects of cluster validity is classified into 2 types
- Unsupervised: measures the goodness of the clustering structure without using external information

$$\text{overall validity} = \sum_{i=1}^{k} w_i \,\text{validity}(C_i)$$

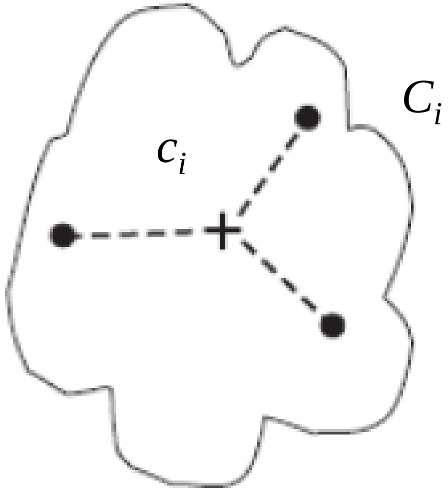
  - Example: SSE
- Supervised: measures the goodness of the clustering structure by the extent it matches some given external structure

Unsupervised Measures – Prototype Based

- Cohesion:
  - measures how close objects are within each cluster

$$cohesion(C_i) = \sum_{x \in C_i} proximity(x, c_i)$$

If proximity is defined as the square of the Euclidean distance, then cohesion of a cluster becomes SSE

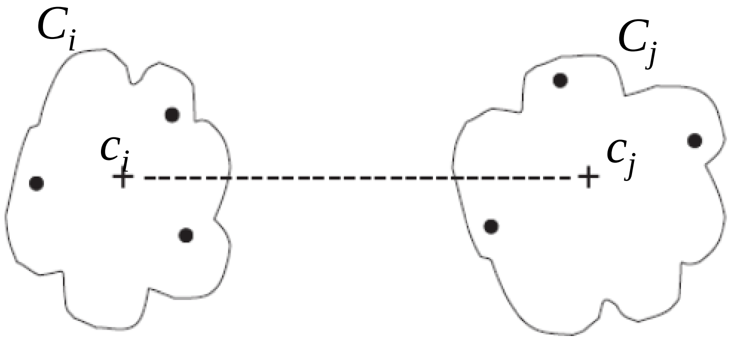


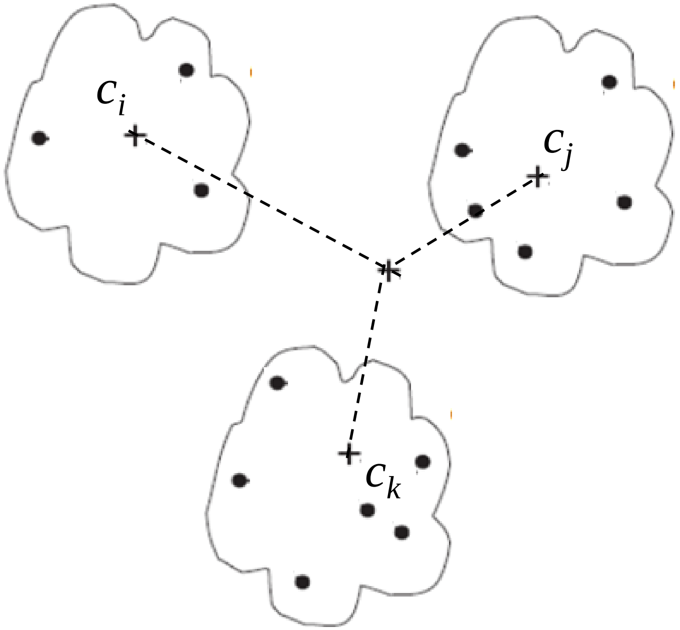
Unsupervised Measures – Prototype Based

- Separation: measures how well separated clusters are from each others

Measure 1: separation of prototypes ci cj from each others

$$separation(C_i, C_j) = proximity(c_i, c_j)$$





Measure 2: separation of prototypes ci from overall prototype c

$$separation(C_i) = proximity(c_i, c)$$

Unsupervised Measures – Prototype Based

- Cohesion and Separation can be used in the overall evaluation of group of clusters as well as individual cluster:

$$cohesion\left(C_i\right) > cohesion\left(C_j\right) \rightarrow C_i \text{ better than } C_j$$

  - If a cluster that not very cohesive, we may split into several sub-clusters
  - If two clusters are relatively cohesive but not well separated, we may merge them.

