Clustering

Victor Kitov

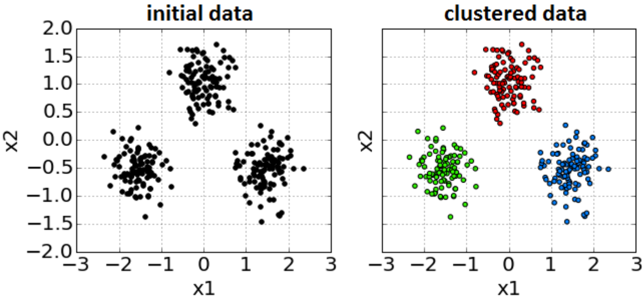
v.v.kitov@yandex.ru

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, cluster average target, 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

Clustering algorithms comparison

We can compare clustering algorithms in terms of:

- computational complexity
- do they build flat or hierarchical clustering?
- can the shape of clustering be arbitrary?
    - if not is it symmetrical, can clusters be of different size?
- can clusters vary in density of contained objects?
- robustness to outliers

Table of Contents

Representative-based clustering

- Clustering is flat (not hierarchical)
- Number of clusters K is specified in advance
- Each object x_n is associated cluster z_n
- Each cluster C_k is defined by its representative μ_k, $k = 1, 2, ...K$.
- Criterion to find representatives $\mu_1, ...\mu_K$:

$$Q(z_1, ...z_K) = \sum_{n=1}^{N} \min_k \rho(x_n, \mu_k) \to \min_{\mu_1, ...\mu_K} \tag{1}$$

Generic algorithm

```
initialize μ_1, ...μ_K from
random training objects

WHILE not converged:
    FOR n = 1, 2, ...N:
        z_n = arg min_k ρ(x_n, μ_k)

    FOR k = 1, 2, ...K:
        μ_k = arg min_μ Σ_{n: z_n = k} ρ(x_n, μ)

RETURN z_1, ...z_N
```

Comments

- different distance functions lead to different algorithms:
  - $\rho(x, x') = \|x - x'\|_2^2$=> K-means
  - $\rho(x, x') = \|x - x'\|_1$=> K-medians
- μ_k may be arbitrary or constrained to be existing objects
- K - unknown parameter
  - if chosen small=>distinct clusters will get merged
  - better to take K larger and then merge similar clusters.
- Shape of clusters is defined by $\rho(\cdot, \cdot)$
- Close clusters will have similar size.

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

$$\sum_{n=1}^{N} \|x_n - \mu_{z_n}\|_2^2 \to \min_{z_1,...z_N,\mu_1,...\mu_K} \tag{2}$$

  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 (2).

K-means algorithm

```
Initialize μ_j, j = 1, 2, ...K.

WHILE not converged:

    FOR i = 1, 2, ...N:
        find cluster number of x_i:
        z_i = arg min_{j∈{1,2,...K}} ||x_i − μ_j||_2^2

    FOR j = 1, 2, ...K:
        μ_j = 1/(∑_{n=1}^N 𝕀[z_n=j]) ∑_{n=1}^N 𝕀[z_n = j] x_i
```

Initialize μ_j, $j = 1, 2, ...K$.

WHILE not converged:

- **FOR** $i = 1, 2, ...N$:
  - find cluster number of x_i:
  - $z_i = \arg\min_{j \in \{1,2,...K\}} ||x_i - \mu_j||_2^2$
- **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 properties

Convergence conditions:

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

Initialization:

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

K-means properties

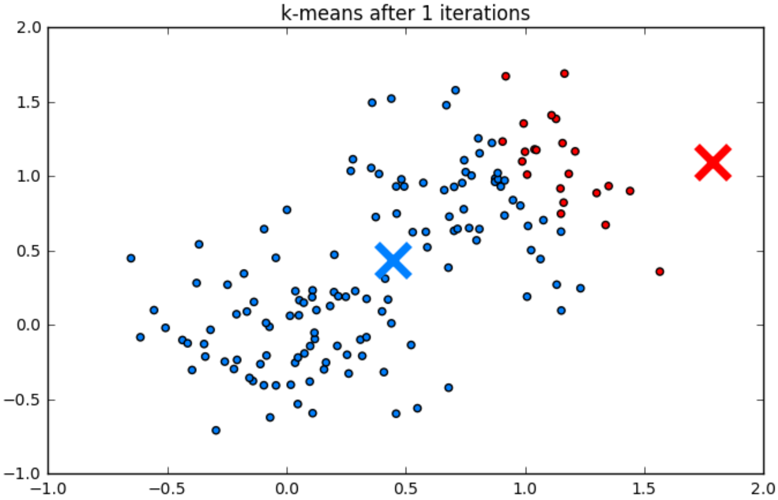
Optimality:

- criteria is non-convex
- solution depends on starting conditions
- may restart several times from different initializations and select solution giving minimal value of (2).

