

Clustering

Victor Kitov

Table of Contents

- 1 K-means
- 2 Hierarchical clustering
- 3 Spectral clustering

K-means algorithm

- Suppose we want to cluster our data into g clusters.
- Cluster i has a center μ_i , $i=1,2,\dots,g$.
- Consider the task of minimizing

$$\sum_{n=1}^N \rho(x_n, \mu_{z_n})^2 \rightarrow \min_{z_1, \dots, z_N} \quad (1)$$

where $z_i \in \{1, 2, \dots, g\}$ is cluster assignment for x_i .

- Direct optimization requires full search and is impractical.
- K-means is a suboptimal algorithm for optimizing (1).

K-means algorithm

Initialize $\mu_j, j = 1, 2, \dots, g$.

repeat while stop condition not satisfied:

for $i = 1, 2, \dots, N$:

find cluster number of x_i :

$$z_i = \arg \min_{j \in \{1, 2, \dots, g\}} \|x_i - \mu_j\|$$

for $j = 1, 2, \dots, g$:

$$\mu_j = \frac{1}{\sum_{n=1}^N \mathbb{I}[z_n = j]} \sum_{n=1}^N \mathbb{I}[z_n = j] x_i$$

Possible stop conditions:

- cluster assignments z_1, \dots, z_N stop to change (typical)
- maximum number of iterations reached
- cluster means $\{\mu_i, i = 1, 2, \dots, g\}$ stop changing significantly

Dynamic K-means algorithm

Initialize $\mu_j, j = 1, 2, \dots, g, z_i = 0, i = 1, 2, \dots, N$

repeat while stop condition not satisfied:

for $i = 1, 2, \dots, N$:

find cluster number of x_i :

$$z'_i = \arg \min_{j \in \{1, 2, \dots, g\}} \|x_i - \mu_j\|$$

if $z'_i \neq z_i$:

recalculate cluster means μ_{z_i} and $\mu_{z'_i}$:

$$\mu_{z_i} = \frac{1}{\sum_{n=1}^N \mathbb{I}[z'_n = z_i]} \sum_{n=1}^N \mathbb{I}[z_n = z_i] x_i$$

$$\mu_{z'_i} = \frac{1}{\sum_{n=1}^N \mathbb{I}[z'_n = z'_i]} \sum_{n=1}^N \mathbb{I}[z'_n = z'_i] x_i$$

$$z_i = z'_i$$

Converges in less iterations, situation when no objects correspond to some cluster is impossible.

Initialization of cluster centers

- 1 We can initialize $\{\mu_i, i = 1, 2, \dots, g\}$ with g randomly chosen measurements without replacement (typical)
- 2 Alternatively we can initialize $\{\mu_i, i = 1, 2, \dots, g\}$ with most distant set of points:

Estimate $\mu = \frac{1}{N} \sum_{i=1}^N x_i$.

set $\mu_1 = \operatorname{argmax}_{x \in \{x_1, \dots, x_N\}} \rho(\mu, x)$

set $\mu_2 = \operatorname{argmax}_{x \in \{x_1, \dots, x_N\}} \{\rho(\mu_1, x)\}$

set $\mu_3 = \operatorname{argmax}_{x \in \{x_1, \dots, x_N\}} \{\rho(\mu_1, x) + \rho(\mu_2, x)\}$

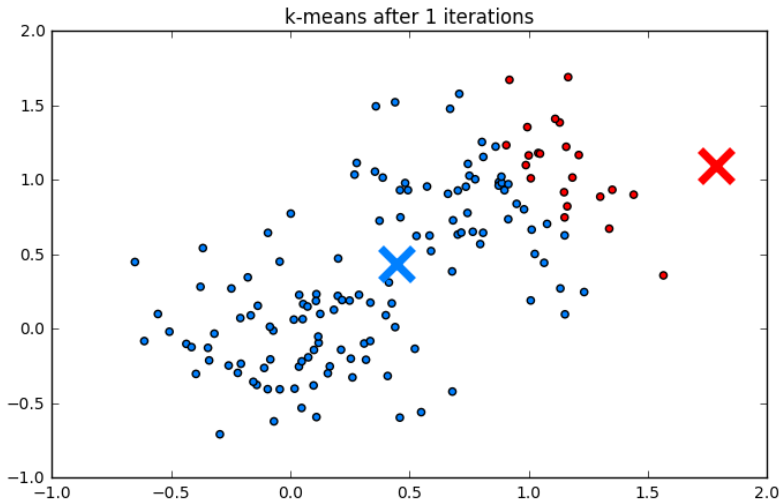
.....