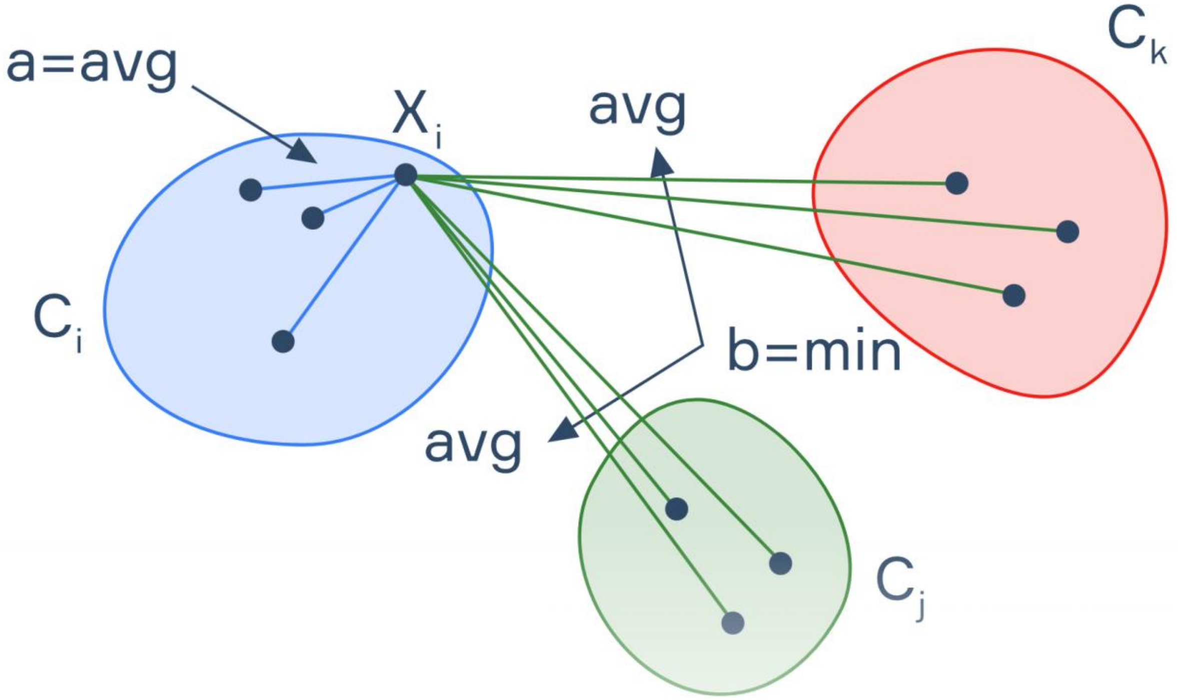
- Can we combine cohesion and separation?

Unsupervised Measures – Prototype Based

- Silhouette Coefficient:

***For a data point i**:*

1. *compute its average distance from each point in its cluster (a_i)*
2. *For each other cluster, compute the average distance from point i to all points in the cluster. Find the smallest with respect to all clusters (b_i)*
3. *Set the silhouette coefficient for point i to $s_i = (b_i - a_i)/max(a_i, b_i)$*

