



BIOST 546: Machine Learning for Biomedical Big Data

Ali Shojaie

Lecture 12: Clustering - Part II
Spring 2017

Recap

- Clustering basics
- Hierarchical clustering (and bi-clustering)

Today's Class

- K-means clustering
- Model based clustering
- Spectral clustering

Hierarchical Clustering vs Partition-Based methods

- In **hierarchical clustering**, a sequence of possible clusterings is generated

Hierarchical Clustering vs Partition-Based methods

- In **hierarchical clustering**, a sequence of possible clusterings is generated
- The analyst then decides on the number of clusters (by deciding where to cut the dendrogram)

Hierarchical Clustering vs Partition-Based methods

- In **hierarchical clustering**, a sequence of possible clusterings is generated
- The analyst then decides on the number of clusters (by deciding where to cut the dendrogram)
- In **partition-based methods**, data is partitioned into a number of *a priori* given clusters

Hierarchical Clustering vs Partition-Based methods

- In **hierarchical clustering**, a sequence of possible clusterings is generated
- The analyst then decides on the number of clusters (by deciding where to cut the dendrogram)
- In **partition-based methods**, data is partitioned into a number of *a priori* given clusters
- No need for intervention by analyst; however, number of clusters needs to be specified.

K -Means Clustering

- The most popular partition-based method is K -means clustering

K -Means Clustering

- The most popular partition-based method is K -means clustering
- Motivated by a simple and intuitive mathematical problem

K -Means Clustering

- The most popular partition-based method is K -means clustering
- Motivated by a simple and intuitive mathematical problem
- Uses Euclidean distance between points

K -Means Clustering

- The most popular partition-based method is K -means clustering
- Motivated by a simple and intuitive mathematical problem
- Uses Euclidean distance between points
- Computationally efficient, and can be applied to datasets with a large number of samples (variables)

K -Means Clustering

- The most popular partition-based method is K -means clustering
- Motivated by a simple and intuitive mathematical problem
- Uses Euclidean distance between points
- Computationally efficient, and can be applied to datasets with a large number of samples (variables)
- No hierarchy among clusters, if K is changed, the cluster memberships will also change

K -Means Clustering: Problem Formulation

- Let C_1, \dots, C_K be any **partition** of the samples

K-Means Clustering: Problem Formulation

- Let C_1, \dots, C_K be any **partition** of the samples
- We want to find the an **optimal partition**, such that the *within-cluster variation is minimized*:

$$\min_{C_1, \dots, C_K} \sum_{k=1}^K WCV(C_k)$$

K-Means Clustering: Problem Formulation

- Let C_1, \dots, C_K be any **partition** of the samples
- We want to find the an **optimal partition**, such that the ***within-cluster variation is minimized***:

$$\min_{C_1, \dots, C_K} \sum_{k=1}^K WCV(C_k)$$

- There are different ways to define $WCV(C_k)$ but squared **Euclidean distance** is often used in K -means :

$$WCV(C_k) = \frac{1}{|C_k|} \sum_{i, i' \in C_k} \sum_{j=1}^p (X_{ij} - X_{i'j})^2$$

K-Means Clustering: Problem Formulation

- Let C_1, \dots, C_K be any **partition** of the samples
- We want to find the an **optimal partition**, such that the ***within-cluster variation is minimized***:

$$\min_{C_1, \dots, C_K} \sum_{k=1}^K WCV(C_k)$$

- There are different ways to define $WCV(C_k)$ but squared **Euclidean distance** is often used in K -means :

$$WCV(C_k) = \frac{1}{|C_k|} \sum_{i, i' \in C_k} \sum_{j=1}^p (X_{ij} - X_{i'j})^2$$

- So, we need to solve:

$$\min_{C_1, \dots, C_K} \sum_{k=1}^K \left\{ \frac{1}{|C_k|} \sum_{i, i' \in C_k} \sum_{j=1}^p (X_{ij} - X_{i'j})^2 \right\}$$

K -Means Clustering: Solution

- Although the optimization problem in K -means is simple and intuitive, **finding exact solutions** (global minimum) is not tractable

K -Means Clustering: Solution

- Although the optimization problem in K -means is simple and intuitive, **finding exact solutions** (global minimum) is not tractable
- However, we can **efficiently find good approximate solutions** for this problem (local minimum) using the following **algorithm**:

K -Means Clustering: Solution

- Although the optimization problem in K -means is simple and intuitive, **finding exact solutions** (global minimum) is not tractable
- However, we can **efficiently find good approximate solutions** for this problem (local minimum) using the following **algorithm**:
 - 1 Randomly assign each observation to one of K clusters.

