Clustering

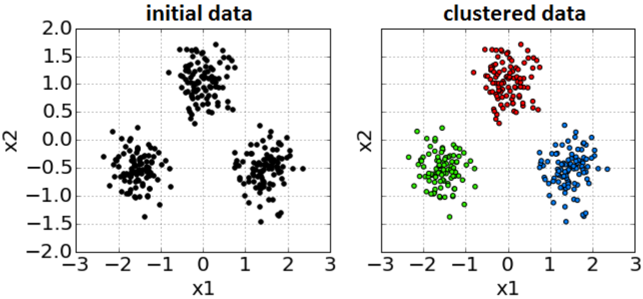
Victor Kitov

Table of Contents

Aim of clustering

- Clustering is partitioning of objects into groups so that:
  - inside groups objects are very similar
  - objects from different groups are dissimilar
- Unsupervised learning
- No definition of “similar”
  - different algorithms use different formalizations of similarity

Clustering demo



Applications of clustering

- data summarization
  - feature vector is replaced by cluster number
- feature extraction
  - cluster number, distance to native cluster center / other clusters
- customer segmentation
  - e.g. for recommender service
- community detection in networks
  - nodes - people, similarity - number of connections
- outlier detection
  - outliers do not belong any cluster

Table of Contents

1. Clustering introduction
2. K-means
3. Hierarchical clustering
4. Spectral clustering

K-means algorithm

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

$$\sum_{n=1}^{N} \rho(x_n, \mu_{z_n})^2 \to \min_{z_1,...z_N,\mu_1,...\mu_K} \tag{1}$$

  where $z_i \in \{1, 2, ...K\}$ is cluster assignment for x_i and $\mu_1, ...\mu_K$ are cluster centers.
- Direct optimization requires full search and is impractical.
- K-means is a suboptimal algorithm for optimizing (1).

K-means algorithm

```
Initialize $\mu_j$, $j = 1, 2, ...K$.

**repeat while** stop condition not satisfied:
    **for** $i = 1, 2, ...N$:
        find cluster number of $x_i$:
        $z_i = \arg\min_{j \in \{1,2,...K\}} ||x_i - \mu_j||$
    **for** $j = 1, 2, ...K$:
        $\mu_j = \frac{1}{\sum_{n=1}^{N} \mathbb{I}[z_n = j]} \sum_{n=1}^{N} \mathbb{I}[z_n = j] x_i$
```