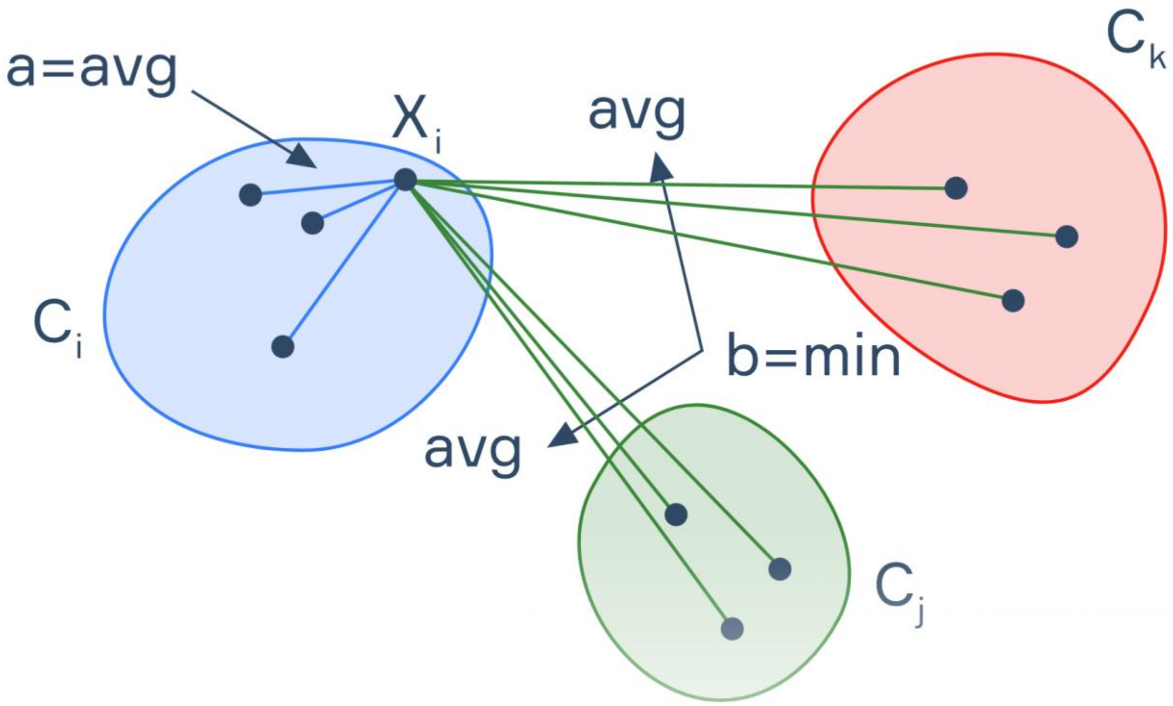


Unsupervised Measures – Prototype Based

- Silhouette Coefficient:

s_i is between -1 and 1

- **s_i < 0** ($b_i < a_i$) => undesirable, point assigned to the wrong cluster
- **$s_i \rightarrow$ 1** ($a_i << b_i$) => desirable
- **s_i = 0** ($a_i = b_i$) => boundary not clearly separated



Unsupervised Measures – Prototype Based

For a cluster:
Average silhouette coefficient of all points in the cluster

For a set of clusters:
Average silhouette coefficient of all points

Question: Will average silhouette coefficients of all points change with different number of cluster?

Class Exercise

Import Libraries

```
import pandas as pd
import numpy as np
import seaborn as sns
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
%matplotlib inline
```

