Cluster Validation.

W. Wang[1]

[1] Department of Mathematics
University of Houston

MATH 4323

Cluster Validation.

Cluster validation has to do with evaluating the quality of clustering algorithm results.

This is important in order to

- Avoid finding patterns in random data ($\approx$ noise);
- Compare two clustering solutions or algorithms with respect to your data. For example
  - compare K-means solutions for different K values,
  - compare K-means and hierarchical clustering algorithms.
  - compare K-means solution in the original predictor space, and K-means solution for reduced predictor space (e.g. via PCA)

Cluster Validation.

Clustering validation statistics can be categorized into 3 types:

1. Internal cluster validation, which uses the internal information of the clustering process. Includes:
   - Silhouette coefficient
   - Dunn index

2. External cluster validation: using some externally known result, such as externally provided class labels, to validate your clusters. Since we presumably know the "true" cluster number & assignments in advance, this approach focuses on selecting the right clustering algorithm for future use on similar data.

3. 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. In previous lectures, gap statistic was an example of relative cluster validation.

Internal Cluster Validation.

Internal validation measures the following aspects of the clustering solution:

- Compactness (or cohesion): how close are the objects within the same cluster.

  Example: A lower **within-cluster** variation is an indicator of a good compactness (i.e., a good clustering).

- Separation: how well-separated a cluster is from other clusters.

  Example:

  - distances between cluster centers;
  - pairwise minimum (or maximum) distances between objects in different clusters;
  - average of pairwise distances between objects in different clusters.

Internal Cluster Validation: Silhouette Coefficient.

Intuition: measure how well an observation i is clustered, by comparing

- its distance to the points from its own cluster C ($i \in C$), to
- its distance to the next closest cluster C^* ($i \notin C^*$).

Observation i is clustered well if it is

- very close to the observations from its own cluster, indicating **good compactness** within its cluster; and
- very far from observations of its next closest neighboring cluster, indication **good separation** from other clusters.

Internal Cluster Validation: Silhouette Coefficient.

Definition: For observation $i \in C$, we proceed to

1. Calculate the average dissimilarity between i and all other points j of the cluster C s.t. $i \in C$:

$$a_i = \frac{1}{|C| - 1} \sum_{j \in C \setminus \{i\}} dist(i, j)$$

2. For all other clusters C^*, s.t. $i \notin C^*$:

   2.1 Record the average dissimilarity $d(i, C)$ of i to all obs. $\in C^*$:

$$d(i, C^*) = \frac{1}{|C^*|} \sum_{j \in C^*} dist(i, j)$$

   2.2 Record the dissimilarity of i with the nearest cluster as follows:

$$b_i = min_{C^*} \; d(i, C^*)$$

3. Silhouette coefficient for observation i is

$$s_i = \frac{(b_i - a_i)}{max(a_i, b_i)}$$

Internal Cluster Validation: Silhouette Coefficient.

The silhouette coefficient is a measure of how similar an object is to its own cluster (cohesion) compared to other clusters (separation).

Silhouette coefficient $s_i, \ i = 1, \ldots, n$, has the following properties:

- $s_i \in [-1, 1]$
- Observations with a large s_i (e.g. ≈ 1) are very well clustered. Silhouette coefficients near +1 indicate that the observation is far away from the neighboring clusters.
- A small s_i (e.g. ≈ 0) means that the observation lies between two clusters. A value of 0 indicates that the sample is on or very close to the decision boundary between two neighboring clusters.
- Observations with a negative s_i are probably placed in the wrong cluster.

Silhouette Coefficient: Microarray Example.

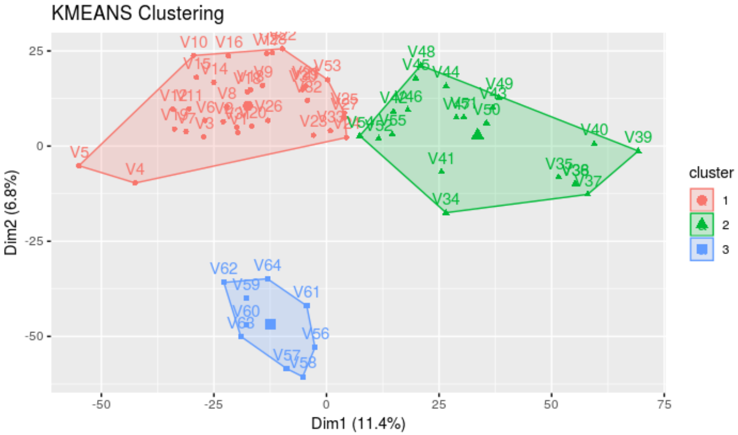
Example. Back to our human tumor microarray data, with 64 tissue samples (observations) and 6830 gene expression measurements (variables). Let's apply K-means clustering with $K = 3$ first:

```
> ec.obj <- eclust(NCI60$data,
                   FUNcluster="kmeans", k = 3,
                   nstart = 50)
> ec.obj$silinfo
$widths
    cluster neighbor     sil_width
V15       1        3  0.1426262921
V13       1        3  0.1403764388
...
V64       3        1  0.2036427394

$clus.avg.widths
[1] 0.09724312 0.07452155 0.23925233

$avg.width
[1] 0.1097576
```