Dynamic K-means algorithm

```
Initialize $\mu_j$, $j=1,2,...K$, $z_i=0, i=1,2,...N$

repeat while stop condition not satisfied:
    for $i=1,2,...N$:
        find cluster number of $x_i$:
        $z_i' = \arg\min_{j\in\{1,2,...K\}} ||x_i-\mu_j||$
        if $z_i'!=z_i$:
            recalculate cluster means $\mu_{z_i}$ 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.

K-means properties

Possible stop conditions:

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

Initialization:

- typically $\{\mu_i\}_{i=1}^K$ are initialized to randomly chosen training objects

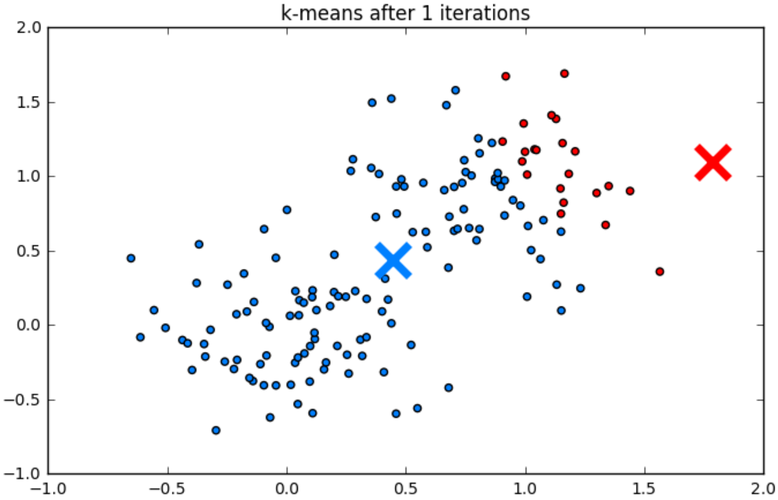
Optimality:

- criteria is non-convex
- solution depends on starting conditions
- we may restart several times from diff. random starting points and select solution giving minimal value of (1).