K -Means Clustering: Solution

- Although the optimization problem in K -means is simple and intuitive, **finding exact solutions** (global minimum) is not tractable
- However, we can **efficiently find good approximate solutions** for this problem (local minimum) using the following **algorithm**:
 - 1 Randomly assign each observation to one of K clusters.
 - 2 Iterate until the cluster assignments don't change:

K -Means Clustering: Solution

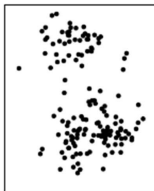
- Although the optimization problem in K -means is simple and intuitive, **finding exact solutions** (global minimum) is not tractable
- However, we can **efficiently find good approximate solutions** for this problem (local minimum) using the following **algorithm**:
 - ① Randomly assign each observation to one of K clusters.
 - ② Iterate until the cluster assignments don't change:
 - (a) For each of the K clusters, compute the cluster **centroid**, i.e. the **mean of the observations assigned to the each cluster**. This is a vector of length p (for p features).

K-Means Clustering: Solution

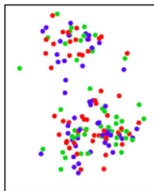
- Although the optimization problem in K -means is simple and intuitive, **finding exact solutions** (global minimum) is not tractable
- However, we can **efficiently find good approximate solutions** for this problem (local minimum) using the following **algorithm**:
 - ① Randomly assign each observation to one of K clusters.
 - ② Iterate until the cluster assignments don't change:
 - (a) For each of the K clusters, compute the cluster **centroid**, i.e. the **mean of the observations assigned to the each cluster**. This is a vector of length p (for p features).
 - (b) Assign each observation to the cluster with closest centroid (based on Euclidean distance).

K -Means Clustering: An Example with Three Clusters

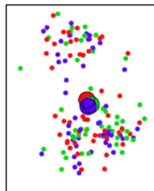
(a) Data



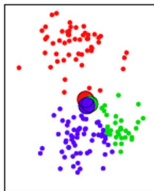
(b) Step 1



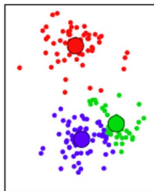
(c) Iteration 1, Step 2a



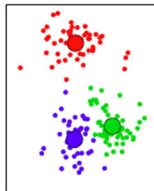
(d) Iteration 1, Step 2b



(e) Iteration 2, Step 2a



(f) Results After 10 Iterations



Properties of K -Means

- In K -means there is **no hierarchy among clusters**, if K is changed, the cluster memberships will also change

Properties of K -Means

- In K -means there is **no hierarchy among clusters**, if K is changed, the cluster memberships will also change
- K -means **works best with compact spherical clusters with comparable number of members**

Properties of K -Means

- In K -means there is **no hierarchy among clusters**, if K is changed, the cluster memberships will also change
- K -means **works best with compact spherical clusters with comparable number of members**
- Does not work well when cluster sizes are different

Properties of K -Means

- In K -means there is **no hierarchy among clusters**, if K is changed, the cluster memberships will also change
- K -means **works best with compact spherical clusters with comparable number of members**
- Does not work well when cluster sizes are different
- **Results depend on the initial cluster assignment**; may not get the best solution

Properties of K -Means

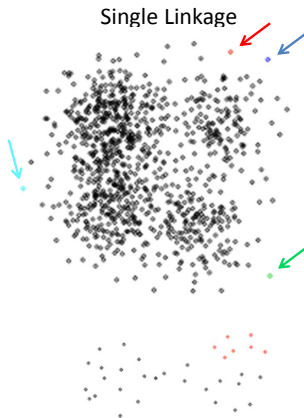
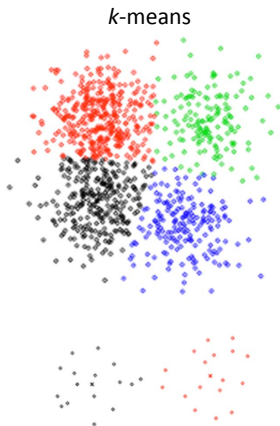
- In K -means there is **no hierarchy among clusters**, if K is changed, the cluster memberships will also change
- K -means **works best with compact spherical clusters with comparable number of members**
- Does not work well when cluster sizes are different
- **Results depend on the initial cluster assignment**; may not get the best solution
- May result in empty clusters

Properties of K -Means

- In K -means there is **no hierarchy among clusters**, if K is changed, the cluster memberships will also change
- K -means **works best with compact spherical clusters with comparable number of members**
- Does not work well when cluster sizes are different
- **Results depend on the initial cluster assignment**; may not get the best solution
- May result in empty clusters
- May result in artificially small clusters (a possible solution is to eliminate outliers)

K -Means Performance

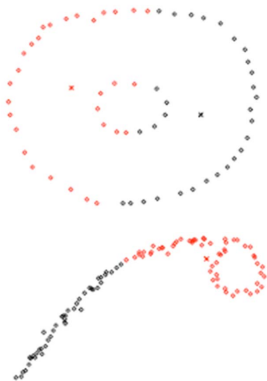
Examples, where K -means works well



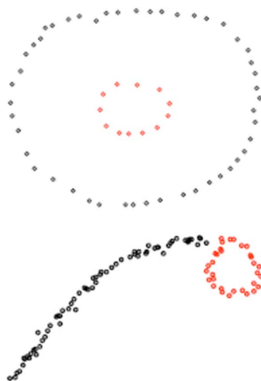
K-Means Performance

Examples, where **K-means does not work well**

k-means



Single Linkage



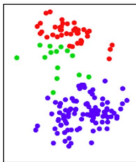
Improving K -Means

A practical solution for obtaining better solutions for K -means is to **run the algorithm with a number of random starting points**, and then choose the solution with lowest objective function.