Silhouette Coefficient: Microarray Example.



Question: which observations of cluster 1 appear badly clustered?

Answer: 5,4 (far from cluster center); 23-25,27,33 (close to cluster 2)

Silhouette Coefficient: Microarray Example.

Example (cont’d). Observations of each cluster are sorted by decreasing order of silhouette coefficient:

```
> ec.obj$silinfo
$widths
    cluster neighbor     sil_width
V15       1        3  0.1426262921
V13       1        3  0.1403764388
V9        1        3  0.1397725368
...
V5        1        3  0.0884049717
...
V27       1        3  0.0698969204
V24       1        2  0.0689139585
V53       1        2  0.0563860051
V25       1        2  0.0548669155
V23       1        3  0.0502590811
V33       1        3  0.0444187947
V4        1        3  0.0283507858
V36       2        3  0.1391359244
V49       2        1  0.1375178545
```

Silhouette Coefficient: Microarray Example.

Question: based on silhouette coefficients, which observations of cluster 1 are well clustered? Where are they located?

```
> ec.obj$silinfo
$widths
    cluster neighbor     sil_width
V15       1        3  0.1426262921
V13       1        3  0.1403764388
V9        1        3  0.1397725368
V14       1        3  0.1379503067
V22       1        3  0.1350643375
V10       1        3  0.1323457744
V16       1        3  0.1300257770
V28       1        3  0.1240437494
V17       1        3  0.1216965423
V12       1        3  0.1200028219
...
```

Answer: On top or near the middle of cluster 1, close to its center and far away from the other 2 clusters.

Silhouette Coefficient: Microarray Example.

Question: judging by the picture, which cluster appears to be the most dense and separated from others?

Answer: Cluster #3, forming an "island". Silhouette coefficients:

```
    cluster neighbor     sil_width
...
V58       3        1  0.2940425055
V57       3        1  0.2688703904
V60       3        1  0.2558921116
V61       3        1  0.2416405224
V63       3        1  0.2365357031
V56       3        1  0.2309498648
V59       3        1  0.2157881145
V62       3        1  0.2059089897
V64       3        1  0.2036427394
$clus.avg.widths
[1] 0.09724312 0.07452155 0.23925233
```

Observations of cluster #3 easily have the largest average silhouette coefficient (≈ 0.24) among three clusters.

Silhouette Coefficient: Microarray Example.

Question: Are there any observations with negative silhoette coefficient values? Where are they located?

Answer:

```
    cluster neighbor     sil_width
...
V55       2        1 -0.0004614699
V42       2        1 -0.0215152013
V54       2        1 -0.0319220031
```

They belong to cluster 2 and are located right at the border of clusters 2 and 1. Hence they are

- far away from many observations of cluster 2, while
- being close to plenty of observations of cluster 1,

indicative of being badly clustered.

Selection K via Largest Silhouette Coefficient.

To calculate and visualize average silhouette coefficients across K-Means solutions for multiple K:

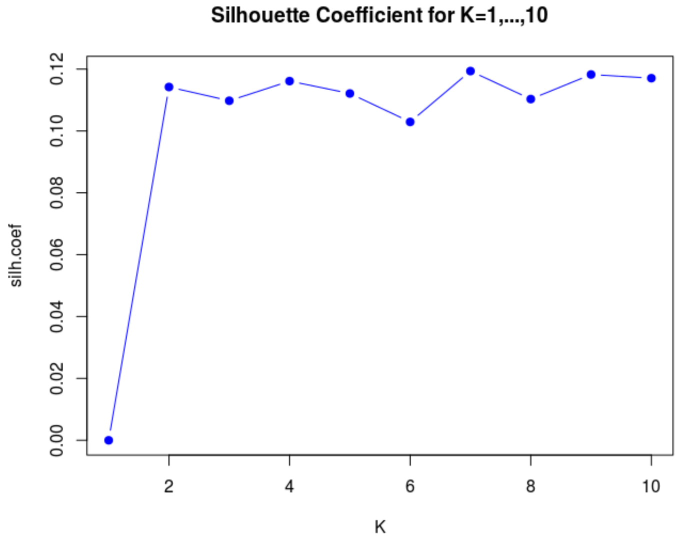
Example (cont'd).

```
k.max <- 10
silh.coef <- numeric(k.max)
for (k in 2:10){
  silh.coef[k] <- eclust(NCI60$data,
                         FUNcluster="kmeans",
                         k = k,
                         graph=0,
                         nstart = 50)$silinfo$avg.width
}

plot(silh.coef,
     type="b", pch=19, col=4)

which.max(silh.coef)
[1] 7
```