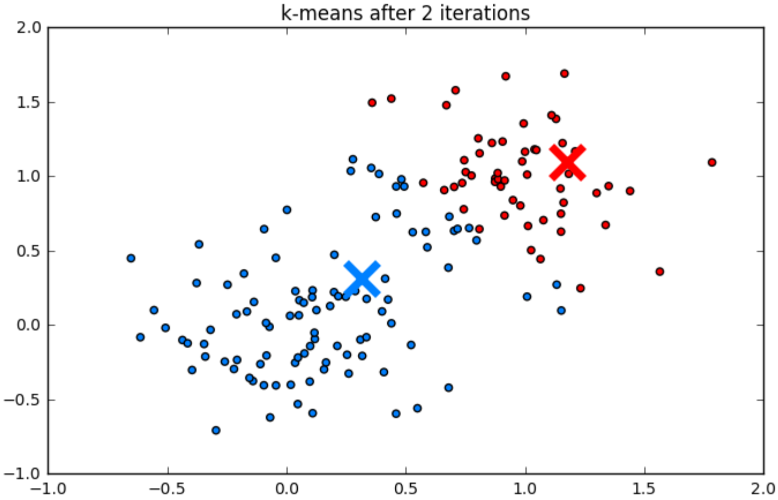
Complexity: $O(NDKI)$, where K is the number of clusters and I is the number of iterations.

- Usually algorithm converges in small number of iterations I.

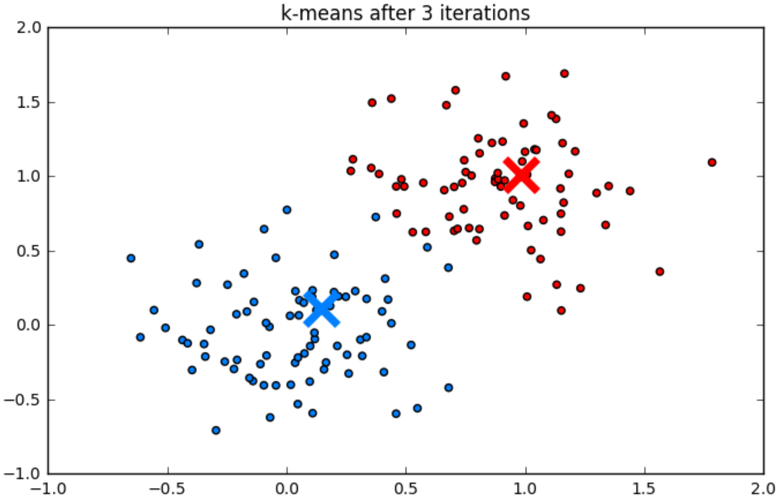
Example of K-means



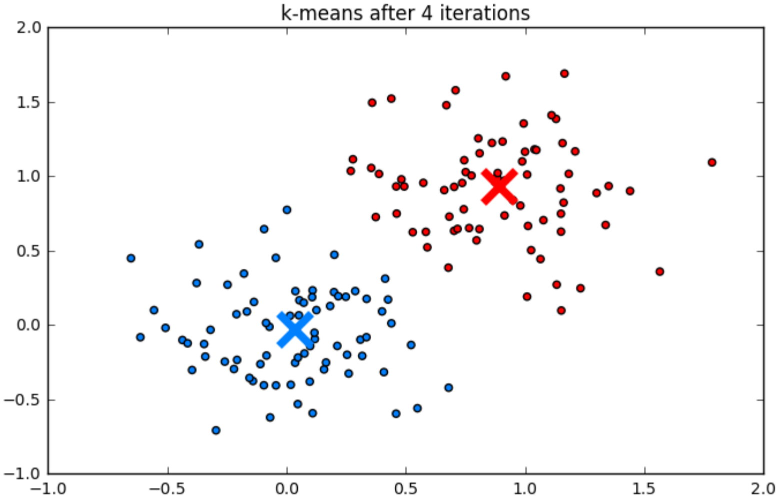
Example of K-means



Example of K-means

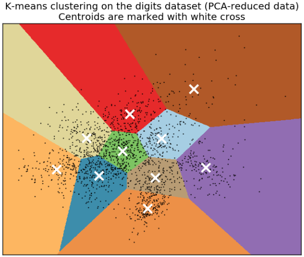


Example of K-means



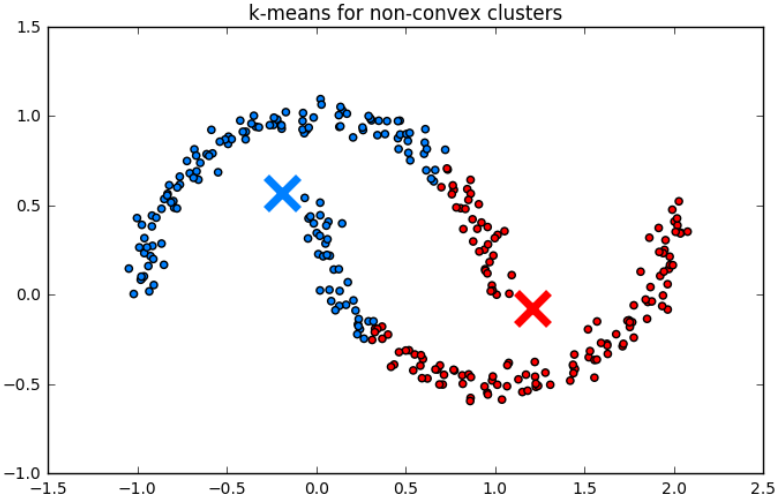
Gotchas

- K-means assumes that clusters are convex:

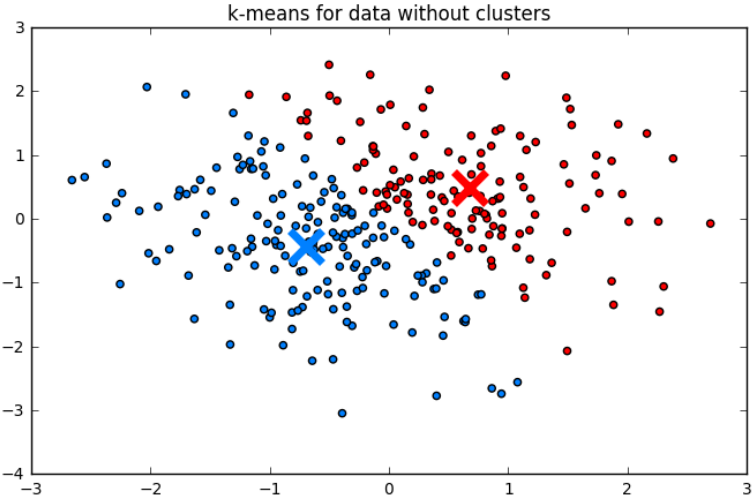


- It always finds clusters even if none actually exist
    - need to control cluster quality metrics

K-means for non-convex clusters



K-means for data without clusters



K-means and EM algorithm