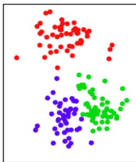
Improving K -Means

A practical solution for obtaining better solutions for K -means is to **run the algorithm with a number of random starting points**, and then choose the solution with lowest objective function.

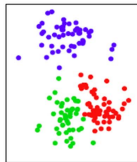
Objective is 320.9



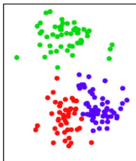
Objective is 235.8



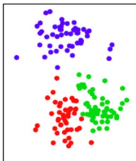
Objective is 235.8



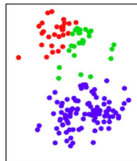
Objective is 235.8



Objective is 235.8



Objective is 310.9



Evaluating the Quality of a Clustering

- **Cluster homogeneity**

Evaluating the Quality of a Clustering

- **Cluster homogeneity**

- ▶ Within sum of squares (WSS)
- ▶ For each **object** the “error” is the **distance to its cluster centroid**:

$$WSS = \sum_{k=1}^K \sum_{i \in C_k} d^2(m_k; X_i)$$

- ▶ Given two clustering solutions, the one with smaller WSS is preferred
- ▶ **WSS usually decreases as K increases**

Evaluating the Quality of a Clustering

● Cluster homogeneity

- ▶ Within sum of squares (WSS)
- ▶ For each **object** the “error” is the **distance to its cluster centroid**:

$$WSS = \sum_{k=1}^K \sum_{i \in C_k} d^2(m_k; X_i)$$

- ▶ Given two clustering solutions, the one with smaller WSS is preferred
- ▶ **WSS usually decreases as K increases**

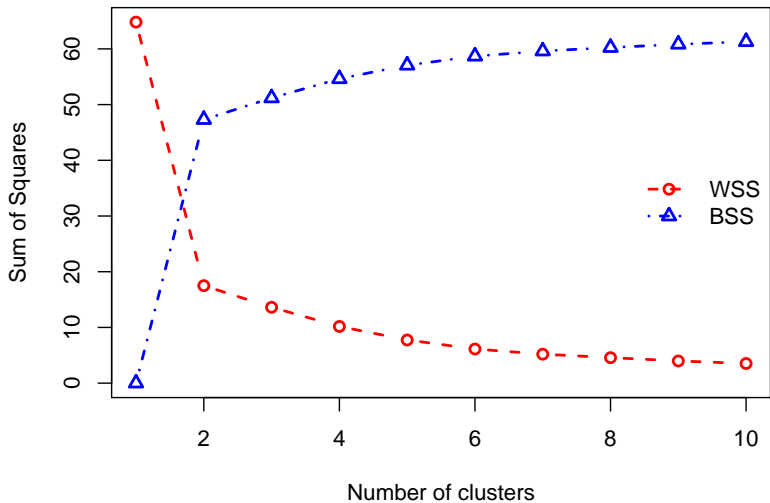
● Cluster separation

- ▶ Between sum of squares (BSS)
- ▶ For each **cluster** the “error” is the **distance between the cluster centroid and the *grand mean***:

$$BSS = \sum_{k=1}^K d^2(m_k; m)$$

BSS vs WSS

For K -means , $d^2(X_i; m_k) = \sum_{j=1}^p (X_{ij} - m_{kj})^2$ (Euclidean dist.)



Choosing K (Number of Clusters)

- Look at the values of within cluster dissimilarity (e.g. WSS) for different K .

Choosing K (Number of Clusters)

- Look at the values of within cluster dissimilarity (e.g. WSS) for different K .
- In general, WSS decreases as K increases.

Choosing K (Number of Clusters)

- Look at the values of within cluster dissimilarity (e.g. WSS) for different K .
- In general, WSS decreases as K increases.
- Suppose there is K^* true clusters

Choosing K (Number of Clusters)

- Look at the values of within cluster dissimilarity (e.g. WSS) for different K .
- In general, WSS decreases as K increases.
- Suppose there is K^* true clusters
- If $K < K^*$, increasing K will improve WSS significantly

