Clustering Evaluation

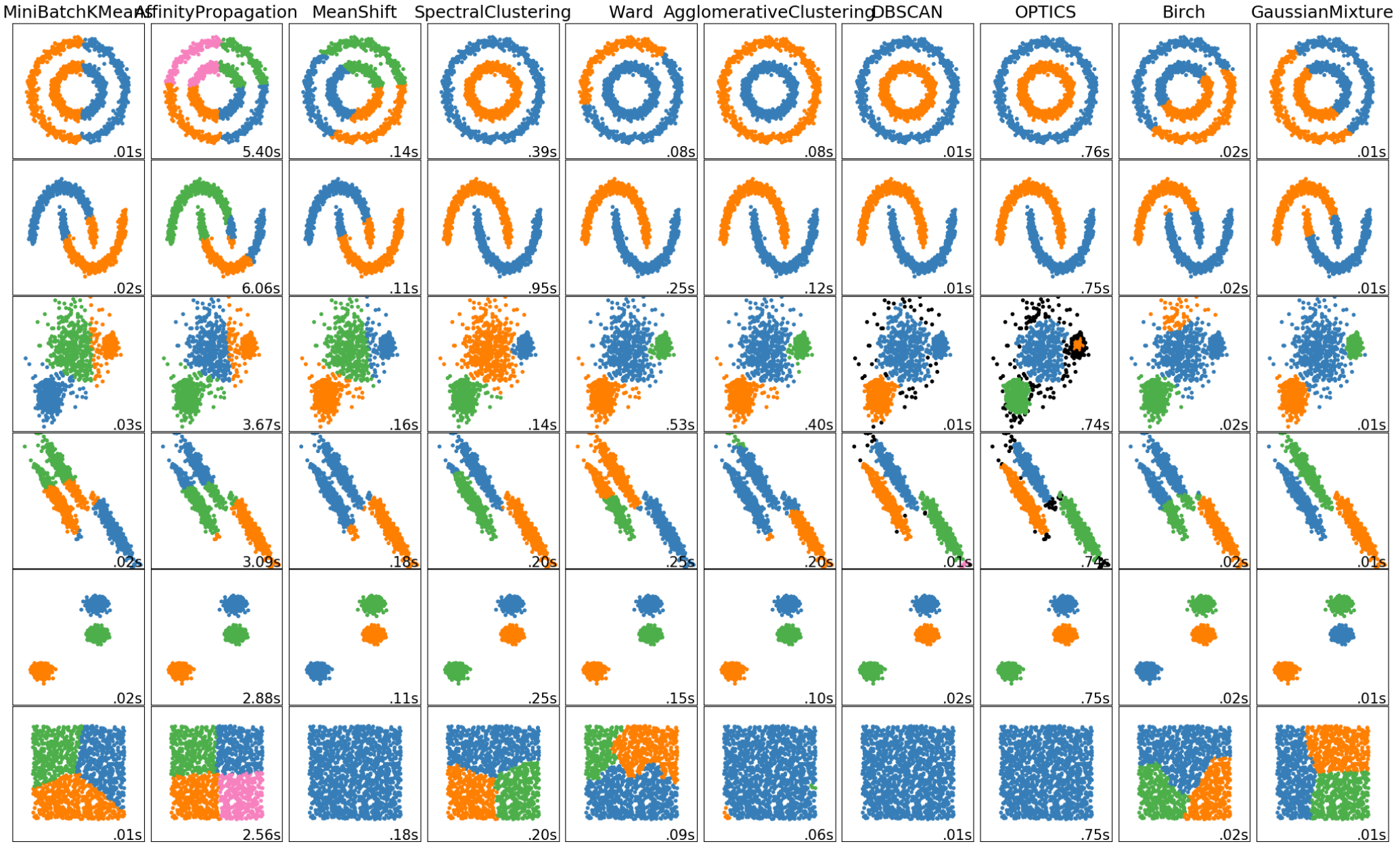
Günce Keziban Orman

Galatasaray University

gunceorman@gmail.com

This presentation is prepared from the slides of

- *Vincent Labatut – Lecture of Datamining in GSU – Comp. Eng. Master*
- *Saeed Aghabozorgi – Lectures of Cognitive Class*
- *Ruoming Jin – Lectures of Data Mining*
- *Tan, Steinbach, Kumar - Lecture Notes for Introduction to Data Mining*

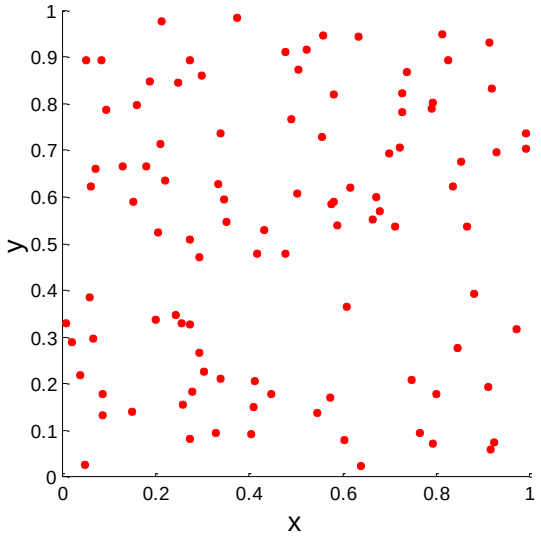

MiniBatchKMeans
AffinityPropagation
MeanShift
SpectralClustering
Ward
AgglomerativeClustering
DBSCAN
OPTICS
Birch
GaussianMixture
.01s
5.40s
.14s
.39s
.08s
.08s
.01s
.76s
.02s
.01s
.02s
6.06s
.11s
.95s
.25s
.12s
.01s
.75s
.02s
.01s
.03s
3.67s
.16s
.14s
.53s
.40s
.01s
.74s
.02s
.01s
.02s
3.09s
.18s
.20s
.25s
.20s
.01s
.74s
.02s
.01s
.02s
2.88s
.11s
.25s
.15s
.10s
.02s
.75s
.02s
.01s
.01s
2.56s
.18s
.20s
.09s
.06s
.01s
.75s
.02s
.01s

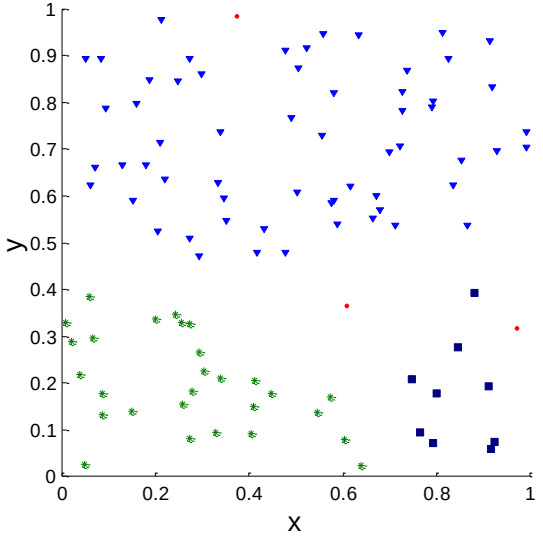
Cluster Validity

- For cluster analysis, the question is how to evaluate the "goodness" of the resulting clusters?
- But "clusters are in the eye of the beholder"!
- Then why do we want to evaluate them?
  - To avoid finding patterns in noise
  - To compare clustering algorithms
  - To compare two sets of clusters
  - To compare two clusters

Clusters found in Random Data

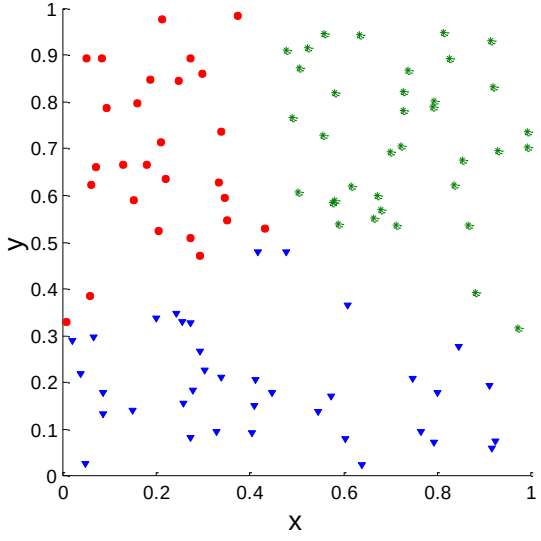
Random Points

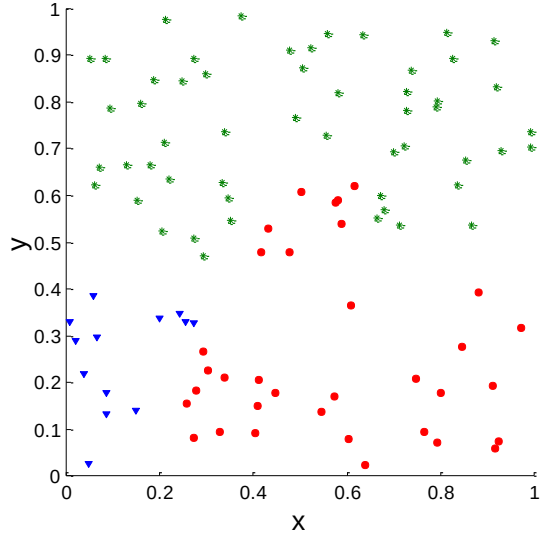




DBSCAN

K-means





Complete Link

Different Aspects of Cluster Validation

1. Determining the clustering tendency of a set of data, i.e., distinguishing whether non-random structure actually exists in the data.
2. Comparing the results of a cluster analysis to externally known results, e.g., to externally given class labels.
3. Evaluating how well the results of a cluster analysis fit the data *without* reference to external information.
    - Use only the data
4. Comparing the results of two different sets of cluster analyses to determine which is better.
5. Determining the 'correct' number of clusters.

For 2, 3, and 4, we can further distinguish whether we want to evaluate the entire clustering or just individual clusters.

Framework for Cluster Validity

- Need a framework to interpret any measure.
  - For example, if our measure of evaluation has the value, 10, is that good, fair, or poor?
- Statistics provide a framework for cluster validity
  - The more “atypical” a clustering result is, the more likely it represents valid structure in the data
  - Can compare the values of an index that result from random data or clusterings to those of a clustering result.
    - If the value of the index is unlikely, then the cluster results are valid
  - These approaches are more complicated and harder to understand.
- For comparing the results of two different sets of cluster analyses, a framework is less necessary.
  - However, there is the question of whether the difference between two index values is significant

Measures of Cluster Validity

- Numerical measures that are applied to judge various aspects of cluster validity, are classified into the following three types.
  - External Index: Used to measure the extent to which *cluster labels match externally supplied class labels*.
    - Purity
  - Internal Index: Used to measure *the goodness of a clustering structure without respect to external information*.
    - Sum of Squared Error (SSE)
  - Relative Index: Used to *compare two different clusterings or clusters.*
    - Often an external or internal index is used for this function, e.g., SSE or entropy
- Sometimes these are referred to as criteria instead of indices
  - However, sometimes criterion is the general strategy and index is the numerical measure that implements the criterion.