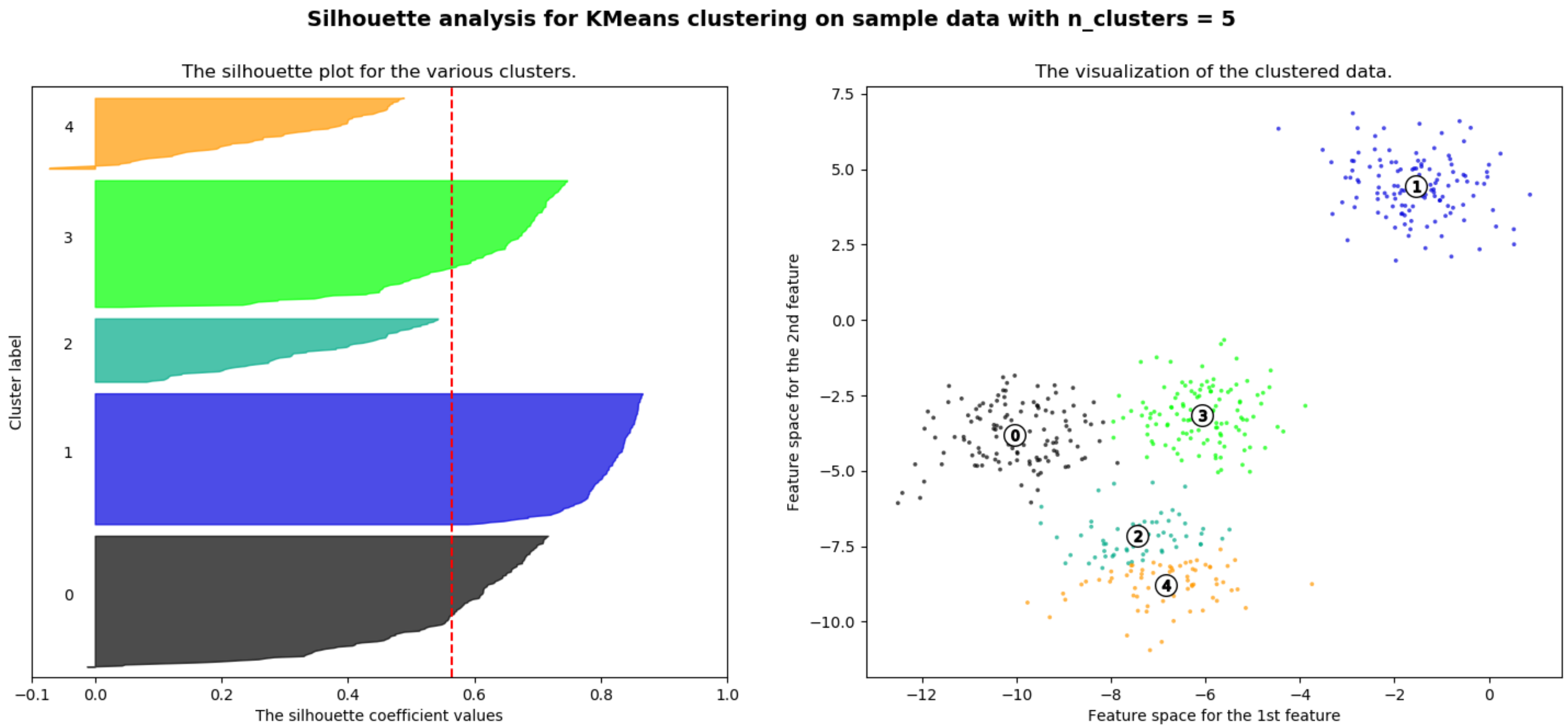
Generate 100 random points

```
X= np.random.rand(50,2)
Y= 2 + np.random.rand(50,2)
Z= np.concatenate((X,Y))
Z=pd.DataFrame(Z) #converting into data frame for ease
```

What will be the silhouette coefficients with 2 clusters and 3 clusters?

