Clustering

Victor Kitov

v.v.kitov@yandex.ru

Yandex School of Data Analysis

Table of Contents

K-means algorithm

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

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

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

K-means algorithm

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

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 $j = 1, 2, ...g$:
        $\mu_j = \frac{1}{\sum_{n=1}^{N} \mathbb{I}[z_n=j]} \sum_{n=1}^{N} \mathbb{I}[z_n = j] x_i$
```

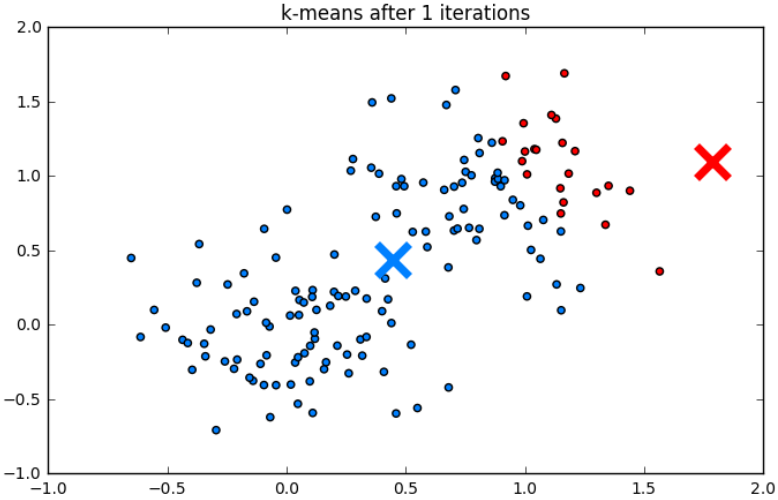
Possible stop conditions:

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

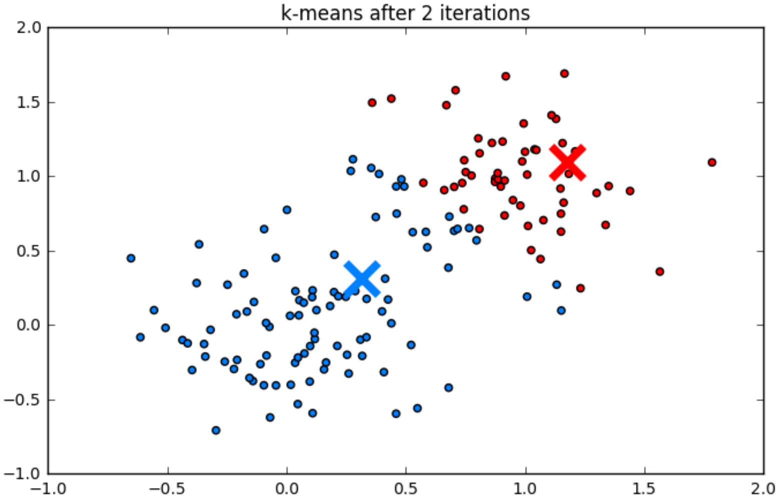
K-means properties