Selection *K* via Largest Silhouette Coefficient.



Selection *K* via Largest Silhouette Coefficient.

Example (cont'd). $K = 7$ has the largest silhouette coefficient:

```
> k=7
> ec.obj <- eclust(NCI60$data,
                   FUNcluster="kmeans",
                   k = k,
                   nstart = 50)
> ec.obj$silinfo
$widths
    cluster neighbor     sil_width
V58       1        2  0.2873262597
V57       1        2  0.2642944007
...
$clus.avg.widths
[1] 0.23473489 0.03437794 0.50065967 0.05709877 0.34741191
    0.12588040 0.03935969

$avg.width
[1] 0.1193519
```

Nice Silhouette Visualization.

To obtain a nicer silhouette summary visualization, apply

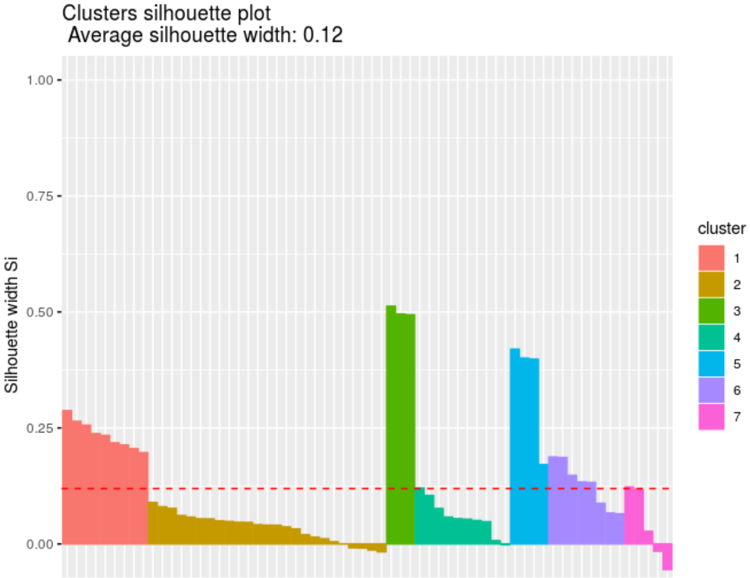
- silhouette() of *cluster* library, and
- fviz_silhouette() of *factoextra* library.

to the clustering assignment resulting from eclust() function (accessed via $cluster) and the dissimilarity matrix of your data (dist() function):

Example (cont'd).

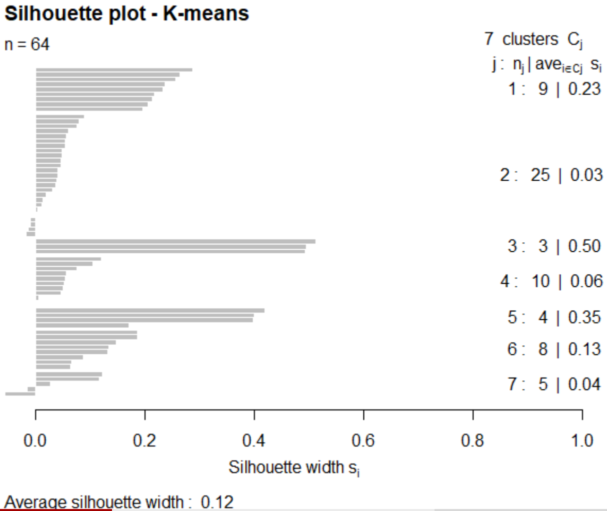
```
> library(cluster)
> sil <- silhouette(ec.obj$cluster,
                     dist(NCI60$data))
> fviz_silhouette(sil)
  cluster size ave.sil.width
1       1    9          0.23
2       2   25          0.03
3       3    3          0.50
4       4   10          0.06
5       5    4          0.35
6       6    8          0.13
7       7    5          0.04
```

Nice Silhouette Visualization.