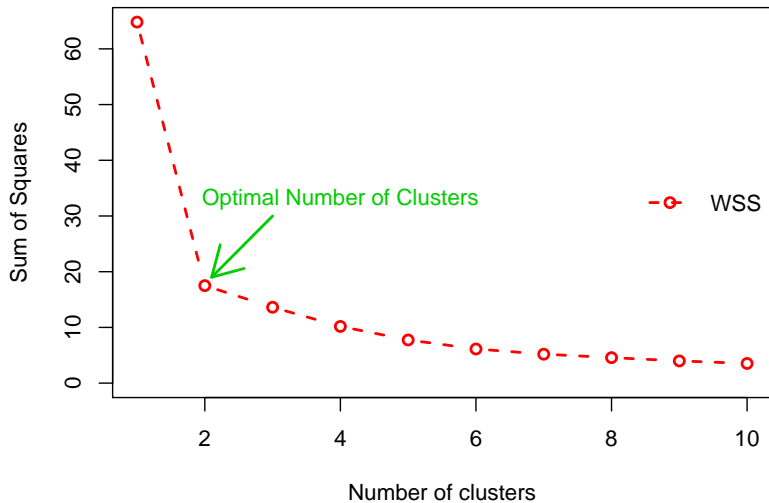
Choosing K (Number of Clusters)

- Look at the values of within cluster dissimilarity (e.g. WSS) for different K .
- In general, WSS decreases as K increases.
- Suppose there is K^* true clusters
- If $K < K^*$, increasing K will improve WSS significantly
- If $K > K^*$, increasing K will improve WSS slightly

Choosing K (Number of Clusters)

- Look at the values of within cluster dissimilarity (e.g. WSS) for different K .
- In general, WSS decreases as K increases.
- Suppose there is K^* true clusters
- If $K < K^*$, increasing K will improve WSS significantly
- If $K > K^*$, increasing K will improve WSS slightly
- Therefore, need to look for an **elbow** in the plot of WSS over K

WSS for K -Means



Other Methods for Choosing K (Number of Clusters)

- **Gap Statistics** (Tibshirani et al, 2001): Compares the curve of $\log WSS$ to the curve obtained from data uniformly distributed (i.e. no clusters), and estimates the optimal number of clusters to be the place where the gap between the two curves is largest.

Other Methods for Choosing K (Number of Clusters)

- **Gap Statistics** (Tibshirani et al, 2001): Compares the curve of $\log WSS$ to the curve obtained from data uniformly distributed (i.e. no clusters), and estimates the optimal number of clusters to be the place where the gap between the two curves is largest.
- The **Silhouette Coefficient** (Rousseeuw, 1986): Combines homogeneity and separation, can find the best number of clusters by minimizing this coefficient over range of values of K .

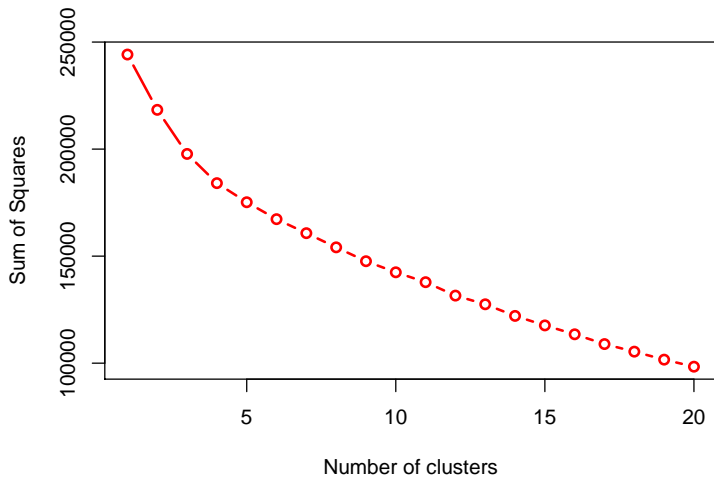
Example: *K*-Means Clustering on NCI60 Data

```
## K-means clustering for the NCI60 data
mydata <- read.table('nci.data')
mydata <- t(mydata)
mydata <- scale(mydata, center=F, scale=T)

Kmax <- 20                                #maximum number of clusters
Nstrt <- 25                                #Number of starting partitions
wss <- numeric(K)
bss <- numeric(K)
tss <- numeric(K)
for(k in 1:Kmax){
  cl <- kmeans(dat, k, nstart=Nstrt)
  tss[k] <- cl$totss
  wss[k] <- cl$tot.withinss
  bss[k] <- cl$betweenss
}

## Plot the results
par(mar=c(4.5,4.5,1,1))
plot(1:Kmax, wss, type='b', lty=1, lwd=2, col=2,
     ylab='Sum of Squares', xlab='Number of clusters')
```

How Many Clusters?



Exercise

- Repeat the analysis without normalizing the data. What do you see?
- Repeat the clustering with different values of $n_{start} \in \{1, 5, 10, 50\}$
- How do the results change?
- What is the optimal number of clusters?
- Compare the optimal number of clusters with what you observed using hierarchical clustering.