Silhouette Analysis

the silhouette analysis is used to choose an optimal value for n_clusters

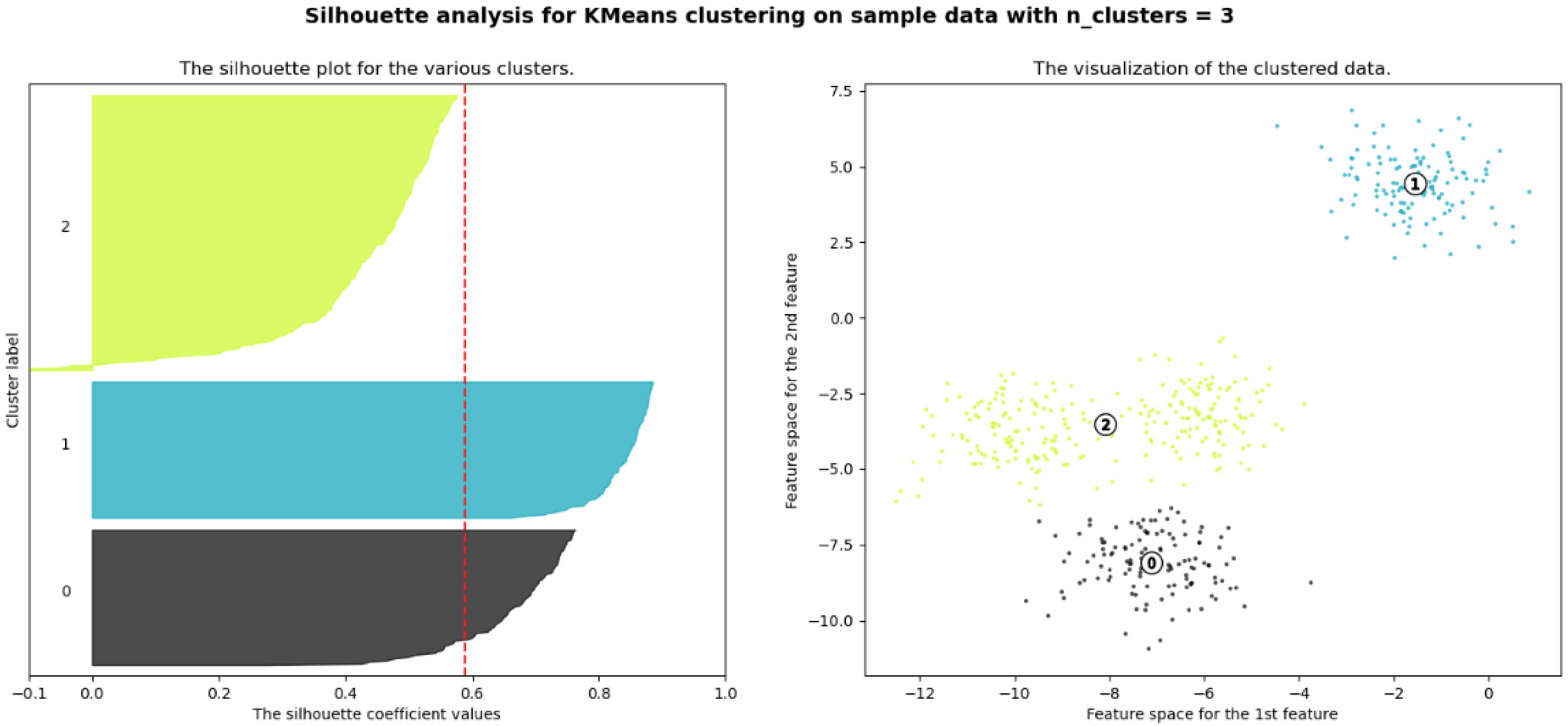


5 is a bad pick for the given data due to the presence of clusters with below average silhouette scores