```
Initialize $\mu_j$, $j = 1, 2, ...K$.

repeat while stop condition not satisfied:
    for $i = 1, 2, ...N$:
        find cluster number of $x_i$:
        $z_i = \arg\min_{j \in \{1,2,...g\}} ||x_i - \mu_j||$
    for $j = 1, 2, ...K$:
        $\mu_j = \frac{1}{\sum_{n=1}^{N} \mathbb{I}[z_n = j]} \sum_{n=1}^{N} \mathbb{I}[z_n = j] x_i$
```

- K-means is EM-algorithm when:

K-means and EM algorithm

```
Initialize $\mu_j$, $j = 1, 2, ...K$.

repeat while stop condition not satisfied:
    for $i = 1, 2, ...N$:
        find cluster number of $x_i$:
        $z_i = \arg\min_{j \in \{1,2,...g\}} ||x_i - \mu_j||$
    for $j = 1, 2, ...K$:
        $\mu_j = \frac{1}{\sum_{n=1}^{N} \mathbb{I}[z_n = j]} \sum_{n=1}^{N} \mathbb{I}[z_n = j] x_i$
```

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

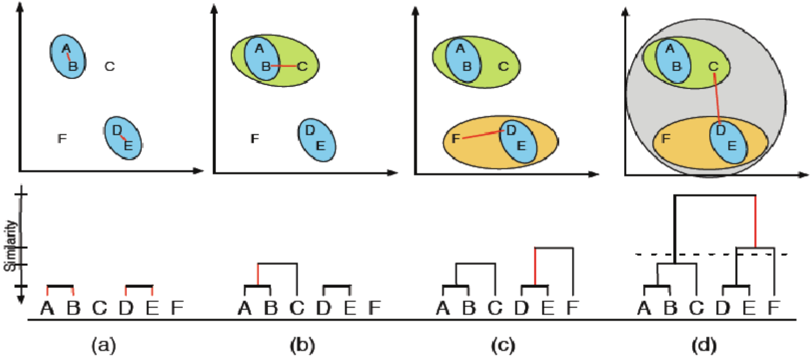
Table of Contents

Hierarchical clustering

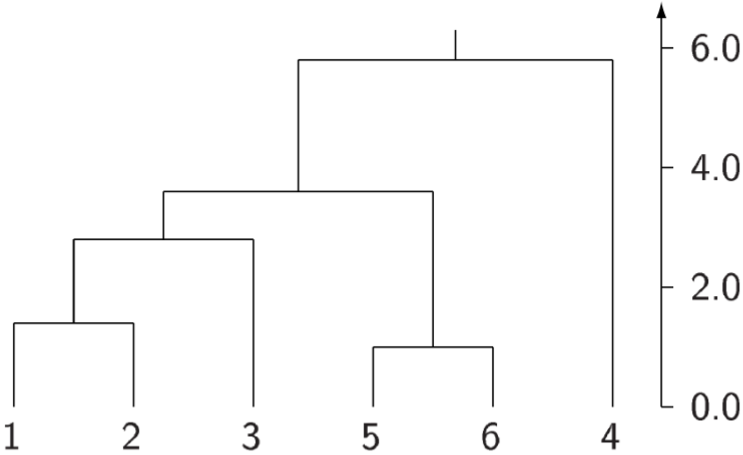
Hierarchical clustering may be:

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

Bottom-up clustering demo



Agglomerative clustering



Agglomerative clustering - distances

- Consider clusters $A = \{x_{i_1}, x_{i_2}, ...\}$ and $B = \{x_{j_1}, x_{j_2}, ...\}$.
- 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 ($m_U = median_{x \in U}\{x\}$)

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

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 $\rho(A, B)$

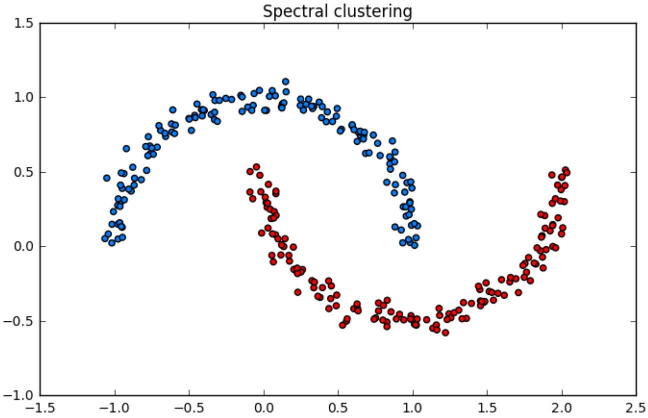
Agglomerative clustering - distance properties

- nearest neighbour may create stretched clusters
- furtherst 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. Clustering introduction
2. K-means
3. Hierarchical clustering
4. Spectral clustering

Spectral clustering - example



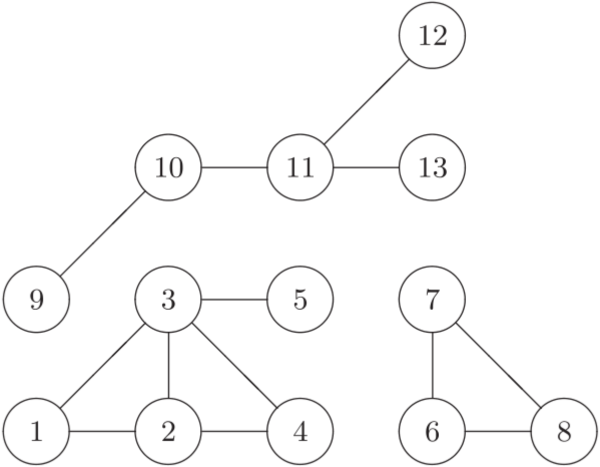
Description

- Spectral clustering relies upon similarity matrix W between objects.
- Similarity matrix <-> 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 $D = \text{diag}\{d_1, ...d_N\}$, 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 $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_N = 0$

Positive semi-definiteness of Laplacian

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

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

Eigenvectors of Laplacian

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

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

- 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 $I_{A_1}, I_{A_2}, ... I_{A_K}$ 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 $v_1, ...v_K$ corresponding to $\lambda = 0$
3. Cluster rows of $V = [v_1, ...v_K]$
4. Each row corresponds to object with the same index. Found clustering is the final clustering of initial objects.