Model-Based Clustering

- Hierarchical or K -means clustering assign each data point to one of the K clusters

Model-Based Clustering

- Hierarchical or K -means clustering assign each data point to one of the K clusters
- This makes these methods **sensitive to outliers and the compactness of the clusters**

Model-Based Clustering

- Hierarchical or K -means clustering assign each data point to one of the K clusters
- This makes these methods **sensitive to outliers and the compactness of the clusters**
- In many applications, some points fall between 2 clusters

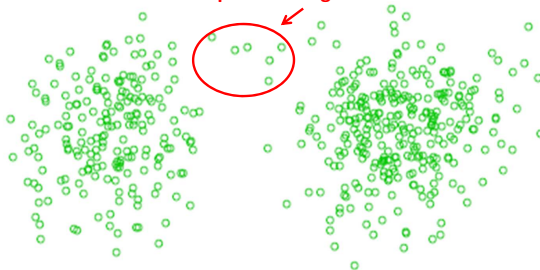
Model-Based Clustering

- Hierarchical or K -means clustering assign each data point to one of the K clusters
- This makes these methods **sensitive to outliers and the compactness of the clusters**
- In many applications, some points fall between 2 clusters
- Need to **allow for soft cluster memberships**

Model-Based Clustering

- Hierarchical or K -means clustering assign each data point to one of the K clusters
- This makes these methods **sensitive to outliers and the compactness of the clusters**
- In many applications, some points fall between 2 clusters
- Need to **allow for soft cluster memberships**

Which cluster do these point belong to?



Model-Based Clustering

- A popular approach for model-based clustering is **Gaussian mixture models**

Model-Based Clustering

- A popular approach for model-based clustering is **Gaussian mixture models**
- It is assumed that data is generated from

$$\sum_{k=1}^K \pi_k N(\mu_k, \Sigma_k)$$

Model-Based Clustering

- A popular approach for model-based clustering is **Gaussian mixture models**
- It is assumed that data is generated from

$$\sum_{k=1}^K \pi_k N(\mu_k, \Sigma_k)$$

- Here π_k gives the **probability** that data is generated from k th distribution

Model-Based Clustering

- A popular approach for model-based clustering is **Gaussian mixture models**
- It is assumed that data is generated from

$$\sum_{k=1}^K \pi_k N(\mu_k, \Sigma_k)$$

- Here π_k gives the **probability** that data is generated from k th distribution
- Often it is assumed that $\Sigma_k = \sigma_k^2 I$

Model-Based Clustering

- A popular approach for model-based clustering is **Gaussian mixture models**
- It is assumed that data is generated from

$$\sum_{k=1}^K \pi_k N(\mu_k, \Sigma_k)$$

- Here π_k gives the **probability** that data is generated from k th distribution
- Often it is assumed that $\Sigma_k = \sigma_k^2 I$
- As σ_k^2 becomes smaller, this becomes more similar to K -means clustering

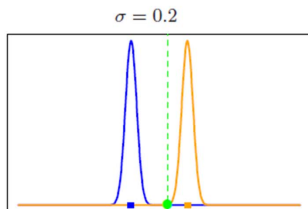
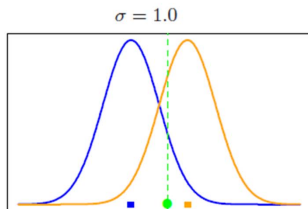
Model-Based Clustering

- A popular approach for model-based clustering is **Gaussian mixture models**
- It is assumed that data is generated from

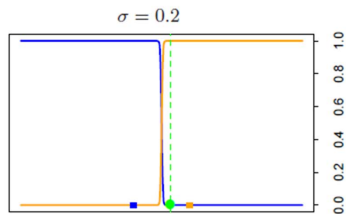
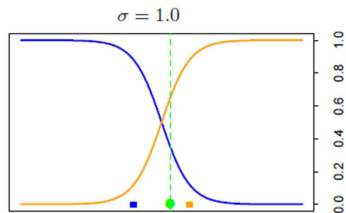
$$\sum_{k=1}^K \pi_k N(\mu_k, \Sigma_k)$$

- Here π_k gives the **probability** that data is generated from k th distribution
- Often it is assumed that $\Sigma_k = \sigma_k^2 I$
- As σ_k^2 becomes smaller, this becomes more similar to K -means clustering
- Function `Mclust` in the R-package `mclust` gives an implementation of this method

Model-Based Clustering



Gaussian Mixtures



Relative Densities

Spectral Clustering

Spectral Clustering

- Traditional clustering methods (K-means, model-based etc) use spherical/elliptical metrics, and may not work with non-convex clusters

Spectral Clustering

- Traditional clustering methods (K-means, model-based etc) use spherical/elliptical metrics, and may not work with non-convex clusters
- Spectral clustering uses the top M **eigenvectors** of the **affinity matrix** (or similarity matrix)

Spectral Clustering

- Traditional clustering methods (K-means, model-based etc) use spherical/elliptical metrics, and may not work with non-convex clusters
- Spectral clustering uses the top M **eigenvectors** of the **affinity matrix** (or similarity matrix)
- Can detect non-spherical clusters, or clusters with different shapes and sizes