Measures of Cluster Validity

- **External cluster validation**, which consists in comparing the results of a cluster analysis to an externally known result, such as externally provided class labels. It measures the extent to which cluster labels match externally supplied class labels. Since we know the “true” cluster number in advance, this approach is mainly used for selecting the right clustering algorithm for a specific data set.
- **Internal cluster validation**, which uses the internal information of the clustering process to evaluate the goodness of a clustering structure without reference to external information. It can be also used for estimating the number of clusters and the appropriate clustering algorithm without any external data.
- **Relative cluster validation**, which evaluates the clustering structure by varying different parameter values for the same algorithm (e.g.,: varying the number of clusters k). It’s generally used for determining the optimal number of clusters.

External Validation

Algorithm 21.4: Algorithm for matching partitions and clusters

MatchPartitionCluster **(**P,C,$match$**)**:

foreach $p \in P$ **do**
    $match(p) \leftarrow \emptyset$
    foreach $c \in C$ **do**
        $overlap(p, c) \leftarrow \frac{|p \cap c|}{|p|}$
while $overlap \neq \emptyset$ **do**
    $(p_{max}, c_{max}) \leftarrow GetMaxOverlap(overlap)$
    $match(p_{max}) \leftarrow c_{max}$
    $overlap \leftarrow overlap - \{overlap(p_{max}, *), overlap(*, c_{max})\}$

Purity Measure

- It is the percent of the total number of objects(data points) that were classified correctly, in the unit range [0..1]

$$Purity = \frac{1}{N}\sum_{i=1}^{k} max_j |c_i \cap t_j|$$

To calculate Purity first create your confusion matrix This can be done by looping through each cluster c_i and counting how many objects were classified as each class t_i.

```
   |  T1 |  T2  |  T3
--------------------
C1 |  0  |  53  |  10
C2 |  0  |  1   |  60
C3 |  0  |  16  |  0
```

Purity Measure

- It is the percent of the total number of objects(data points) that were classified correctly, in the unit range [0..1]

$$Purity = \frac{1}{N}\sum_{i=1}^{k} max_j |c_i \cap t_j|$$

To calculate Purity first create your confusion matrix This can be done by looping through each cluster c_i and counting how many objects were classified as each class t_i.

```
   |  T1 |  T2  |  T3
--------------------
C1 |  0  |  53  |  10
C2 |  0  |  1   |  60
C3 |  0  |  16  |  0
```

Then for each cluster c_i, select the maximum value from its row, sum them together and finally divide by the total number of data points.

```
Purity = (53 + 60 + 16) / 140 = 0.92142
```

Precision & Recall & F-measures

- True Positive (TP) Assignment: when similar members are assigned to the same cluster. *This is a correct decision*.
- True Negative (TN) Assignment: when dissimilar members are assigned to different clusters. *This is a correct decision.*
- False Negative (FN) Assignment: when similar members are assigned to different clusters. *This is an incorrect decision.*
- False Positive (FP) Assignment: when dissimilar members are assigned to the same cluster. *This is an incorrect decision.*
- Precision and Recall: $P = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \qquad R = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$
- F-measure: $F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}.$

Precision & Recall & F-measures

Calculate TP, FP, TN, FN, Precision, Recall and F-measure

Precision & Recall & F-measures

For FP, we need to compute dissimilar pairs that are in the same community. For instance, for community 1, this is $(5\times1+5\times1+1\times1)$. *Therefore,*

$$FP=\underbrace{(5\times1+5\times1+1\times1)}_{\text{Community 1}}+\underbrace{(6\times1)}_{\text{Community 2}}+\underbrace{(4\times2)}_{\text{Community 3}}=25. \quad (6.37)$$

FN computes similar members that are in different communities. For instance, for label +, this is $(6\times1+6\times2+2\times1)$. *Similarly,*

$$FN=\underbrace{(5\times1)}_{\times}+\underbrace{(6\times1+6\times2+2\times1)}_{+}+\underbrace{(4\times1)}_{\triangle}=29. \quad (6.38)$$

Finally, TN computes the number of dissimilar pairs in dissimilar communities:

$$TN=\underbrace{(\overbrace{5\times6}^{\times,+}+\overbrace{1\times1}^{+,\times}+\overbrace{1\times6}^{\triangle,+}+\overbrace{1\times1}^{\triangle,\times})}_{\text{Communities 1 and 2}}+\underbrace{(\overbrace{5\times4}^{\times,\triangle}+\overbrace{5\times2}^{\times,+}+\overbrace{1\times4}^{+,\triangle}+\overbrace{1\times2}^{\triangle,+})}_{\text{Communities 1 and 3}}+\underbrace{(\overbrace{6\times4}^{+,\triangle}+\overbrace{1\times2}^{\times,+}+\overbrace{1\times4}^{\times,\triangle}}_{\text{Communities 2 and 3}}=104. \quad (6.39)$$

Hence,

$$P=\frac{32}{32+25}=0.56 \quad (6.40)$$

$$R=\frac{32}{32+29}=0.52. \quad (6.41)$$

Entropy

- Data clustering involves solving two main problems.
  1. Defining exactly what makes a good clustering of data.
  2. Determining an effective technique to search through all possible combinations of clustering to find the best clustering.
- Entropy addresses the first problem.
- It is the measure of the amount of disorder in a vector. There are several variations of entropy. The most common is called Shannon's entropy. Expressed mathematically, Shannon's entropy is:

$$H(X) = -\sum_{i=0}^{n-1} P(x_i) * log_2(P(x_i))$$

Entropy

$$H(X) = -\sum_{i=0}^{n-1} P(x_i) * log_2(P(x_i))$$

- Suppose you have a vector = { red, red, blue, green, green, green }.
  - x0 = red, x1 = blue and x2 = green.
  - The probability of red is P(x0) = 2/6 = 0.33. P(x1) = 1/6 = 0.17 and P(x2) = 3/6 = 0.50.

```
H(x) = - [ 0.33 * log2(0.33) + 0.17 * log (0.17) + 0.50 * log(0.50) ]
     = - [ (0.33 * -1.58) + (0.17 * -2.58) + (0.50 * -1.00) ]
     = - [ -0.53 + -0.43 + -0.50 ]
     = 1.46
```

- The smallest possible value for entropy is 0.0, which occurs when all symbols in a vector are the same. In other words, there's no disorder in the vector. The larger the value of entropy, the more disorder there is in the associated vector.
- Smaller values of entropy indicate less disorder in a clustering, which means a better clustering.

Entropy ~ EMIAC Algorithm

- Here there are three clusters, k = 0, k = 1 and k = 2.
- Let us define the overall entropy of a clustering as the weighted sum of entropies for each cluster, where the entropy of a cluster is the sum of the entropies of each column.
- For k = 0, the three column entropies are:

```
Color:   H = - [ 1/3 * log2(1/3) + 2/3 * log2(2/3) ]
           = 0.92

Size:    H = - [ 2/3 * log2(2/3) + 1/3 * log2(1/3) ]
           = 0.92

Texture: H = - [ 3/3 * log2(3/3) ]
           = 0.00
```

- The entropy for cluster k = 0 is 0.92 + 0.92 + 0.00 = 1.84.
- The entropy for cluster k = 1 is 1.59 + 0.00 + 0.00 = 1.59.
- The entropy for cluster k = 2 is 0.00 + 1.00 + 0.00 = 1.00.

```
--------------------
Red     Small   Soft
Green   Small   Soft
Green   Large   Soft
--------------------
Red     Medium  Hard
Orange  Medium  Hard
Green   Medium  Hard
--------------------
Blue    Large   Hard
Blue    Medium  Hard
--------------------
```

Entropy~ EMIAC Algorithm

- Now the overall entropy for the clustering is the weighted sum of the cluster entropies, where the weight for each cluster is the probability of the cluster, which is just the number of tuples in the cluster divided by the total number of tuples. So,
  - P(cluster 0) = 3/8 = 0.375,
  - P(cluster 1) = 3/8 = 0.375 and
  - P(cluster 2) = 2/8 = 0.250.
- Putting the individual cluster entropies and their weights together gives the overall EMIAC entropy of the clustering:

```
E = (1.84)(0.375) + (1.59)(0.375) + (1.00)(0.250)
  = 0.688 + 0.595 + 0.250
  = 1.533
```

```
--------------------
Red     Small   Soft
Green   Small   Soft
Green   Large   Soft
--------------------
Red     Medium  Hard
Orange  Medium  Hard
Green   Medium  Hard
--------------------
Blue    Large   Hard
Blue    Medium  Hard
--------------------
```

Normalized Mutual Information

- Normalized Mutual Information:

$$NMI(Y, C) = \frac{2 \times I(Y; C)}{[H(Y) + H(C)]}$$

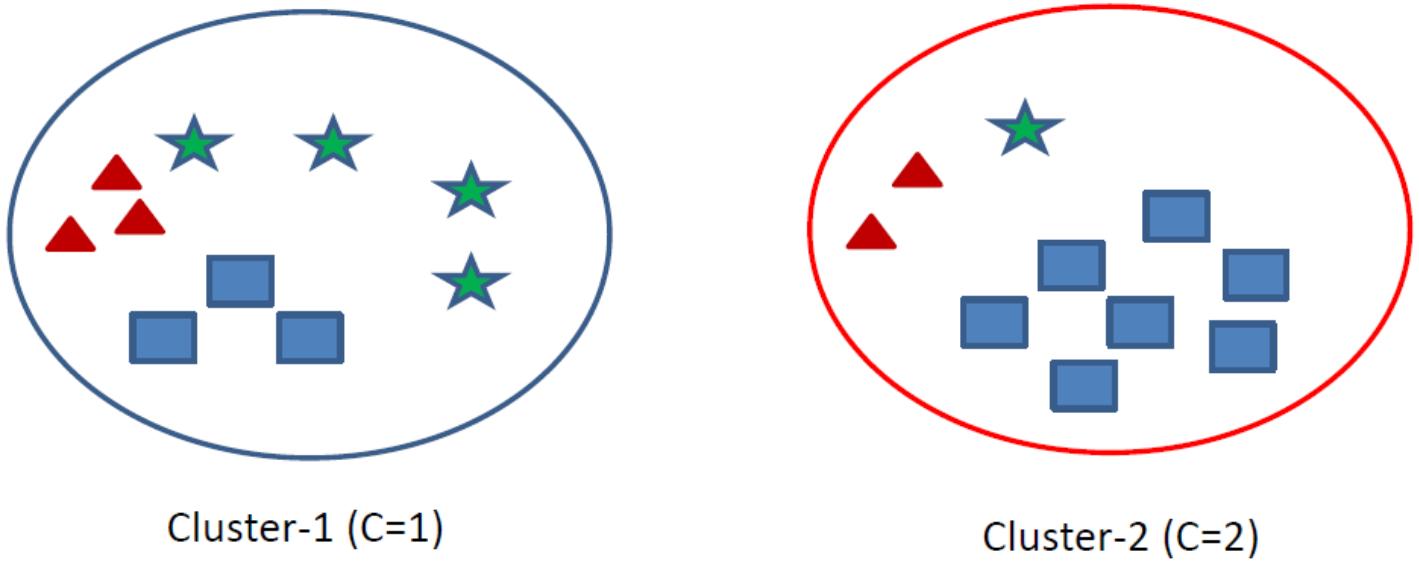
where,

1) Y = class labels
2) C = cluster labels
3) H(.) = Entropy
4) I(Y;C) = Mutual Information b/w Y and C

Note: All logs are base-2.