set $\mu_g = \operatorname{argmax}_{x \in \{x_1, \dots, x_N\}} \{\sum_{i=1}^{g-1} \rho(\mu_i, x)\}$

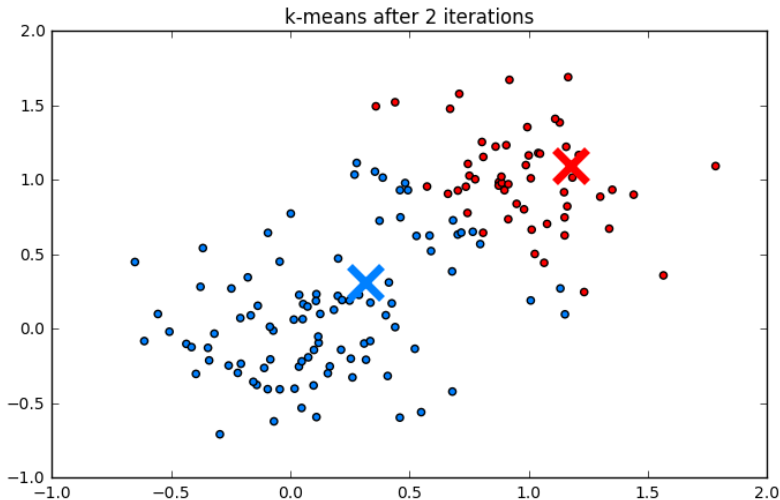
K-means properties

- Only local optimum is found
- Results depends on initialization
 - It is common to run algorithm multiple times with different initializations and then select the result minimizing criterion in (1).
- *Complexity: $O(NDgI)$, where g is the number of clusters and I is the number of iterations. Why?*
 - If clusters exist, algorithm converges with few iterations and complexity is $O(NDg)$