Silhouette plot in library(cluster)

```
> plot(sil, main ="Silhouette plot - K-means") black and white but
has summary
```



Validating Arbitrary Cluster Assignments

Silhouette coefficient can be used not only to compare K-means clustering solutions for different K, but also to compare completely distinct clustering approaches.

Example. For our microarray data, we can also proceed as follows:

1. Calculate first two principal components (instead of using all 6830 variables).
2. Calculate optimal K via gap statistic, assign observations to those K clusters.

Validating Arbitrary Cluster Assignments.

Example (cont'd). Below is the code performing those steps:

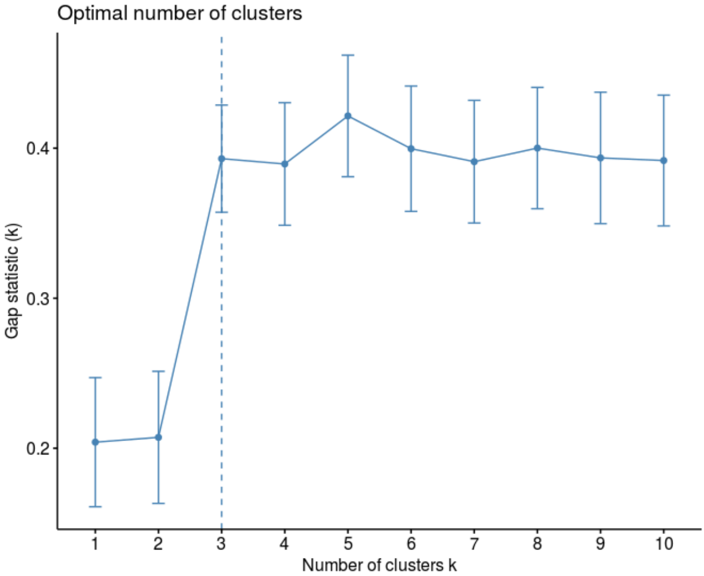
```
### Via PCA:
pca.obj <- prcomp(NCI60$data)
dim(pca.obj$x)
n.PCs <- 2
pca.data <- pca.obj$x[,c(1:n.PCs)]  # Extracting the PCs

## Gap statistic

fviz_nbclust(pca.data, kmeans, k.max=10,
             nstart = 50,
             method = "gap_stat",
             nboot = 50)
```

Judging by the gap statistic plot, $K = 3$ is the optimal # of clusters.

Validating Arbitrary Cluster Assignments.



Validating Arbitrary Cluster Assignments.

Example (cont'd). Now, let's fit that $K = 3$ clustering solution, using first two principal components:

```
km.obj <- eclust(pca.data,
                 FUNcluster = "kmeans",
                 k=3,
                 nstart=50)
```

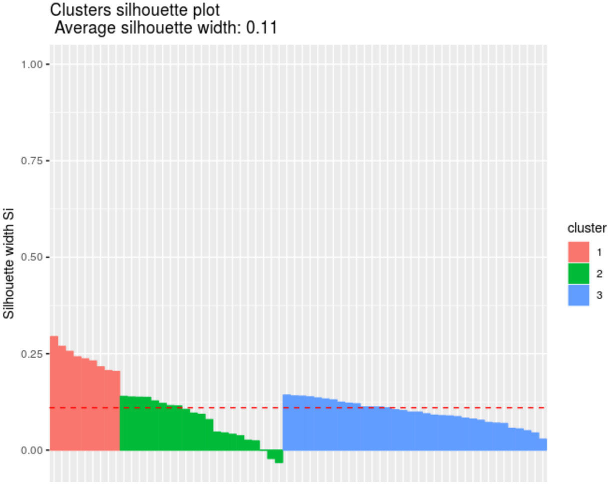
and evaluate its assignment of observations to clusters (*km.obj$cluster*) via silhouette coefficient on the original 6830 predictors (NOT on just two principal components):

```
> sil <- silhouette(km.obj$cluster, dist(NCI60$data))
> sil
      cluster neighbor     sil_width
 [1,]       3        1  0.0827639206
 [2,]       3        1  0.0772903586
 ...
> mean(sil[,3])  # Worse than the K=7 solution, with 11.9
[1] 0.1097576
```

Validating Arbitrary Cluster Assignments.

Example (cont'd). We get a worse silhouette coefficient value of 10.9, compared to 11.9 for previous $K = 7$ solution. To obtain a visualization:

```
> fviz_silhouette(sil)
```


Clusters silhouette plot
Average silhouette width: 0.11
Silhouette width Si
cluster