Normalized Mutual Information

- Assume m=3 classes and k=2 clusters



Normalized Mutual Information

H(Y) = Entropy of Class Labels

- P(Y=1) = 5/20 = ¼
- P(Y=2) = 5/20 = ¼
- P(Y=3) = 10/20 = ½
- $H(Y) = -\frac{1}{4}\log\left(\frac{1}{4}\right) - \frac{1}{4}\log\left(\frac{1}{4}\right) - \frac{1}{2}\log\left(\frac{1}{2}\right) = 1.5$

This is calculated for the entire dataset and can be calculated prior to clustering, as it will not change depending on the clustering output.

Normalized Mutual Information

H(C) = Entropy of Cluster Labels

- P(C=1) = 10/20 = 1/2
- P(C=2) = 10/20 = ½
- H(Y) = $-\frac{1}{2}\log\left(\frac{1}{2}\right) - \frac{1}{2}\log\left(\frac{1}{2}\right) = 1$

This will be calculated every time the clustering changes. You can see from the figure that the clusters are balanced (have equal number of instances).

Normalized Mutual Information

I(Y;C)= Mutual Information

- Mutual information is given as:
  - $I(Y;C) = H(Y) - H(Y|C)$
  - We already know H(Y)
  - H(Y|C) is the entropy of class labels within each cluster, how do we calculate this??

Mutual Information tells us the reduction in the entropy of class labels that we get if we know the cluster labels. (Similar to Information gain in deicison trees)

Normalized Mutual Information

H(Y|C): conditional entropy of class labels for clustering C

- Consider Cluster-1:
  - P(Y=1|C=1)=3/10 (three triangles in cluster-1)
  - P(Y=2|C=1)=3/10 (three rectangles in cluster-1)
  - P(Y=3|C=1)=4/10 (four stars in cluster-1)
  - Calculate conditional entropy as:

$$H(Y|C=1) = -P(C=1) \sum_{y\in\{1,2,3\}} P(Y=y|C=1)\log(P(Y=y|C=1))$$

$$=-\frac{1}{2}\times[\frac{3}{10}\log\left(\frac{3}{10}\right)+\frac{3}{10}\log(\frac{3}{10})+\frac{4}{10}\log(\frac{4}{10})] = 0.7855$$

Normalized Mutual Information

H(Y|C): conditional entropy of class labels for clustering C

- Now, consider Cluster-2:
  - P(Y=1|C=2)=2/10 (two triangles in cluster-1)
  - P(Y=2|C=2)=7/10 (seven rectangles in cluster-1)
  - P(Y=3|C=2)=1/10 (one star in cluster-1)
  - Calculate conditional entropy as:

$$H(Y|C=2) = -P(C=2) \sum_{y\in\{1,2,3\}} P(Y=y|C=2)\log(P(Y=y|C=2))$$

$$=-\frac{1}{2}\times[\frac{2}{10}\log\left(\frac{2}{10}\right)+\frac{7}{10}\log(\frac{7}{10})+\frac{1}{10}\log(\frac{1}{10})] = 0.5784$$

Normalized Mutual Information

I(Y;C)

- Finally the mutual information is:

$$I(Y;C) = H(Y) - H(Y|C)$$
$$= 1.5 - [0.7855 + 0.5784]$$
$$= 0.1361$$

The NMI is therefore,

$$NMI(Y,C) = \frac{2 \times I(Y;C)}{[H(Y) + H(C)]}$$

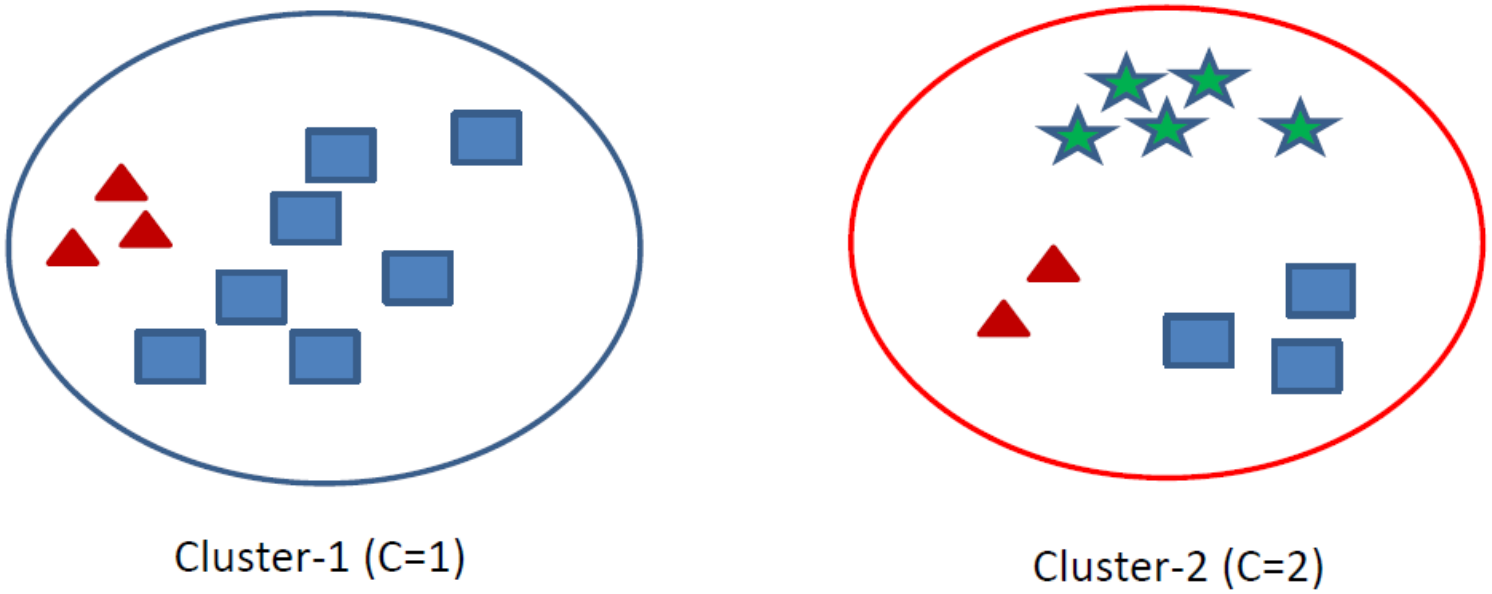
$$NMI(Y,C) = \frac{2 \times 0.1361}{[1.5 + 1]} = 0.1089$$

Normalized Mutual Information

- NMI is a good measure for determining the quality of clustering.
- It is an external measure because we need the class labels of the instances to determine the NMI.
- Since it's normalized we can measure and compare the NMI between different clusterings having different number of clusters.

Normalized Mutual Information

- Calculate the NMI:



Cluster-1 (C=1)

Cluster-2 (C=2)

Class-1 (Y=1) Class-2 (Y=2) Class-3 (Y=3)

Normalized Mutual Information

H(Y|C): conditional entropy of class labels for clustering C

- Consider Cluster-1:
  - P(Y=1|C=1)=3/10 (three triangles in cluster-1)
  - P(Y=2|C=1)=7/10 (seven rectangles in cluster-1)
  - P(Y=3|C=1)=0/10 (no stars in cluster-1)
  - Calculate conditional entropy as:

$$H(Y|C=1) = -P(C=1)\sum_{y\in\{1,2,3\}} P(Y=y|C=1)\log(P(Y=y|C=1))$$

$$=-\frac{1}{2}\times\left[\frac{3}{10}\log\left(\frac{3}{10}\right)+\frac{0}{10}\log\left(\frac{0}{10}\right)+\frac{7}{10}\log\left(\frac{7}{10}\right)\right] = 0.4406$$

We used 0log(0)=0

Normalized Mutual Information

H(Y|C): conditional entropy of class labels for clustering C

- Now, consider Cluster-2:
  - P(Y=1|C=2)=2/10 (two triangles in cluster-1)
  - P(Y=2|C=2)=3/10 (three rectangles in cluster-1)
  - P(Y=3|C=2)=5/10 (five stars in cluster-1)
  - Calculate conditional entropy as:

$$H(Y|C=2) = -P(C=2) \sum_{y\in\{1,2,3\}} P(Y=y|C=2)\log(P(Y=y|C=2))$$

$$=-\frac{1}{2}\times\left[\frac{2}{10}\log\left(\frac{2}{10}\right)+\frac{3}{10}\log\left(\frac{3}{10}\right)+\frac{5}{10}\log\left(\frac{5}{10}\right)\right] = 0.7427$$

Normalized Mutual Information

I(Y;C)

- Finally the mutual information is:

$$I(Y;C) = H(Y) - H(Y|C)$$
$$= 1.5 - [0.4406 + 0.7427]$$
$$= 0.3167$$

The NMI is therefore,

$$NMI(Y,C) = \frac{2 \times I(Y;C)}{[H(Y) + H(C)]}$$

$$NMI(Y,C) = \frac{2 \times 0.3167}{[1.5 + 1]} = 0.2533$$

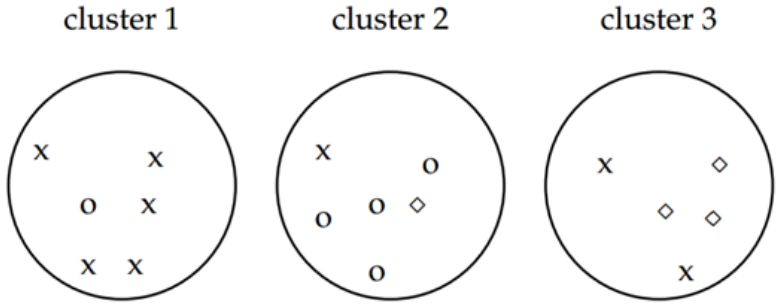
Normalized Mutual Information

- NMI for the second clustering is higher than the first clustering. It means we would prefer the second clustering over the first.
  - You can see that one of the clusters in the second case contains all instances of class-3 (stars).
- If we have to compare two clustering that have different number of clusters we can still use NMI.