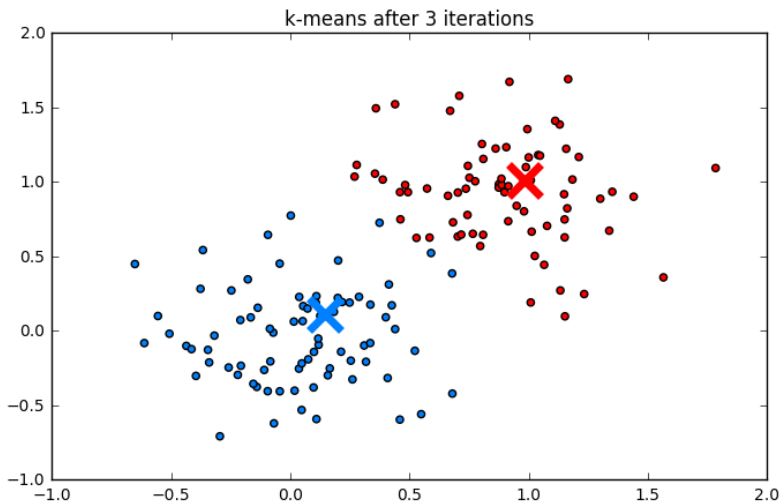
Example of K-means



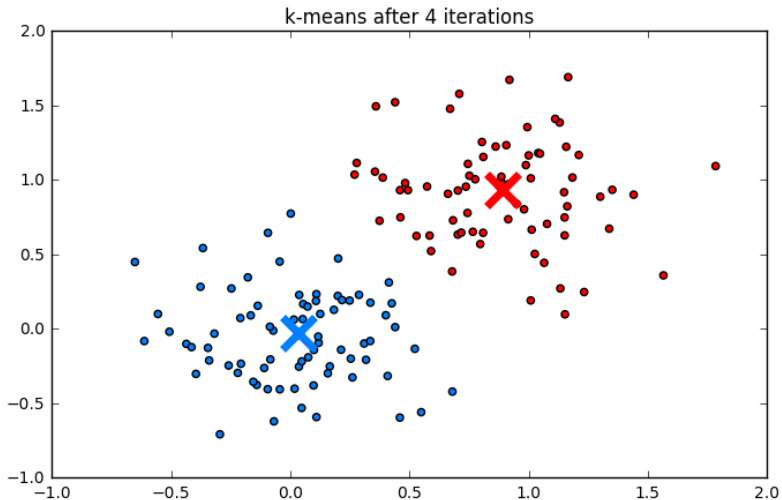
Example of K-means



Example of K-means



Example of K-means



Gotchas

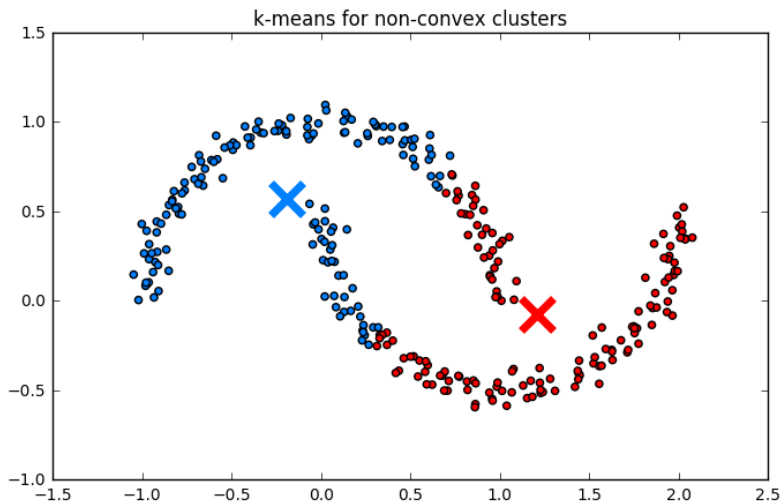
- K-means assumes that clusters are convex:

K-means clustering on the digits dataset (PCA-reduced data)
Centroids are marked with white cross

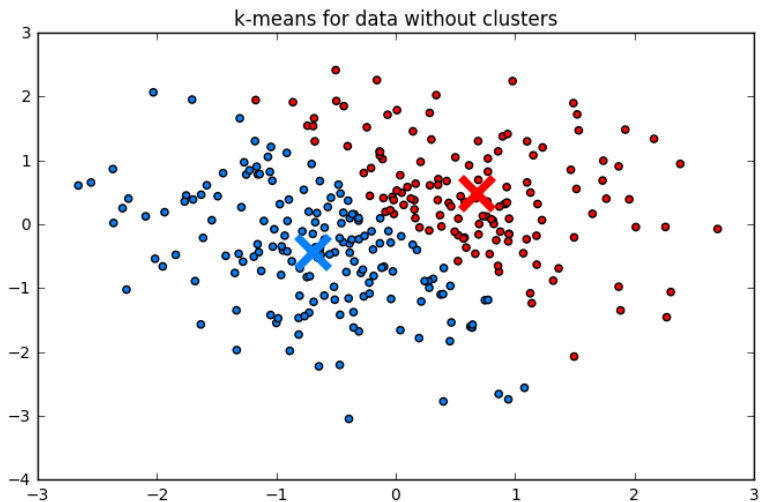


- It always finds clusters even if none actually exist
 - need to control cluster quality metrics

K-means for non-convex clusters



K-means for data without clusters



K-means and EM algorithm

Initialize $\mu_j, j = 1, 2, \dots, g$.

repeat while stop condition not satisfied:

for $i = 1, 2, \dots, N$:

 find cluster number of x_i :

$$z_i = \arg \min_{j \in \{1, 2, \dots, g\}} \|x_i - \mu_j\|$$

for $j = 1, 2, \dots, g$:

$$\mu_j = \frac{1}{\sum_{n=1}^N \mathbb{I}[z_n = j]} \sum_{n=1}^N \mathbb{I}[z_n = j] x_n$$