Spectral Clustering

- Traditional clustering methods (K-means, model-based etc) use spherical/elliptical metrics, and may not work with non-convex clusters
- Spectral clustering uses the top M **eigenvectors** of the **affinity matrix** (or similarity matrix)
- Can detect non-spherical clusters, or clusters with different shapes and sizes
- Can work with any similarity matrix, and is hence more flexible

Spectral Clustering

- Traditional clustering methods (K-means, model-based etc) use spherical/elliptical metrics, and may not work with non-convex clusters
- Spectral clustering uses the top M **eigenvectors** of the **affinity matrix** (or similarity matrix)
- Can detect non-spherical clusters, or clusters with different shapes and sizes
- Can work with any similarity matrix, and is hence more flexible
- Similar ideas are used in the **PageRank** algorithm used by **Google**

Spectral Clustering: Details

Spectral Clustering: Details

- Start with a $n \times n$ matrix of pairwise similarities $s_{ii'} \geq 0$

Spectral Clustering: Details

- Start with a $n \times n$ matrix of **pairwise similarities** $s_{ii'} \geq 0$
- Represent observations as an undirected **similarity graph** $G = (V, E)$:

Spectral Clustering: Details

- Start with a $n \times n$ matrix of **pairwise similarities** $s_{ii'} \geq 0$
- Represent observations as an undirected **similarity graph** $G = (V, E)$:
 - ▶ **vertices**/nodes of G are observations;
 - ▶ **edges** of G show positive similarities;
 - ▶ in other words, the similarity matrix is the **adjacency matrix** of the graph;
 - ▶ ideally the similarity graphs should represent the local neighborhood relationships.

Spectral Clustering: Details

Spectral Clustering: Details

- In spectral clustering, the problem of finding clusters is translated to a **graph partitioning** problem:

Spectral Clustering: Details

- In spectral clustering, the problem of finding clusters is translated to a **graph partitioning** problem:
 - ▶ want to find groups of nodes that are highly connected to each other, but less connected to other nodes;

Spectral Clustering: Details

- In spectral clustering, the problem of finding clusters is translated to a **graph partitioning** problem:
 - ▶ want to find groups of nodes that are highly connected to each other, but less connected to other nodes;
 - ▶ let D be a diagonal matrix of **degrees**: $d_{ii} = \sum_{i' \neq i} s_{ii'}$

Spectral Clustering: Details

- In spectral clustering, the problem of finding clusters is translated to a **graph partitioning** problem:
 - ▶ want to find groups of nodes that are highly connected to each other, but less connected to other nodes;
 - ▶ let D be a diagonal matrix of **degrees**: $d_{ii} = \sum_{i' \neq i} s_{ii'}$
 - ▶ optimal graph partitions can be found using the **eigenvectors** of the **graph Laplacian**: $L = D - S$;

Spectral Clustering: Details

- In spectral clustering, the problem of finding clusters is translated to a **graph partitioning** problem:
 - ▶ want to find groups of nodes that are highly connected to each other, but less connected to other nodes;
 - ▶ let D be a diagonal matrix of **degrees**: $d_{ii} = \sum_{i' \neq i} s_{ii'}$
 - ▶ optimal graph partitions can be found using the **eigenvectors** of the **graph Laplacian**: $L = D - S$;
 - ▶ if the graph has e.g. K **connected components**, then L has (exactly) K eigenvectors $v_k, k = 1, \dots, K$, with eigenvalue 0;

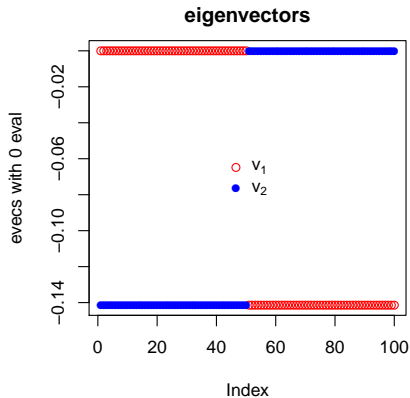
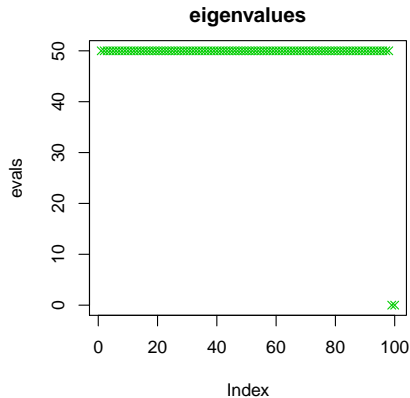
Spectral Clustering: Details

- In spectral clustering, the problem of finding clusters is translated to a **graph partitioning** problem:
 - ▶ want to find groups of nodes that are highly connected to each other, but less connected to other nodes;
 - ▶ let D be a diagonal matrix of **degrees**: $d_{ii} = \sum_{i' \neq i} s_{ii'}$
 - ▶ optimal graph partitions can be found using the **eigenvectors** of the **graph Laplacian**: $L = D - S$;
 - ▶ if the graph has e.g. K **connected components**, then L has (exactly) K eigenvectors $v_k, k = 1, \dots, K$, with eigenvalue 0;
 - ▶ the non-zero values in each v_k identify one of the connected components.