Rand Index

- Rand Index (RI) is based on comparing pairs of elements.
- Theory suggests, that similar pairs of elements should be placed in the same cluster, while dissimilar pairs of elements should be placed in separate clusters.
- RI does not care about difference in number of clusters.
- It just cares about True/False pairs of elements.
- Based on this assumption, RI, is calculated;

$$RI = \frac{a+b}{\binom{n}{2}} = \frac{correct\ similar\ pairs + correct\ dissimilar\ pairs}{total\ possible\ pairs}$$



$$RI = \frac{a+b}{\binom{n}{2}} = \frac{\text{correct similar pairs} + \text{correct dissimilar pairs}}{\text{total possible pairs}}$$

	1	2	3
x	5	1	2
o	1	4	0
◊	0	1	3

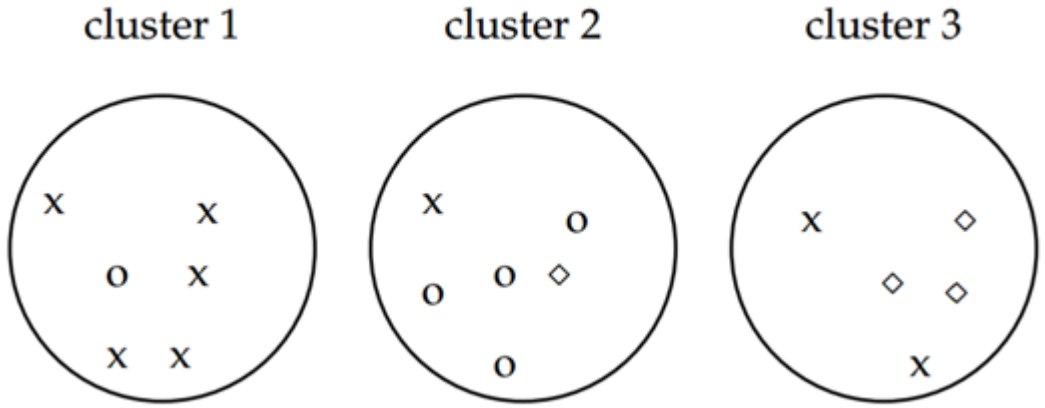
- In denominator, we have total possible pairs, which is (17 2) = 136
- *True Positives* or *correct similar;*
  - a = (5 2) + (1 2) + (2 2) + (1 2) + (4 2) + (0 2) + (0 2) + (1 2) + (3 2) = 10 + 0 + 1 + 0 + 6 + 0 + 0 + 0 + 3 = 20
- *False Positives* or *incorrect dissimilar*;
  - c = 5*1 + 5*2 + 1*2 + 1*4 + 1*0 + 4*0 + 0*1 + 0*3 + 1*3 = 5 + 10 + 2 + 4 + 0 + 0 + 0 + 0 + 3 = 24
- *False Negative* or *incorrect similar*;
  - d = 5*1 + 5*0 + 1*0 +1*4 + 1*1 + 4*1+2*0 + 2*3 + 0*3 = 5 + 0 + 0 + 4 + 1 + 4 + 0 + 6 + 0 = 20
- *True Negatives* or *correct dissimilar;*
  - d = 5*4 + 5*0 + 5*1 + 5*3 + 1*1 + 1*0 + 1*0 + 1*3 + 2*1 + 2*4 + 2*0 + 2*1+ 1*1 + 1*3+ 4*0 + 4*3 = 72
- Rand Index=(20 + 72) / 136 = 0.676

Rand Index

$$RI = \frac{a+b}{\binom{n}{2}} = \frac{\text{correct similar pairs} + \text{correct dissimilar pairs}}{\text{total possible pairs}}$$

	1	2	3
x	5	1	2
o	1	4	0
◊	0	1	3

- In denominator, we have total possible pairs, which is (17 2) = 136
- *True Positives* or *correct similar;*
  - a = (5 2) + (1 2) + (2 2) + (1 2) + (4 2) + (0 2) + (0 2) + (1 2) + (3 2) = 10 + 0 + 1 + 0 + 6 + 0 + 0 + 0 + 3 = 20
- *False Positives* or *incorrect dissimilar*;
  - c = 5*1 + 5*2 + 1*2 + 1*4 + 1*0 + 4*0 + 0*1 + 0*3 + 1*3 = 5 + 10 + 2 + 4 + 0 + 0 + 0 + 0 + 3 = 24

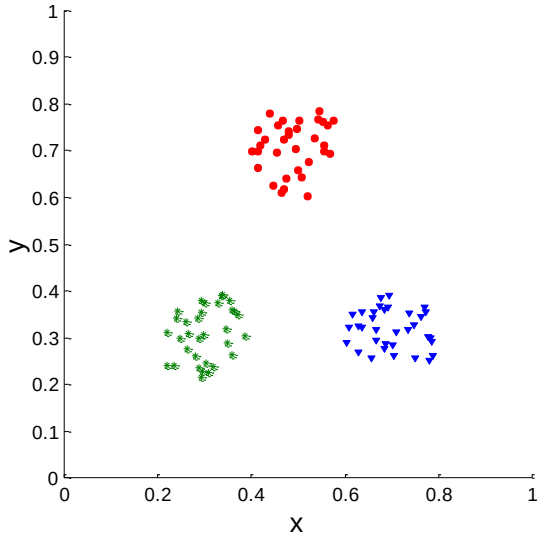


Measuring Cluster Validity Via Correlation

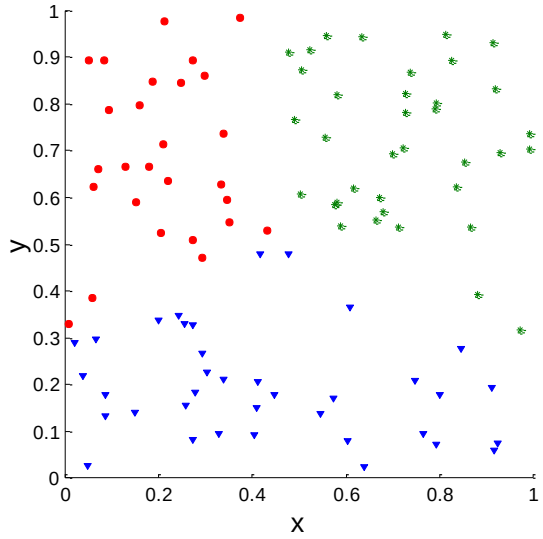
- Two matrices
  - Proximity Matrix
  - “Incidence” Matrix
    - One row and one column for each data point
    - An entry is 1 if the associated pair of points belong to the same cluster
    - An entry is 0 if the associated pair of points belongs to different clusters
- Compute the correlation between the two matrices
  - Since the matrices are symmetric, only the correlation between n(n-1) / 2 entries needs to be calculated.
- High correlation indicates that points that belong to the same cluster are close to each other.
- Not a good measure for some density or contiguity based clusters.

Measuring Cluster Validity Via Correlation

- Correlation of incidence and proximity matrices for the K-means clusterings of the following two data sets.



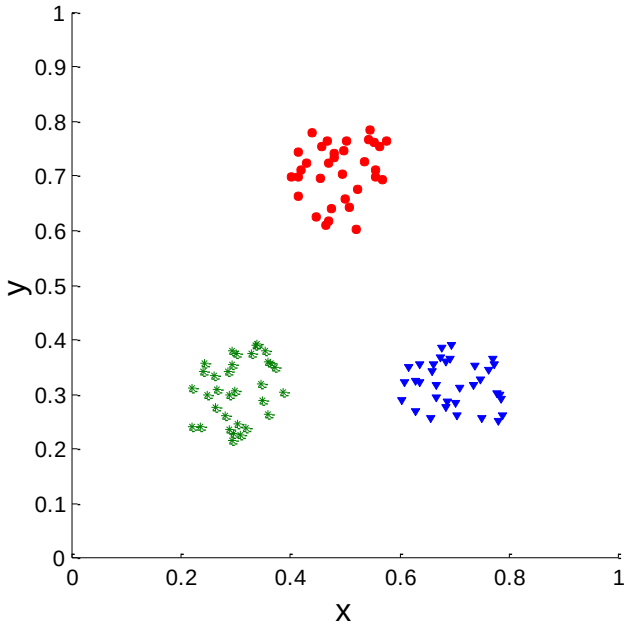
Corr = -0.9235

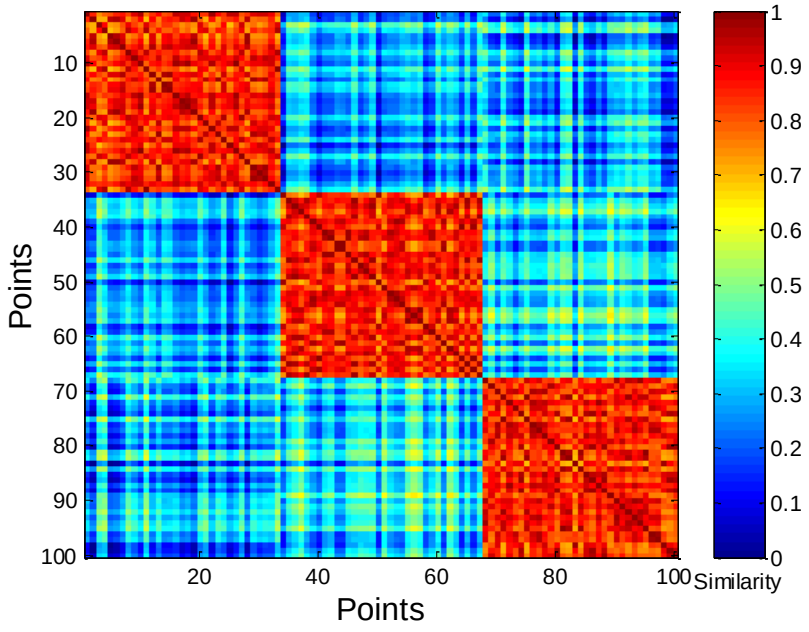


Corr = -0.5810

Using Similarity Matrix for Cluster Validation

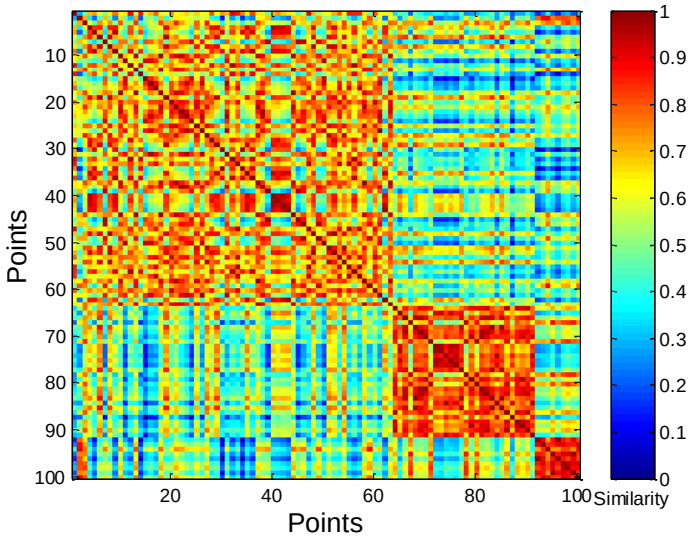
- Order the similarity matrix with respect to cluster labels and inspect visually.