http://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html

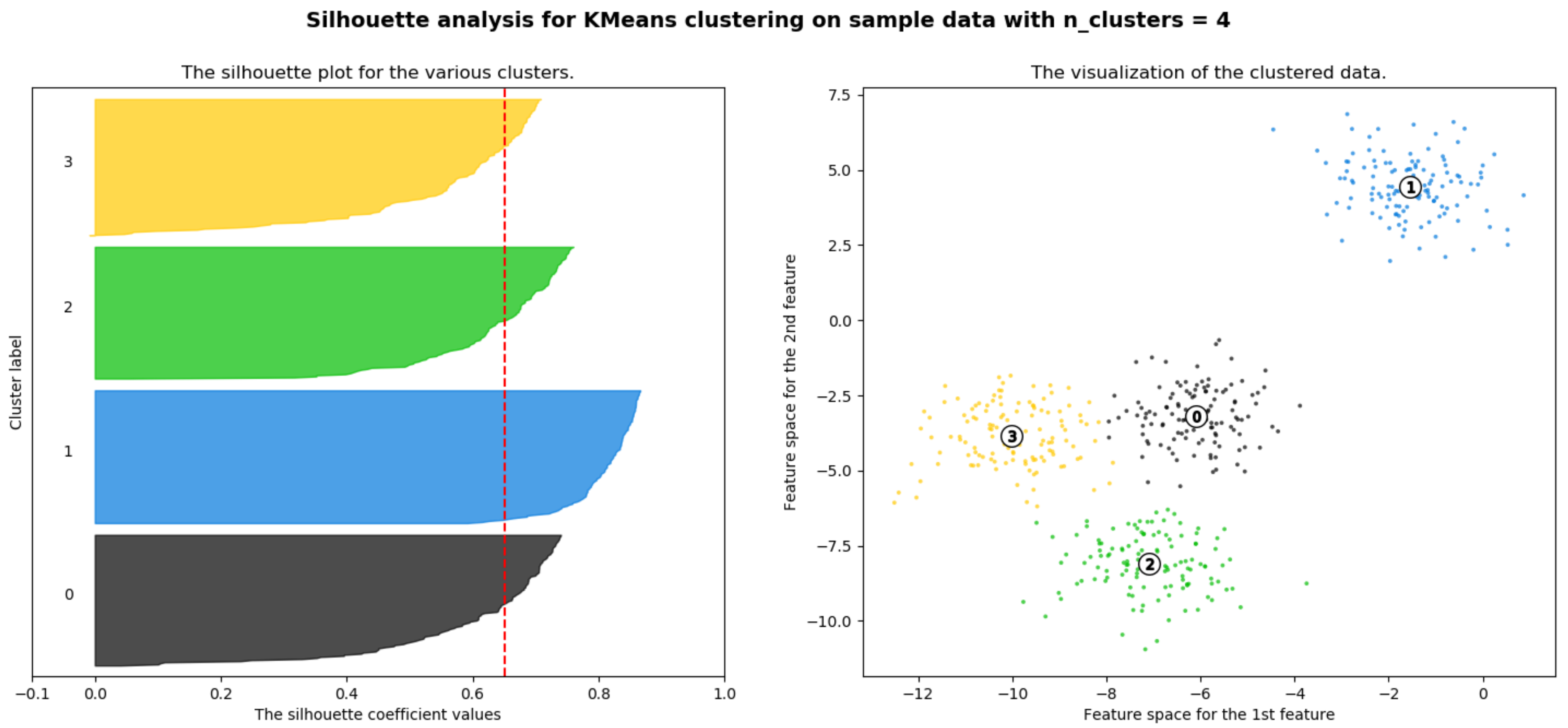
Silhouette Analysis

3 is also a bad pick



http://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html

Silhouette Analysis



When we choose 4, all the plots are more or less of similar thickness and hence are of similar sizes as can be also verified from the labelled scatter plot on the right.

Also all have higher than average silhouette scores.

http://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html

Supervised Measures

- External information available about the data such as class labels
- Classification oriented measures:
  - The degree to which the predicted class labels correspond to the actual class labels
  - e.g., entropy

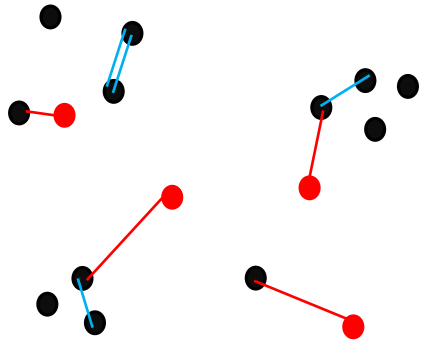
Cluster	Entertainment	Financial	Foreign	Metro	National	Sports	Entropy
1	3	5	40	506	96	27	1.2270
2	4	7	280	29	39	2	1.1472
3	1	1	1	7	4	671	0.1813
4	10	162	3	119	73	2	1.7487
5	331	22	5	70	13	23	1.3976
6	5	358	12	212	48	13	1.5523
Total	354	555	341	943	273	738	1.1450

Cluster Tendency

- Does the data have a cluster structure?
  - Before clustering a dataset we can test if there are actually clusters
- Approach 1:
  - Evaluate the resulting clusters and claim that data has a cluster if at least some of the clusters are of good quality
- Approach 2:
  - Evaluate data without clustering using statistical methods for spatial randomness
    - Hopkins Statistic
    - Comparison against random data sets

Cluster Tendency

- Generate *p* points that are randomly distributed in the data space
- Sample *p* actual points from the space
- Find nearest neighbor distances:
  - u_i from each artificial random point
  - w_i from each real data sample point



- Compute the Hopkins Statistic: $$H = \frac{\sum_{i=1}^{p} u_i}{\sum_{i=1}^{p} u_i + \sum_{i=1}^{p} w_i}$$
- If there is a clustering: distances u_i will be larger than distances $w_{i.}$ A value close to 1 tends to indicate the data is highly clustered,
- If objects are randomly distributed: distance u_i and w_i will be similar, and H will be closer to 0.5
- uniformly distributed data will tend to result in values close to 0

Class Exercise

Implement Hopkin statistic

Generate different types of data (naturally clustered; uniform; random) and calculate their Hopkin statistic.