- K-means is EM-algorithm when:

K-means and EM algorithm

Initialize $\mu_j, j = 1, 2, \dots, g$.

repeat while stop condition not satisfied:

for $i = 1, 2, \dots, N$:

 find cluster number of x_i :

$$z_i = \arg \min_{j \in \{1, 2, \dots, g\}} \|x_i - \mu_j\|$$

for $j = 1, 2, \dots, g$:

$$\mu_j = \frac{1}{\sum_{n=1}^N \mathbb{I}[z_n = j]} \sum_{n=1}^N \mathbb{I}[z_n = j] x_n$$

- K-means is EM-algorithm when:
 - applied to Gaussians
 - with equal priors
 - with unity covariance matrices
 - with hard clustering

Table of Contents

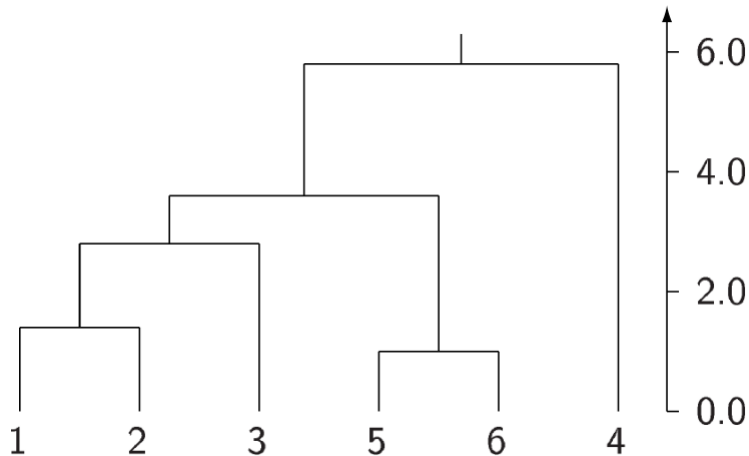
- 1 K-means
- 2 Hierarchical clustering
- 3 Spectral clustering

Hierarchical clustering

Hierarchical clustering may be:

- top-down
 - hierarchical K-means
- bottom-up
 - agglomerative clustering

Agglomerative clustering



Agglomerative clustering - distances

- Consider clusters $A = \{x_{i_1}, x_{i_2}, \dots\}$ and $B = \{x_{j_1}, x_{j_2}, \dots\}$.
- We can define the following natural distances

- nearest neighbour (or single link)

$$\rho(A, B) = \min_{a \in A, b \in B} \rho(a, b)$$

- furthest neighbour (or complete-link)

$$\rho(A, B) = \max_{a \in A, b \in B} \rho(a, b)$$

- group average link

$$\rho(A, B) = \text{mean}_{a \in A, b \in B} \rho(a, b)$$

- centroid distance ($\mu_U = \frac{1}{|U|} \sum_{x \in U} x$)

$$\rho(A, B) = \rho(\mu_A, \mu_B)$$

- median distance ($m_U = \text{median}_{x \in U} \{x\}$)

$$\rho(A, B) = \rho(m_a, m_b)$$

Agglomerative clustering - distance properties

- *Suppose we modify distance $\rho(x, x')$ with monotone transformation F : $\rho'(x, x') = F(\rho(x, x'))$. Which of the cluster distances will not be affected by this change?*
- Lance-Williams recurrence formula:
 - $\rho(A \cup B, C)$ can be computed in $O(1)$ time using $\rho(A, C)$, $\rho(B, C)$ and $\rho(A, B)$