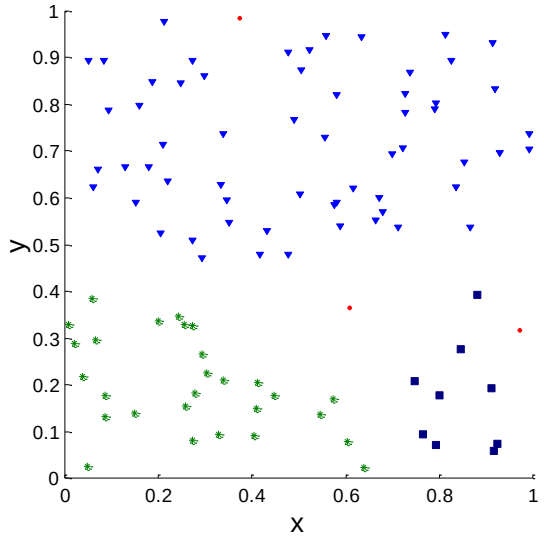




Using Similarity Matrix for Cluster Validation

- Clusters in random data are not so crisp

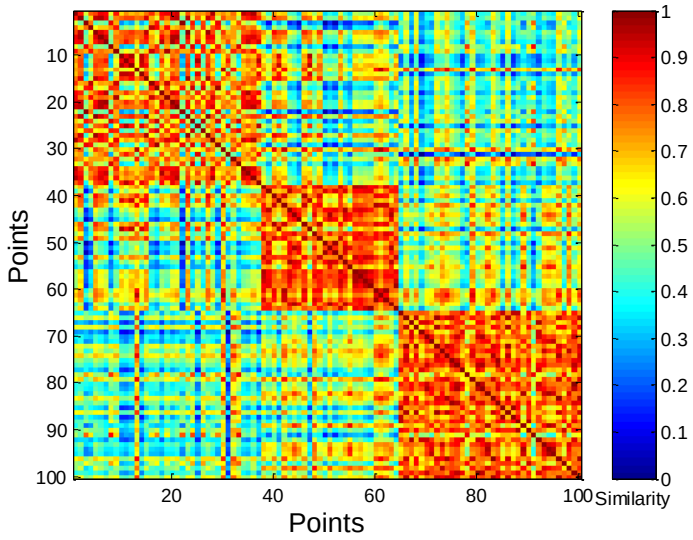


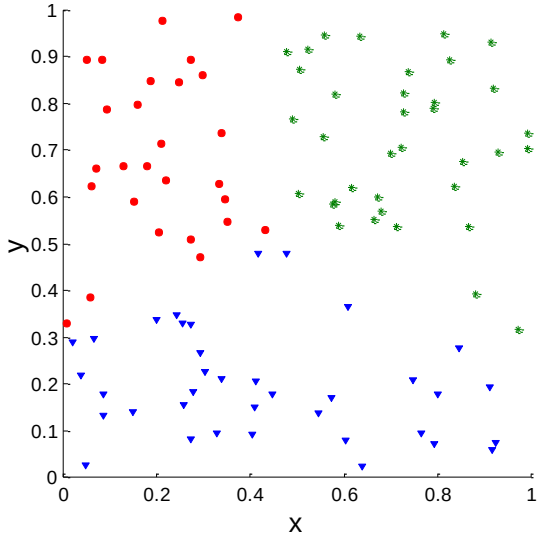


DBSCAN

Using Similarity Matrix for Cluster Validation

- Clusters in random data are not so crisp

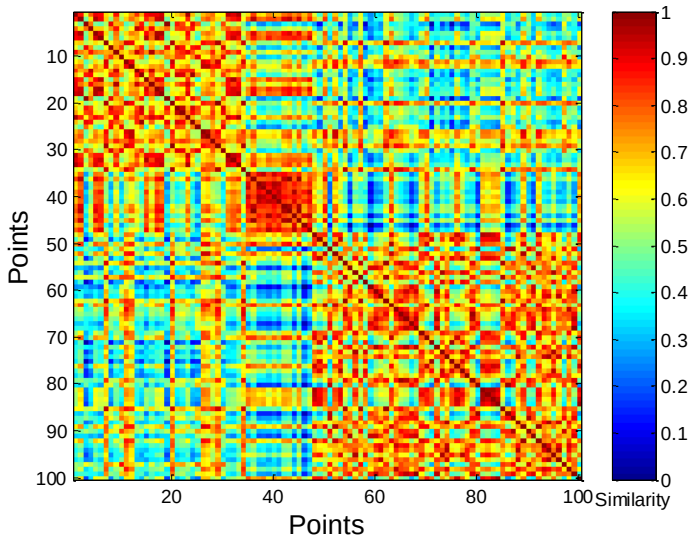


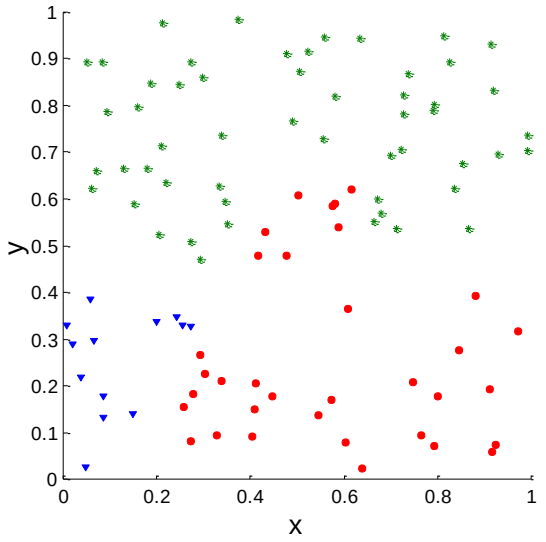


K-means

Using Similarity Matrix for Cluster Validation

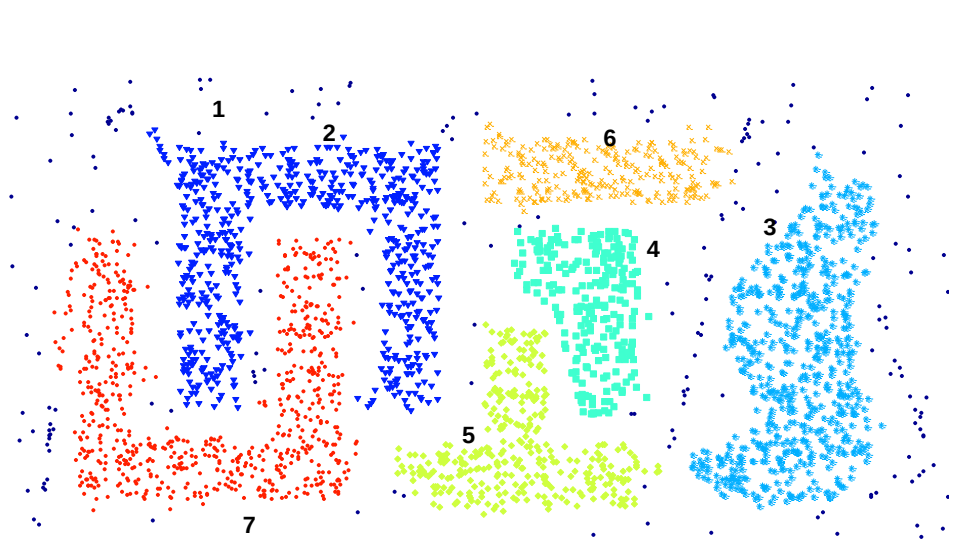
- Clusters in random data are not so crisp

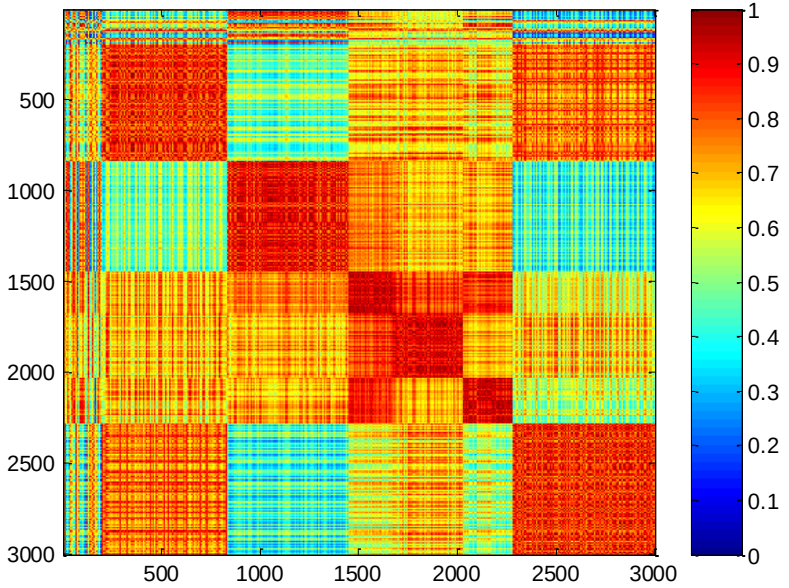




Complete Link

Using Similarity Matrix for Cluster Validation

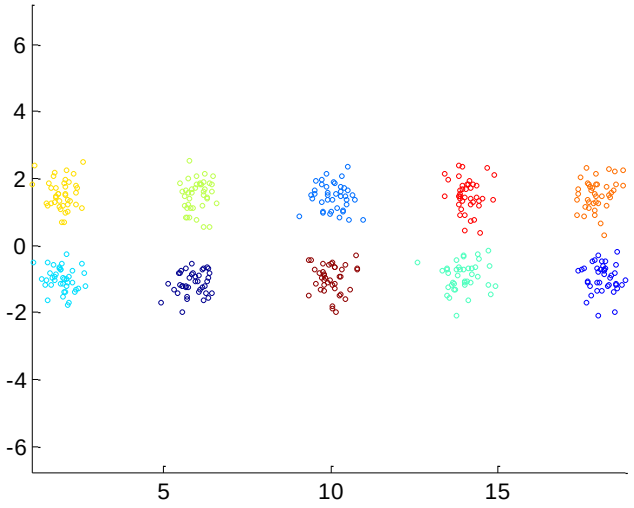


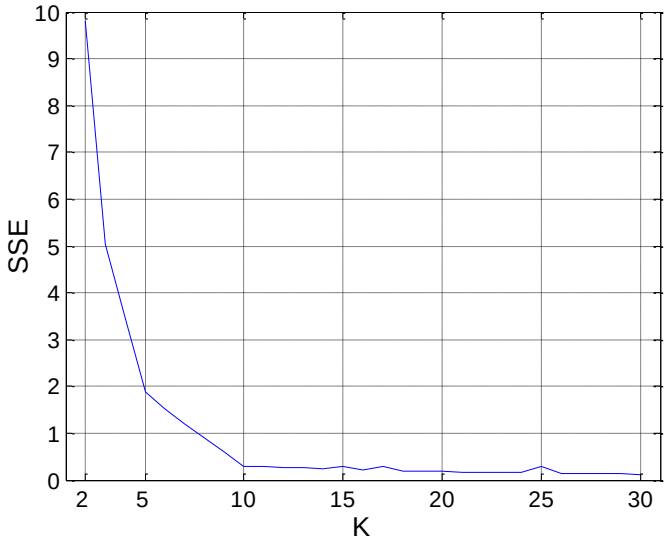


DBSCAN

Internal Measures: SSE

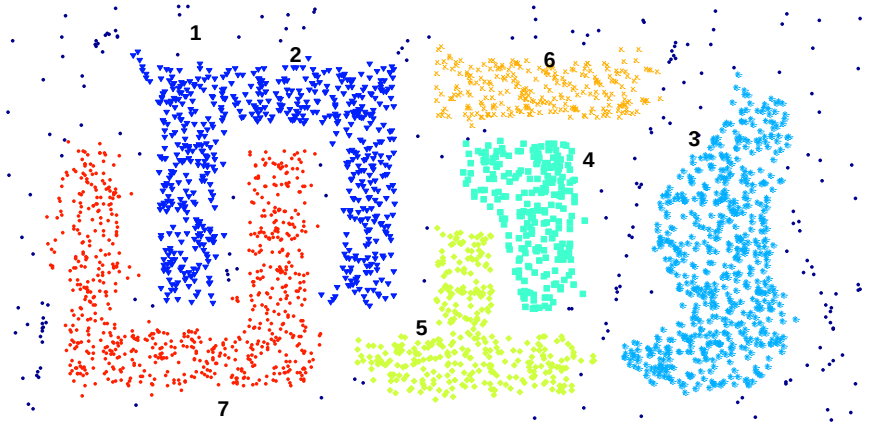
- Clusters in more complicated figures are not well separated
- **Internal Index:** Used to measure the goodness of a clustering structure without respect to external information
  - Sum of Squared Error
- SSE is good **for comparing two clusterings or two clusters** (average SSE).
- Can also be used to estimate the number of clusters