- 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 $O(NDg)$

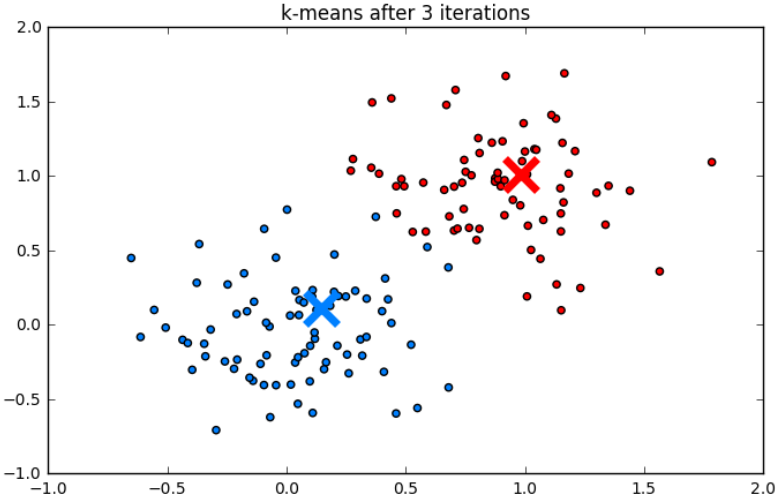
Example of K-means



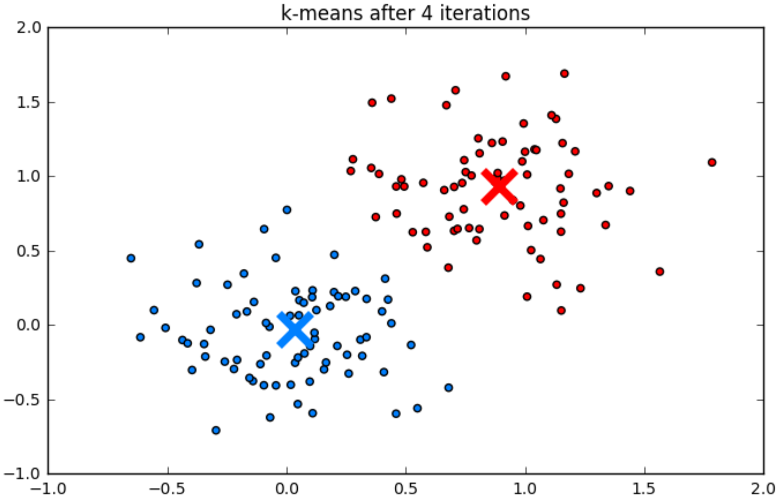
Example of K-means



Example of K-means

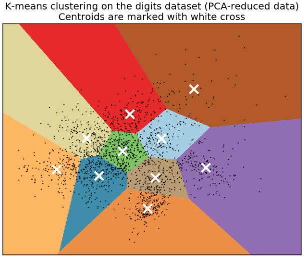


Example of K-means



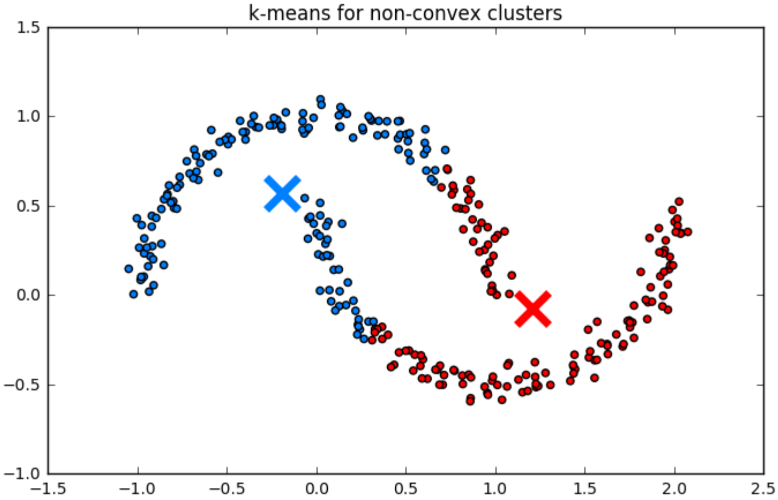
Gotchas

- K-means assumes that clusters are convex:



- 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

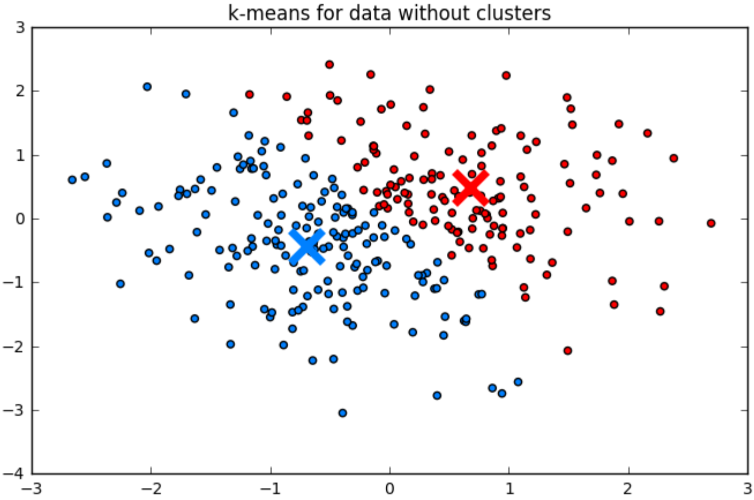


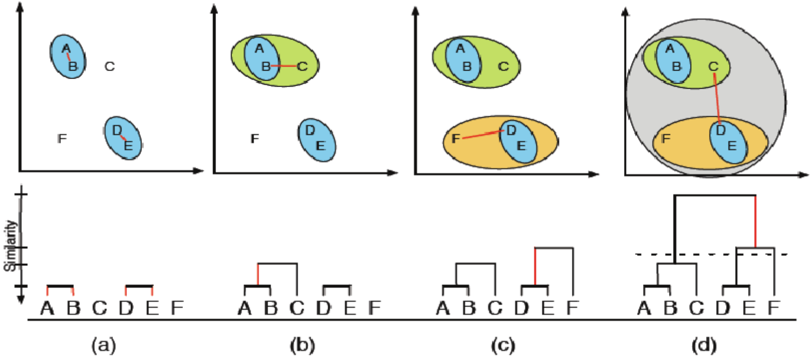
Table of Contents

Hierarchical clustering

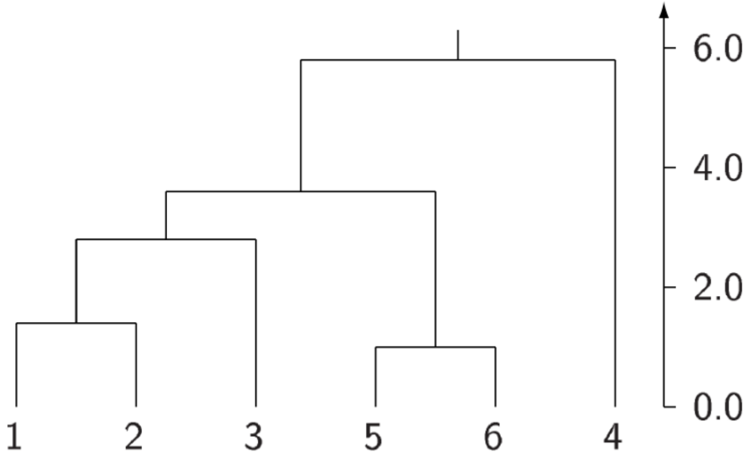
Hierarchical clustering may be:

- 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

- 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.