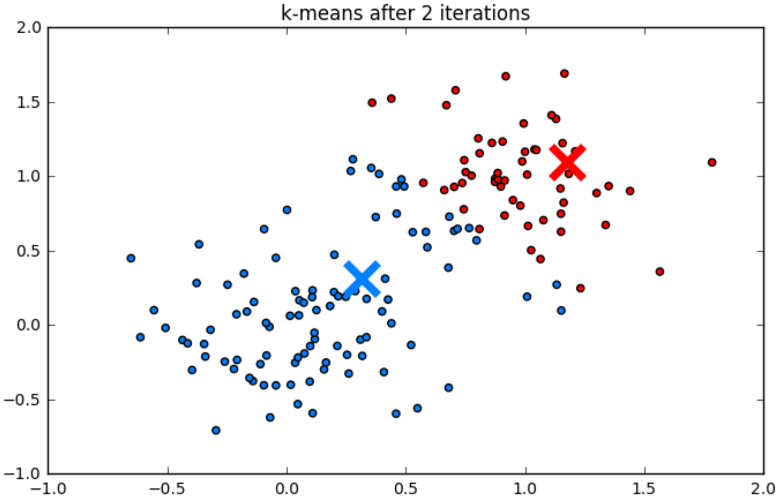
Complexity: $O(NDKI)$

- K is the number of clusters
- I is the number of iterations.
    - usually few iterations are enough for convergence.