Spectral Clustering: Details

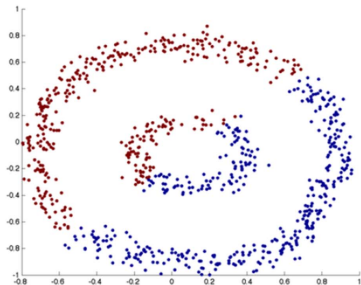
- In spectral clustering, the problem of finding clusters is translated to a **graph partitioning** problem:
 - ▶ want to find groups of nodes that are highly connected to each other, but less connected to other nodes;
 - ▶ let D be a diagonal matrix of **degrees**: $d_{ii} = \sum_{i' \neq i} s_{ii'}$
 - ▶ optimal graph partitions can be found using the **eigenvectors** of the **graph Laplacian**: $L = D - S$;
 - ▶ if the graph has e.g. K **connected components**, then L has (exactly) K eigenvectors $v_k, k = 1, \dots, K$, with eigenvalue 0;
 - ▶ the non-zero values in each v_k identify one of the connected components.
- In practice, we find *approximate* partitioning of the graph by selecting K eigenvectors, and dropping the nodes corresponding to “small” entries

Spectral Clustering

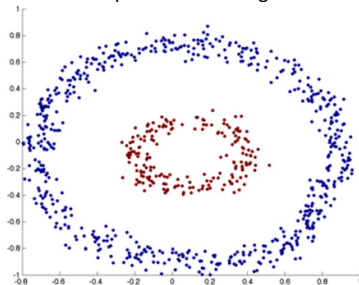


Spectral Clustering

K-means



Spectral Clustering



Spectral clustering and Principal Components

Spectral clustering and Principal Components

- Consider a **fully connected graph**, with **Euclidean distances** between each pair of points

Spectral clustering and Principal Components

- Consider a **fully connected graph**, with **Euclidean distances** between each pair of points
- Then, spectral clustering is equivalent to clustering the observations after **projecting them into the PC-space**

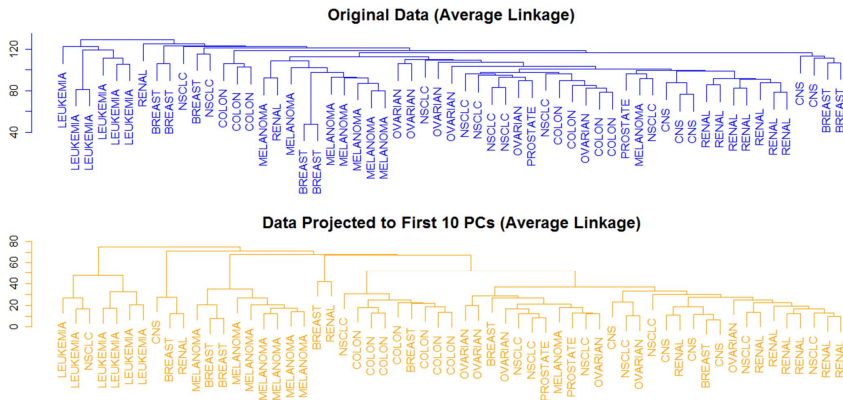
Spectral clustering and Principal Components

- Consider a **fully connected graph**, with **Euclidean distances** between each pair of points
- Then, spectral clustering is equivalent to clustering the observations after **projecting them into the PC-space**
- Basically, in this case, the affinity matrix is the sample covariance matrix, and eigenvectors are PCs

Spectral clustering and Principal Components

- Consider a **fully connected graph**, with **Euclidean distances** between each pair of points
- Then, spectral clustering is equivalent to clustering the observations after **projecting them into the PC-space**
- Basically, in this case, the affinity matrix is the sample covariance matrix, and eigenvectors are PCs
- The **type of similarity graph** (connected or nearest neighbors) as well as **number of eigenvectors** are important choices in spectral clustering.

NCI60 Data, again



The second approach is basically a version of spectral clustering, using the correlation matrix

Caveats of Cluster Analysis (Dihaeseleer, 2005)

Although many clustering algorithms are available, this is still a challenging problem:

Caveats of Cluster Analysis (Dihaeseleer, 2005)

Although many clustering algorithms are available, this is still a challenging problem:

- There is **no one-size-fits-all solution to clustering**, or even a consensus of what a “good” clustering should look like.

Caveats of Cluster Analysis (Dihaeseleer, 2005)

Although many clustering algorithms are available, this is still a challenging problem:

- There is **no one-size-fits-all solution to clustering**, or even a consensus of what a “good” clustering should look like.
- There is **no single best criterion for obtaining a partition** because no precise and workable definition of “cluster” exists.