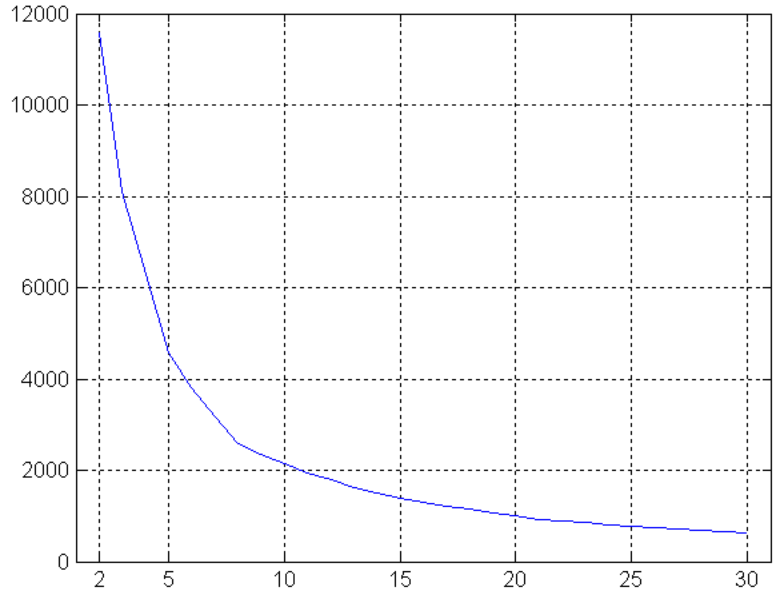




Internal Measures: SSE

- SSE curve for a more complicated data set





SSE of clusters found using K-means

Internal Measures: Cohesion and Separation

- Cluster Cohesion: Measures how closely related are objects in a cluster
- Cluster Separation: Measure how distinct or well-separated a cluster is from other clusters
- Example: Squared Error
  - Cohesion is measured by the within cluster sum of squares (SSE)

$$\boldsymbol{WSS} = \sum_{\boldsymbol{i}} \sum_{\boldsymbol{x} \in \boldsymbol{C_i}} (\boldsymbol{x} - \boldsymbol{m_i})^2$$

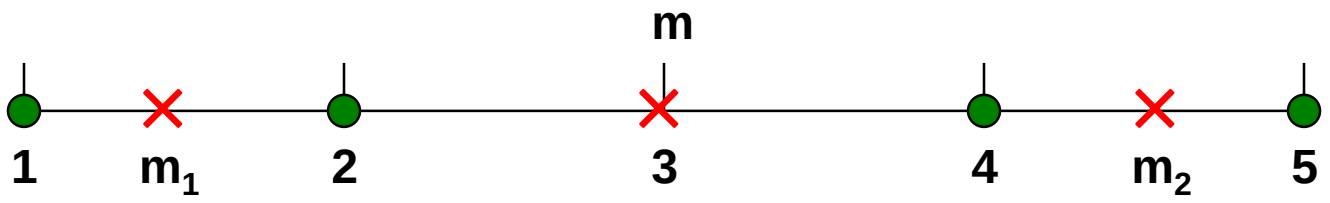
  - Separation is measured by the between cluster sum of squares

$$BSS = \sum_{i} |C_i| (m - m_i)^2$$

Where $|C_i|$ is the size of cluster i

Internal Measures: Cohesion and Separation

- Example: SSE
  - BSS + WSS = constant



K=1 cluster:

$$WSS = (1-3)^2 + (2-3)^2 + (4-3)^2 + (5-3)^2 = 10$$

$$BSS = 4 \times (3-3)^2 = 0$$

$$Total = 10 + 0 = 10$$

K=2 clusters:

$$WSS = (1-1.5)^2 + (2-1.5)^2 + (4-4.5)^2 + (5-4.5)^2 = 1$$

$$BSS = 2 \times (3-1.5)^2 + 2 \times (4.5-3)^2 = 9$$

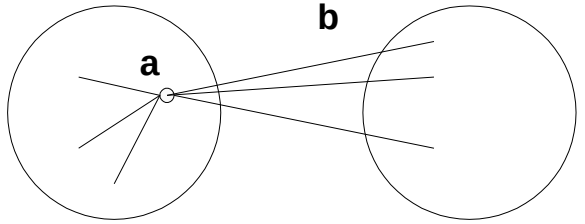
$$Total = 1 + 9 = 10$$

Internal Measures: Silhouette Coefficient

- Silhouette Coefficient combine ideas of both cohesion and separation, but for individual points, as well as clusters and clusterings
- For an individual point, *i*
  - Calculate ***a*** = average distance of *i* to the points in its cluster
  - Calculate ***b*** = min (average distance of *i* to points in another cluster)
  - The silhouette coefficient for a point is then given by

    $s = 1 - a/b$ if $a < b$, (or $s = b/a - 1$ if $a \geq b$, not the usual case)

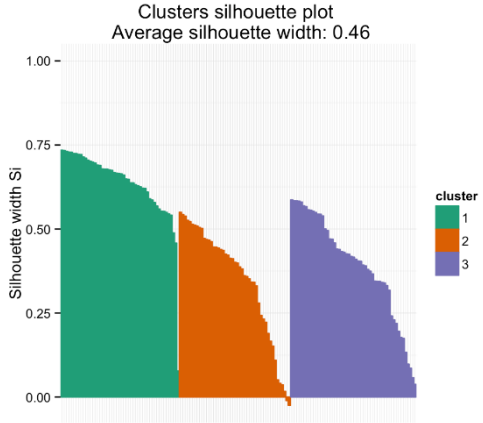
  - Typically between 0 and 1.
  - The closer to 1 the better.