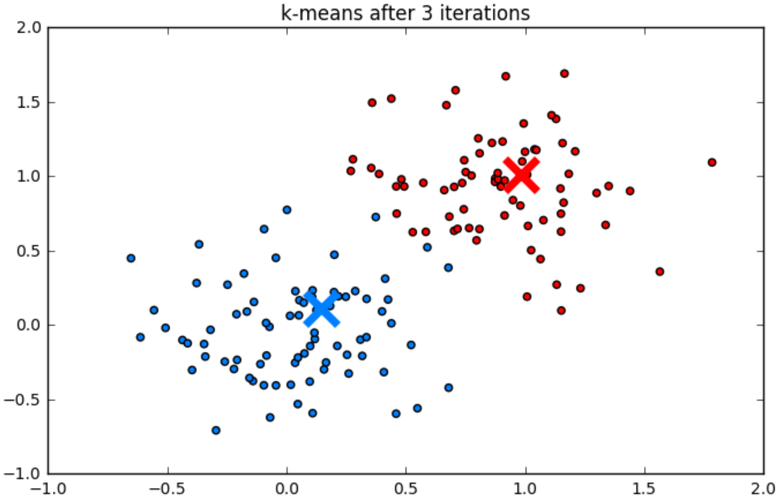
Example of K-means



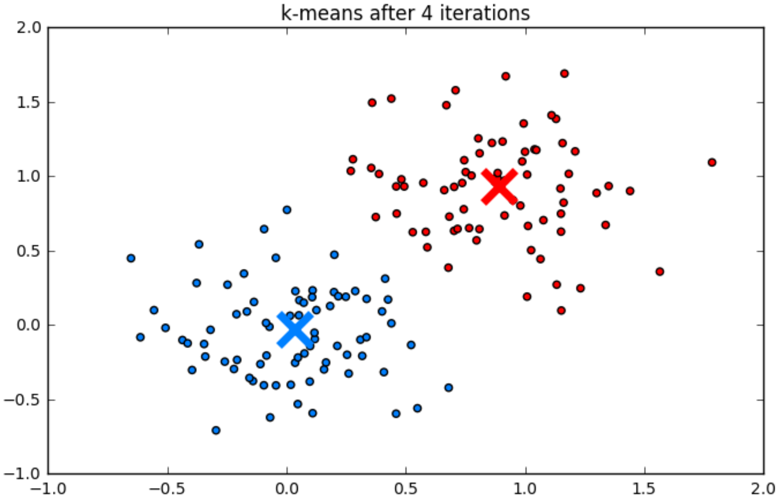
Example of K-means



Example of K-means

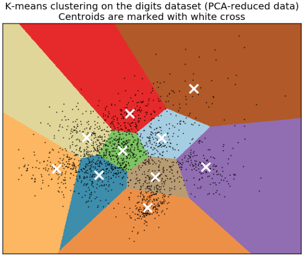


Example of K-means



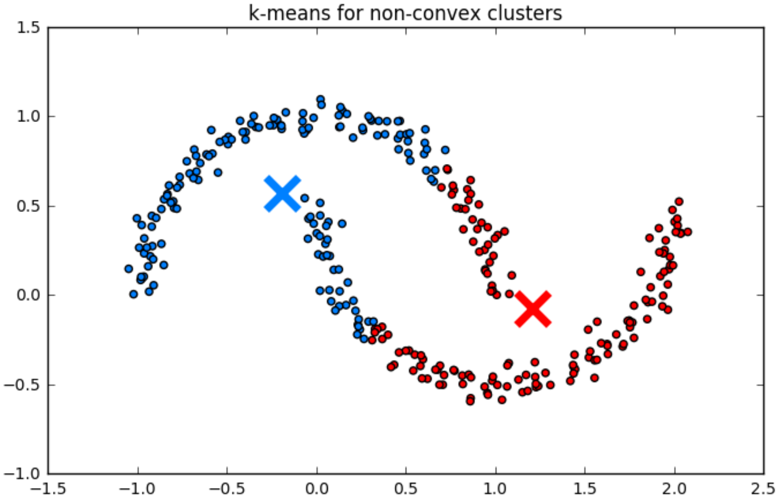
Gotchas

- K-means assumes that clusters are convex:

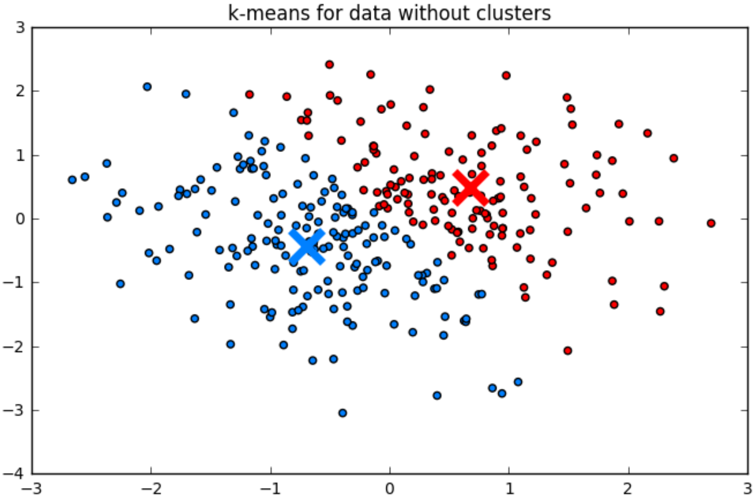


- It always finds clusters even if none actually exist
  - need to control cluster quality metrics

K-means for non-convex clusters



K-means for data without clusters



K-means and EM algorithm

```
Initialize μ_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∈{1,2,...g}} ||x_i − μ_j||
  for j = 1, 2, ...K:
    μ_j = 1/(Σ_{n=1}^N 𝕀[z_n=j]) Σ_{n=1}^N 𝕀[z_n = j] x_i
```

- K-means is EM-algorithm when:

K-means and EM algorithm

```
Initialize μ_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∈{1,2,...g}} ||x_i − μ_j||
   for j = 1,2,...K:
      μ_j = 1/(Σ_{n=1}^N 𝕀[z_n=j]) Σ_{n=1}^N 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

K-means

- Not robust to outliers
    - K-medians is robust
- K-representatives may create singleton clusters in outliers if centroids get initialized with outlier
    - better to init centroids with mean of m randomly chosen objects
- Constructs spherical clusters of similar radii
    - Allows kernel version which can find non-convex clusters in original space

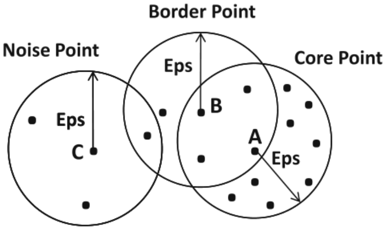
General comments on K-representatives

- Init $\{\mu_k\}_{k=1}^{K}$ with
  - random objects from training set
  - centroids of m randomly selected objects from training set (more robust to outliers)
- K-representatives has non-convex optimization criteria
  - depends in initialization of $\{\mu_k\}_{k=1}^{K}$
  - so we can restart clustering from different starting conditions and select the one, maximizing (1)
- Outliers can create singleton clusters consisting of 1 point.
  - apply outlier filtering beforehand
  - alternatively during clustering for clusters with too few points replace cluster centroids with random objects.

Table of Contents

DBScan

- Core point: point having $\geq k$ points in its ε neighbourhood
- Border point: not core point, having at least 1 core point in its ε neighbourhood
- Noise point: neither a core point nor a border point



- k, ε - parameters of the method.

Algorithm

```
INPUT: training set, parameters ε, k.

1) Determine core, border and noise points with ε, k.
2) Create graph in which core points are connected if they
   are within ε of one another
3) Determine connected components in the graph
4) Assign each border point to connected component with which
   it is best connected

RETURN points in each connected component as a cluster
```