Caveats of Cluster Analysis (Dihaeseleer, 2005)

Although many clustering algorithms are available, this is still a challenging problem:

- There is **no one-size-fits-all solution to clustering**, or even a consensus of what a “good” clustering should look like.
- There is **no single best criterion for obtaining a partition** because no precise and workable definition of “cluster” exists.
- Clusters can be of any **arbitrary shapes and sizes in a high dimensional data**.

Caveats of Cluster Analysis (Dihaeseleer, 2005)

Although many clustering algorithms are available, this is still a challenging problem:

- There is **no one-size-fits-all solution to clustering**, or even a consensus of what a “good” clustering should look like.
- There is **no single best criterion for obtaining a partition** because no precise and workable definition of “cluster” exists.
- Clusters can be of any **arbitrary shapes and sizes in a high dimensional data**.
- Each clustering criterion imposes a certain structure on the data, and if the data happen to conform to the requirements of a particular criterion, the true clusters are recovered.

Caveats of Cluster Analysis (Dihaeseleer, 2005)

Although many clustering algorithms are available, this is still a challenging problem:

- There is **no one-size-fits-all solution to clustering**, or even a consensus of what a “good” clustering should look like.
- There is **no single best criterion for obtaining a partition** because no precise and workable definition of “cluster” exists.
- Clusters can be of any **arbitrary shapes and sizes in a high dimensional data**.
- Each clustering criterion imposes a certain structure on the data, and if the data happen to conform to the requirements of a particular criterion, the true clusters are recovered.
- It is **very hard to evaluate how well a clustering algorithm performs** on typical omics data sets.

Caveats of Cluster Analysis

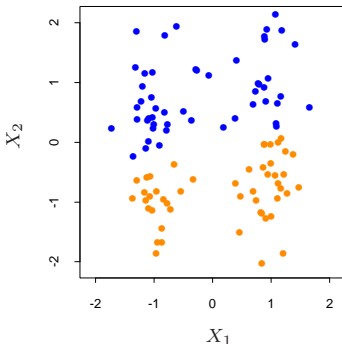
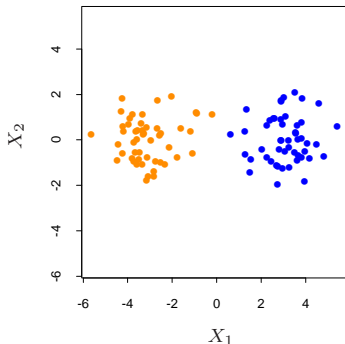
- Also, note that centering and scaling will significantly affect the results of cluster analysis

Caveats of Cluster Analysis

- Also, note that centering and scaling will significantly affect the results of cluster analysis
 - ▶ Left: K means clustering on the **original** data;
 - ▶ Right: K means clustering on the **scaled** data.

Caveats of Cluster Analysis

- Also, note that centering and scaling will significantly affect the results of cluster analysis
 - Left: K means clustering on the **original** data;
 - Right: K means clustering on the **scaled** data.



Final Remarks on Clustering

- Clustering is a very **useful for gaining insight** in high dimensional omics data

Final Remarks on Clustering

- Clustering is a very **useful for gaining insight** in high dimensional omics data
- Different clustering results for the same data, using different methods/distance matrices

Final Remarks on Clustering

- Clustering is a very **useful for gaining insight** in high dimensional omics data
- Different clustering results for the same data, using different methods/distance matrices
- No formal way to verify the results of clustering

Final Remarks on Clustering

- Clustering is a very **useful for gaining insight** in high dimensional omics data
- Different clustering results for the same data, using different methods/distance matrices
- No formal way to verify the results of clustering
- **Determining number of clusters is a challenging problem**

Final Remarks on Clustering

- Clustering is a very **useful for gaining insight** in high dimensional omics data
- Different clustering results for the same data, using different methods/distance matrices
- No formal way to verify the results of clustering
- **Determining number of clusters is a challenging problem**
- Results of clustering may vary for normalized observations

Final Remarks on Clustering

- Clustering is a very **useful for gaining insight** in high dimensional omics data
- Different clustering results for the same data, using different methods/distance matrices
- No formal way to verify the results of clustering
- **Determining number of clusters is a challenging problem**
- Results of clustering may vary for normalized observations
- **Best practice** is to **try different methods/distances and use the results that are more consistent** across methods/distances

Final Remarks on Clustering

- Clustering is a very **useful for gaining insight** in high dimensional omics data
- Different clustering results for the same data, using different methods/distance matrices
- No formal way to verify the results of clustering
- **Determining number of clusters is a challenging problem**
- Results of clustering may vary for normalized observations
- **Best practice** is to **try different methods/distances and use the results that are more consistent** across methods/distances
- Clustering is a **data exploration step**, and its results should be interpreted that way; cannot make strong claims...