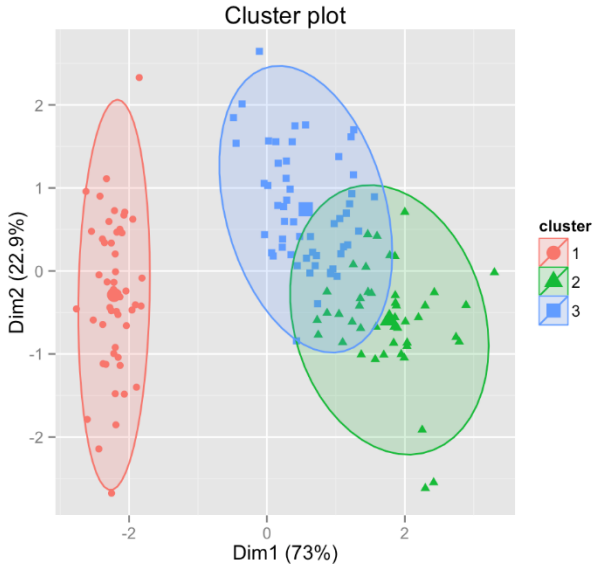


- Can calculate the Average Silhouette width for a cluster or a clustering

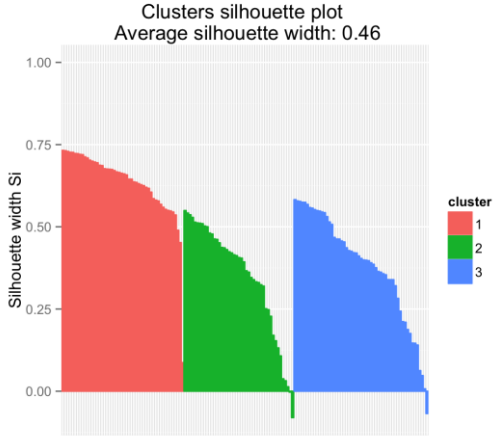
Iris Dataset Example

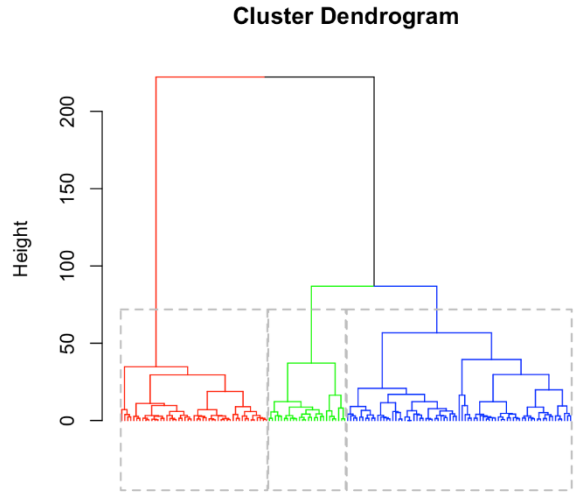
KMEANS



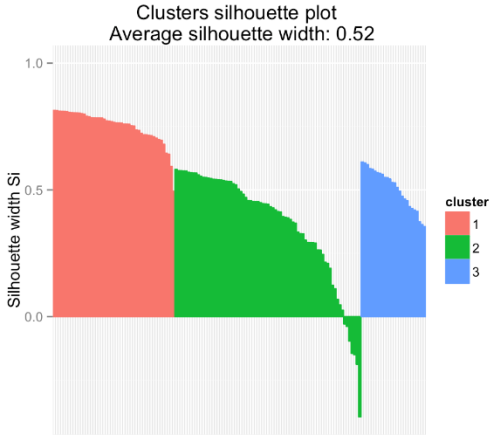


PAM

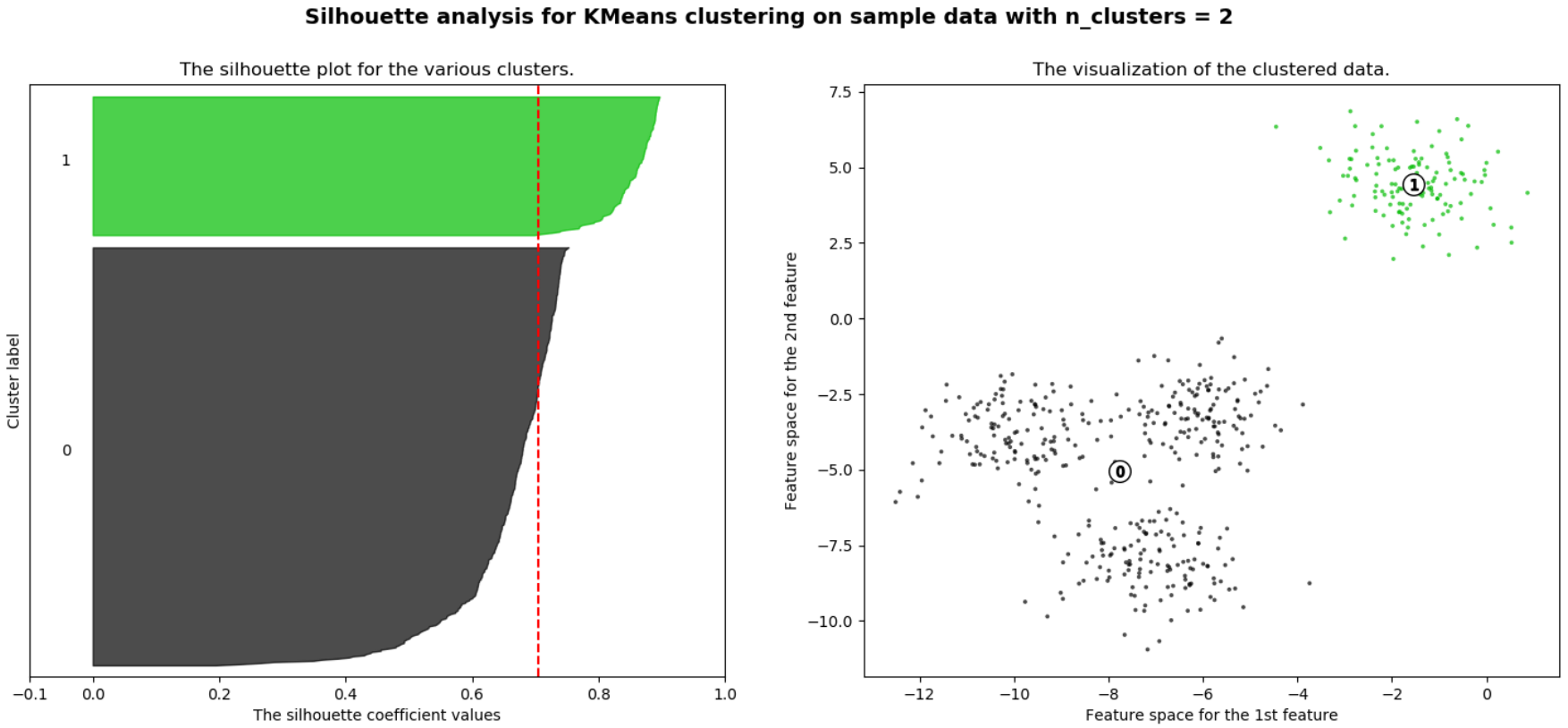




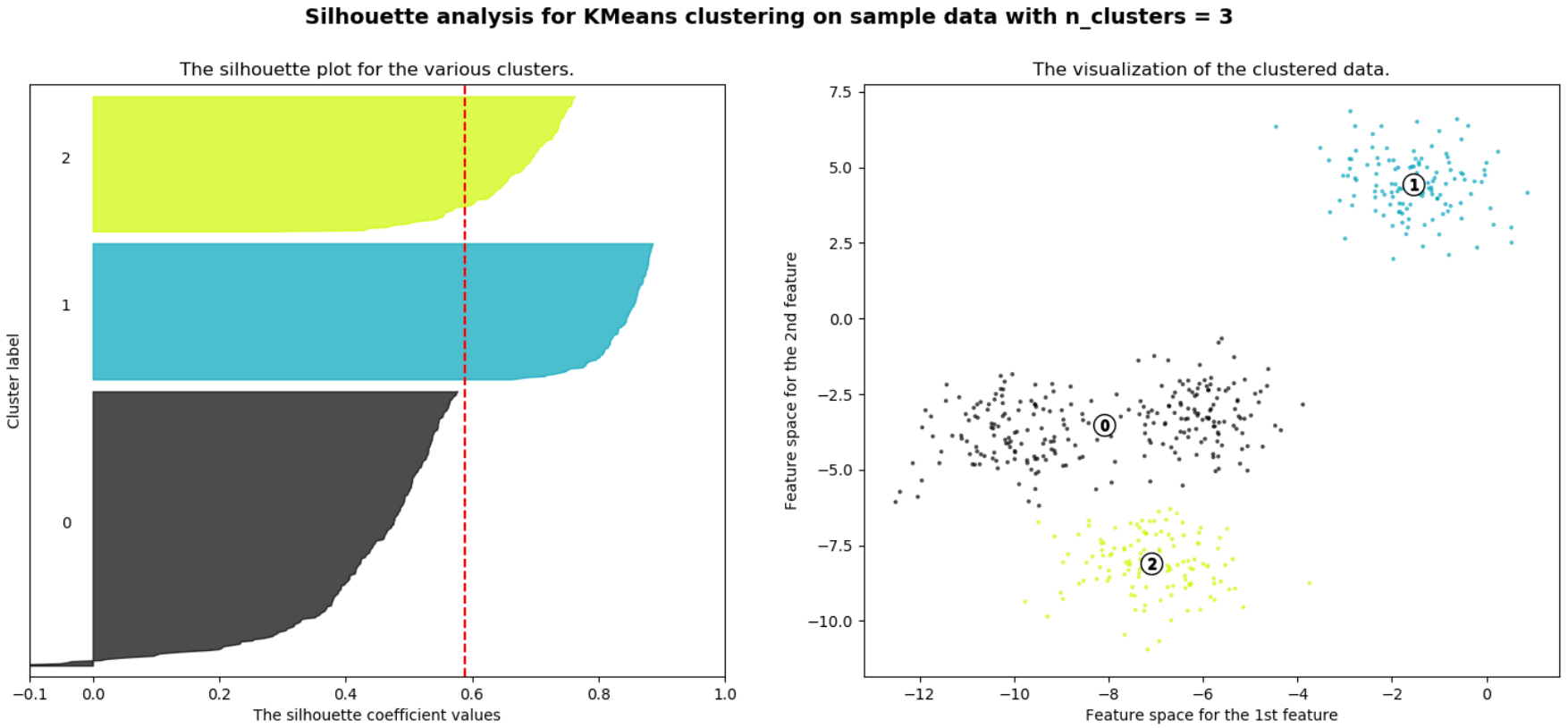
Hierarchical



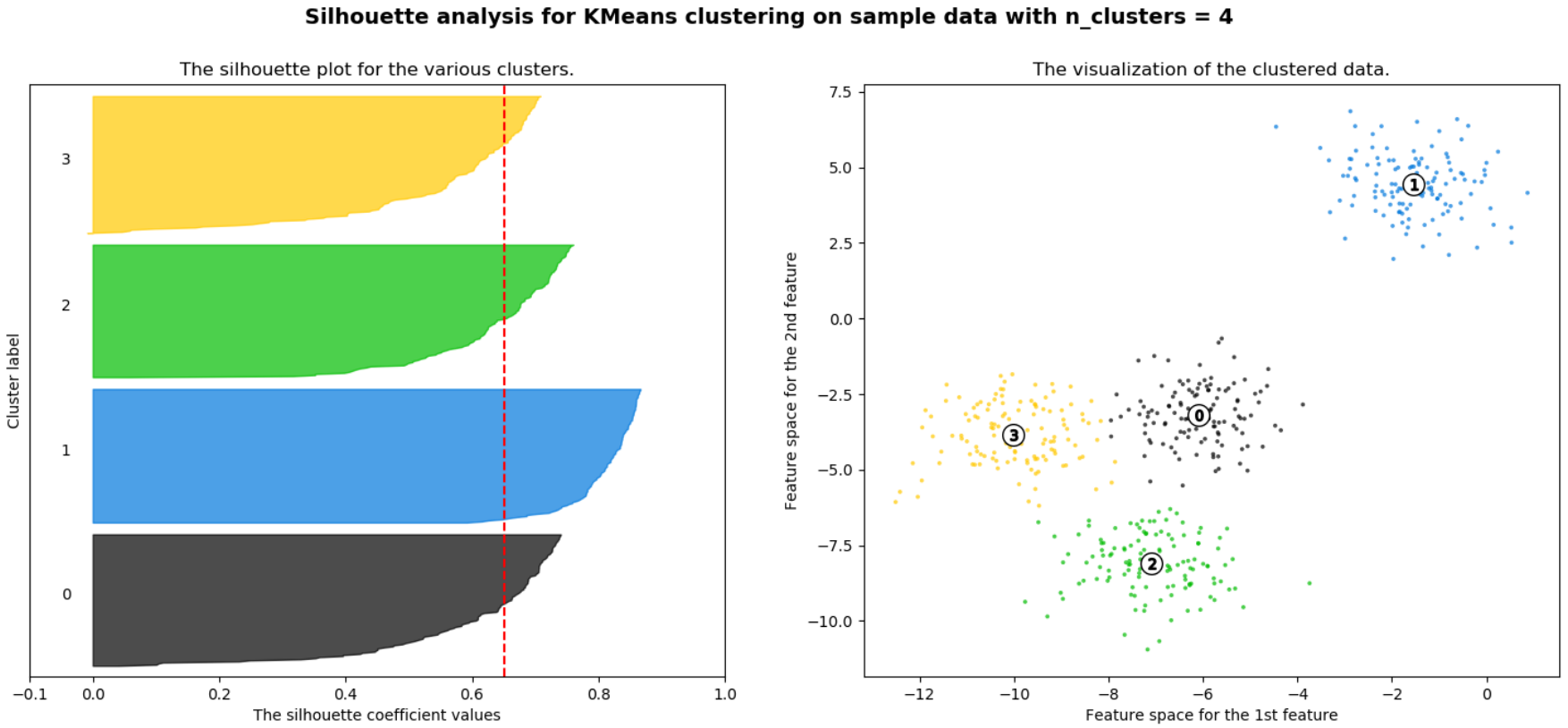
Silhouette Coefficient Example



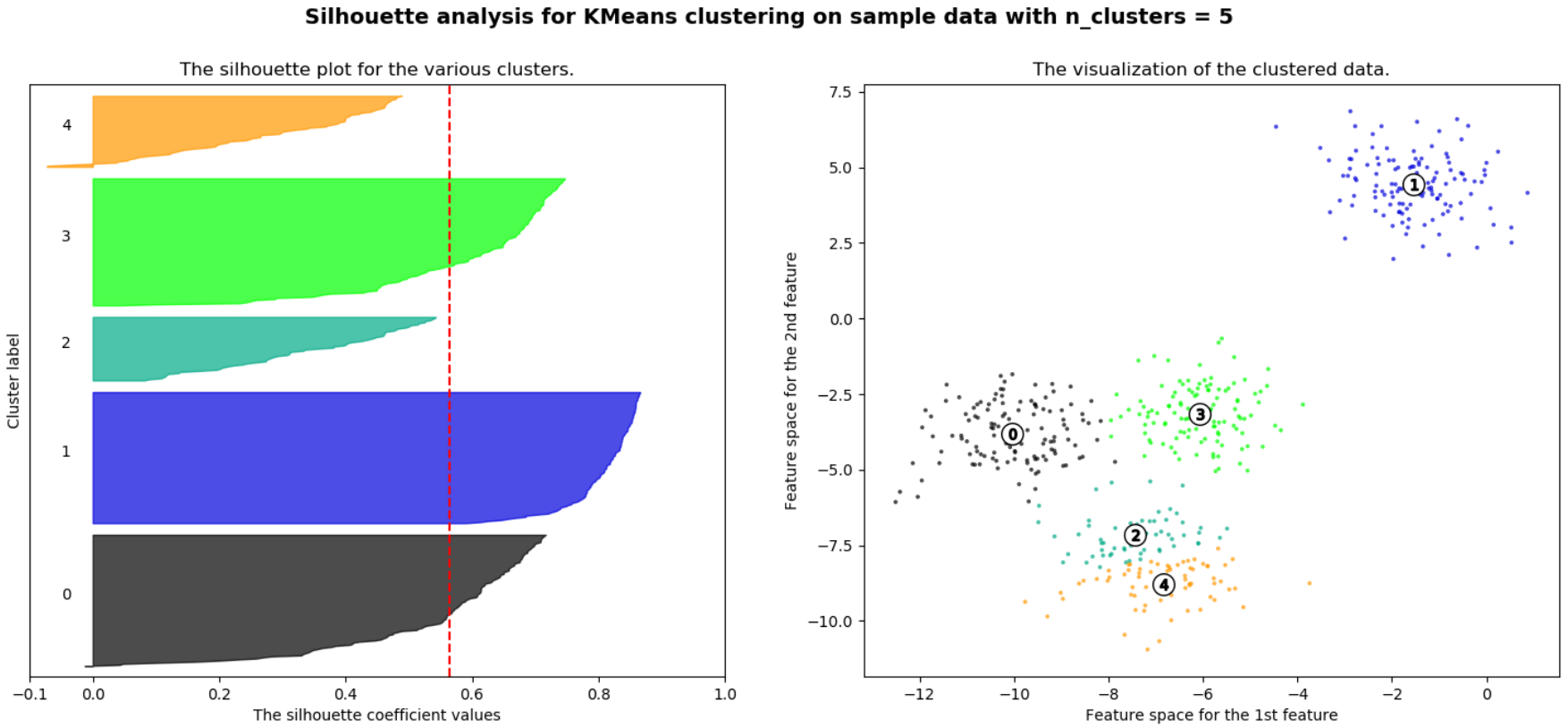
Silhouette Coefficient Example



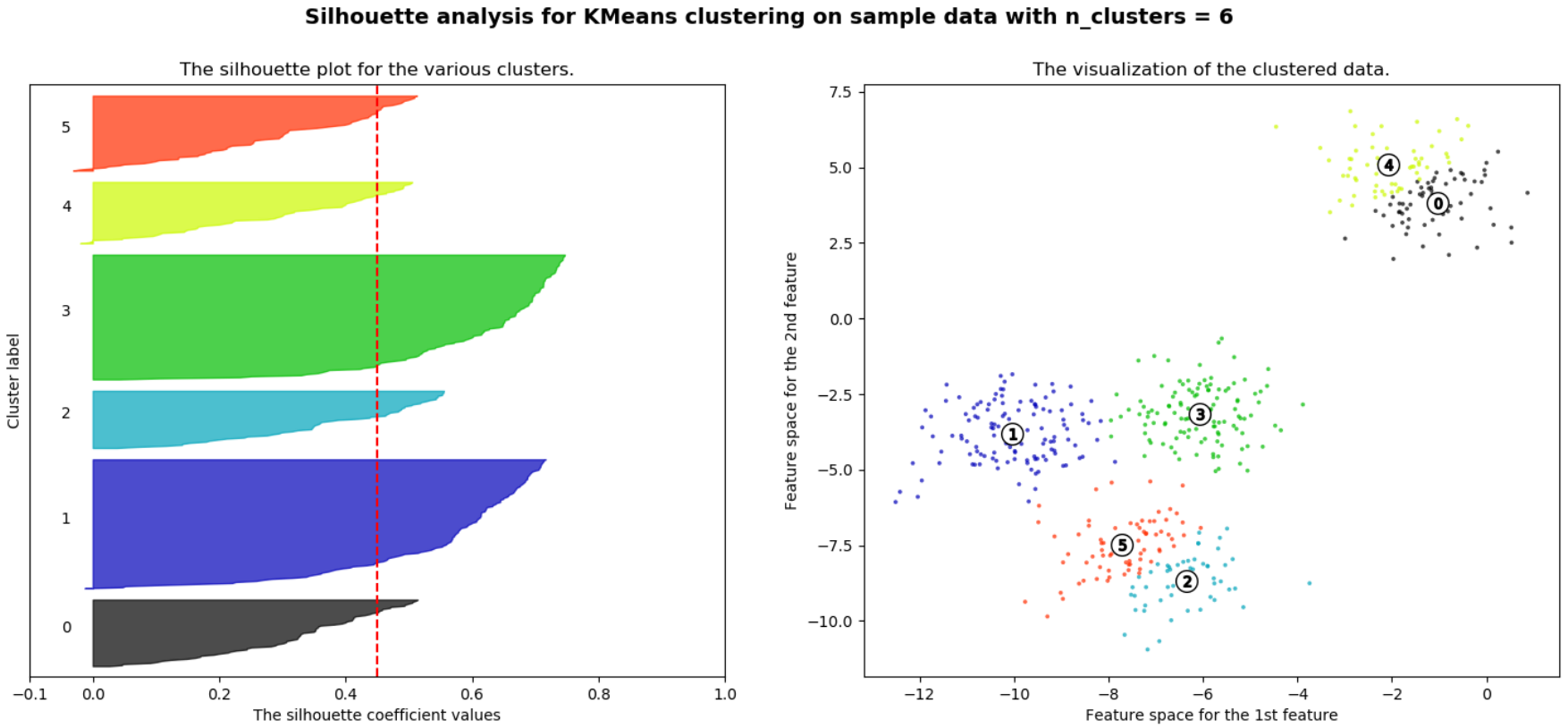
Silhouette Coefficient Example



Silhouette Coefficient Example



Silhouette Coefficient Example



Silhouette Coefficient Example

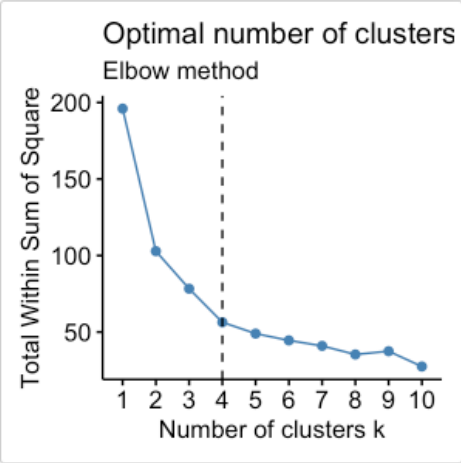
```
For n_clusters = 2 The average silhouette_score is : 0.7049787496083262
For n_clusters = 3 The average silhouette_score is : 0.5882004012129721
For n_clusters = 4 The average silhouette_score is : 0.6505186632729437
For n_clusters = 5 The average silhouette_score is : 0.563764690261 94
For n_clusters = 6 The average silhouette_score is : 0.4504666294372765
```

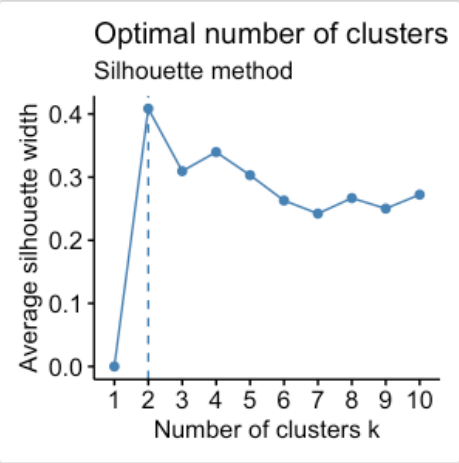
Internal Measures: Gap statistic

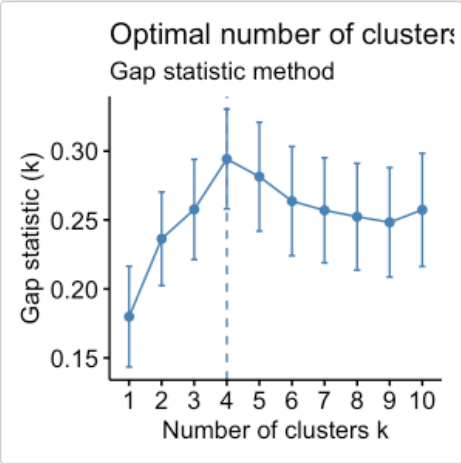
- The gap statistic compares the **total within intra-cluster variation** for different values of k **with their expected values under null reference distribution of the data**.
- The estimate of the optimal clusters will be value that maximize the gap statistic (i.e, that yields the largest gap statistic).
- This means that the clustering structure is far away from the random uniform distribution of points.

1. Cluster the observed data, varying the number of clusters from k = 1, ..., k_{max}, and compute the corresponding total within intra-cluster variation W_k.