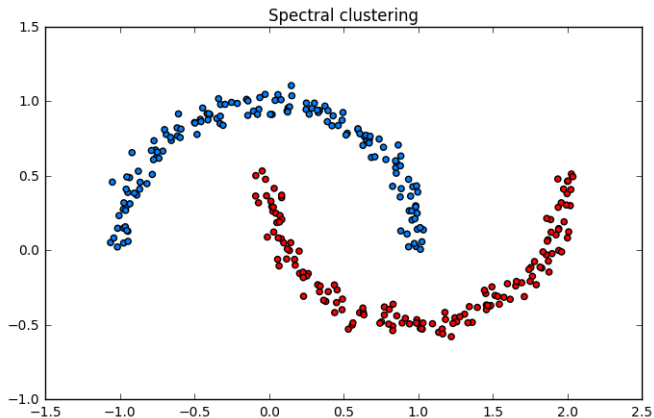
Agglomerative clustering - distance properties

- nearest neighbour may create stretched clusters
- furthest neighbour creates very compact clusters.
- group average link, centroid and median distance give the compromise.
- however centroid and median distance may lead to non-monotonous joining distance sequences in agglomerative algorithm.
- in short - group average link is preferred.

Table of Contents

- 1 K-means
- 2 Hierarchical clustering
- 3 Spectral clustering**

Spectral clustering - example



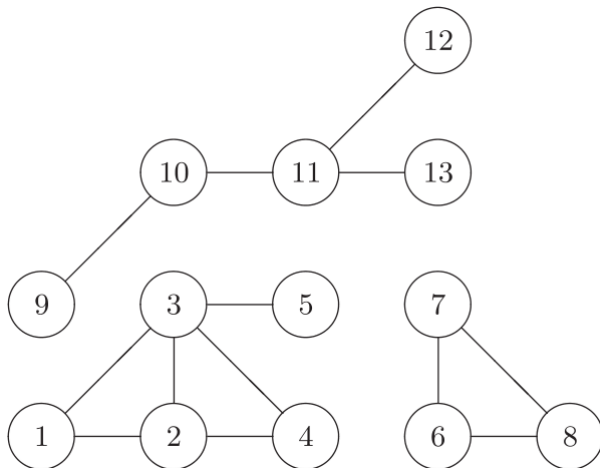
Description

- Spectral clustering relies upon similarity matrix W between objects.
- Similarity matrix \leftrightarrow weighted connection graph
- Examples:
 - nodes represent people, edge weights - how much they communicate
 - nodes represent web-pages, edge weights - scalar products of $TF - IDF$

Similarity matrix calculation

- $\|x_i - x_j\| < \textit{threshold}$
- RBF
- based on nearest neighbours

Graph with disjoint components



Graph Laplacian

- $W = W^T$, $w_{ij} \geq 0$ - the similarity between object i and object j .
- Define $D = \text{diag}\{d_1, \dots, d_N\}$, where $d_i = \sum_{j=1}^N w_{ij}$ -weighted degree of node i .
- Define graph Laplacian

$$L = D - W$$

- Properties of graph Laplacian:
 - it is symmetric
 - *It has eigenvector $\mathbf{1} \in \mathbb{R}^N$ consisting of ones with eigenvalue 0. Why?*
 - it is positive semi-definite: $\forall f \in \mathbb{R}^N : f^T L f \geq 0$.
 - L has eigenvalues $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_N = 0$

Positive semi-definiteness of Laplacian

Consider arbitrary $f \in \mathbb{R}^N$:

$$\begin{aligned} f^T Lf &= f^T Df - f^T Wf = \sum_i d_i f_i^2 - \sum_{i,j} f_i f_j w_{ij} = \\ &= \frac{1}{2} \left(\sum_i d_i f_i^2 - 2 \sum_{i,j} w_{ij} f_i f_j + \sum_j d_j f_j^2 \right) = \\ &= \frac{1}{2} \left(\sum_{i,j} w_{ij} f_i^2 - 2 \sum_{i,j} w_{ij} f_i f_j + \sum_{j,i} w_{ji} f_j^2 \right) = \quad (2) \\ &= \frac{1}{2} \left(\sum_{i,j} w_{ij} f_i^2 - 2 \sum_{i,j} w_{ij} f_i f_j + \sum_{i,j} w_{ij} f_j^2 \right) = \\ &= \frac{1}{2} \sum_{i,j} w_{ij} (f_i - f_j)^2 \geq 0 \end{aligned}$$