2. Generate B reference data sets with a random uniform distribution. Cluster each of these reference data sets with varying number of clusters k = 1, ..., k_{max}, and compute the corresponding total within intra-cluster variation W_{kb}.
3. Compute the estimated gap statistic as the deviation of the observed W_k value from its expected value W_{kb} under the null hypothesis: $Gap(k) = \frac{1}{B}\sum_{b=1}^{B} log(W_{kb}^*) - log(W_k)$. Compute also the standard deviation of the statistics.
4. Choose the number of clusters as the smallest value of k such that the gap statistic is within one standard deviation of the gap at k+1: $Gap(k) \geq Gap(k+1) - s_{k+1}$.

How to decide best cluster number?







- Elbow method: 4 clusters solution suggested
- Silhouette method: 2 clusters solution suggested
- Gap statistic method: 4 clusters solution suggested

According to these observations, it's possible to define k = 4 as the optimal number of clusters in the data.

The disadvantage of elbow and average silhouette methods is that, they measure a global clustering characteristic only. A more sophisticated method is to use the gap statistic which provides a statistical procedure to formalize the elbow/silhouette heuristic in order to estimate the optimal number of clusters.

7/22/2019 https://www.datanovia.com/en/lessons/determining-the-optimal-number-of-clusters-3-must-know-methods/#gap-statistic-method

Final Comment on Cluster Validity

"The validation of clustering structures is the most difficult and frustrating part of cluster analysis.

Without a strong effort in this direction, cluster analysis will remain a black art accessible only to those true believers who have experience and great courage."

Algorithms for Clustering Data, Jain and Dubes