Eigenvectors of Laplacian

- Consider eigenvector f corresponding to eigenvalue $\lambda = 0$.
 - $f^T Lf = \lambda f^T f = 0$
- Using (2) we have that

$$0 = f^T Lf = \frac{1}{2} \sum_{i,j} w_{i,j} (f_i - f_j)^2 \quad (3)$$

- If objects i and j are connected on the graph, there exists a path with $w_{uv} > 0$ along the path and from (3) it should be that $f_i = f_j$.
- So the set of eigenvectors of L is spanned by indicator vectors $I_{A_1}, I_{A_2}, \dots, I_{A_K}$ where A_i is i -th isolated region on the graph.
- Order of $\lambda = 0$ gives the number of isolated components.

Spectral clustering algorithm:

- 1 Find order K of $\lambda = 0$
- 2 Find set of eigenvectors v_1, \dots, v_K corresponding to $\lambda = 0$
- 3 Cluster rows of $V = [v_1, \dots, v_K]$
- 4 Each row corresponds to object with the same index. Found clustering is the final clustering of initial objects.

Spectral clustering (unnormalized)

Unnormalized spectral clustering

Input: Similarity matrix $S \in \mathbb{R}^{n \times n}$, number k of clusters to construct.

- Construct a similarity graph by one of the ways described in Section 2. Let W be its weighted adjacency matrix.
- Compute the unnormalized Laplacian L .
- **Compute the first k eigenvectors u_1, \dots, u_k of L .**
- Let $U \in \mathbb{R}^{n \times k}$ be the matrix containing the vectors u_1, \dots, u_k as columns.
- For $i = 1, \dots, n$, let $y_i \in \mathbb{R}^k$ be the vector corresponding to the i -th row of U .
- Cluster the points $(y_i)_{i=1, \dots, n}$ in \mathbb{R}^k with the k -means algorithm into clusters C_1, \dots, C_k .

Output: Clusters A_1, \dots, A_k with $A_i = \{j \mid y_j \in C_i\}$.

Practical application

- $L' = D^{-1}L$ is considered instead of L (“Normalized” Laplacian)
 - to account for different connectivity levels of different nodes
- ① Most often singular values of L' are not exactly zero, but close to zero. So we select K smallest eigenvalues and corresponding K smallest eigenvectors.
- ② Cluster rows of $[v_1, \dots, v_K]$
- ③ Found clustering is applied to objects with the same indexes.

Normalized spectral clustering

Normalized spectral clustering according to Shi and Malik (2000)

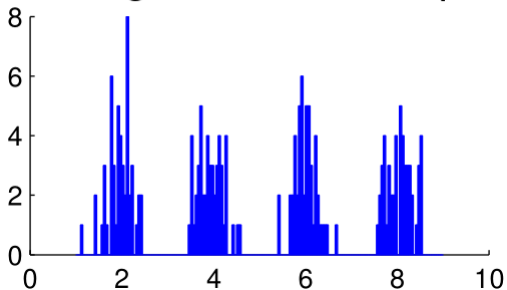
Input: Similarity matrix $S \in \mathbb{R}^{n \times n}$, number k of clusters to construct.

- Construct a similarity graph by one of the ways described in Section 2. Let W be its weighted adjacency matrix.
- Compute the unnormalized Laplacian L .
- **Compute the first k eigenvectors u_1, \dots, u_k of the generalized eigenproblem $Lu = \lambda Du$.**
- Let $U \in \mathbb{R}^{n \times k}$ be the matrix containing the vectors u_1, \dots, u_k as columns.
- For $i = 1, \dots, n$, let $y_i \in \mathbb{R}^k$ be the vector corresponding to the i -th row of U .
- Cluster the points $(y_i)_{i=1, \dots, n}$ in \mathbb{R}^k with the k -means algorithm into clusters C_1, \dots, C_k .

Output: Clusters A_1, \dots, A_k with $A_i = \{j \mid y_j \in C_i\}$.

Example

Histogram of the sample



Example

