Clustering

Victor Kitov

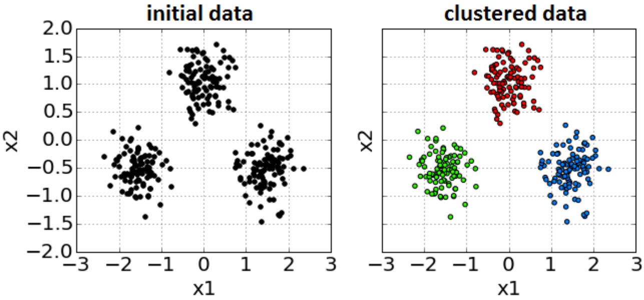
v.v.kitov@yandex.ru

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)  #mean for L2 sq

RETURN z_1,...z_N
```

Comments

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/

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 properties

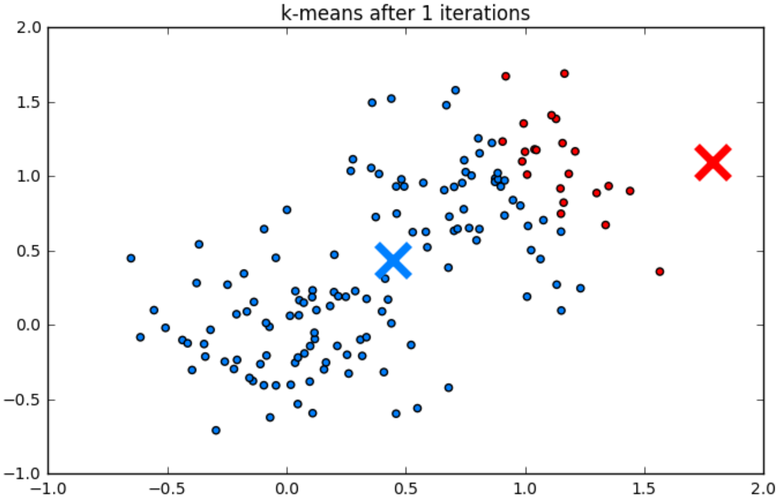
Optimality:

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

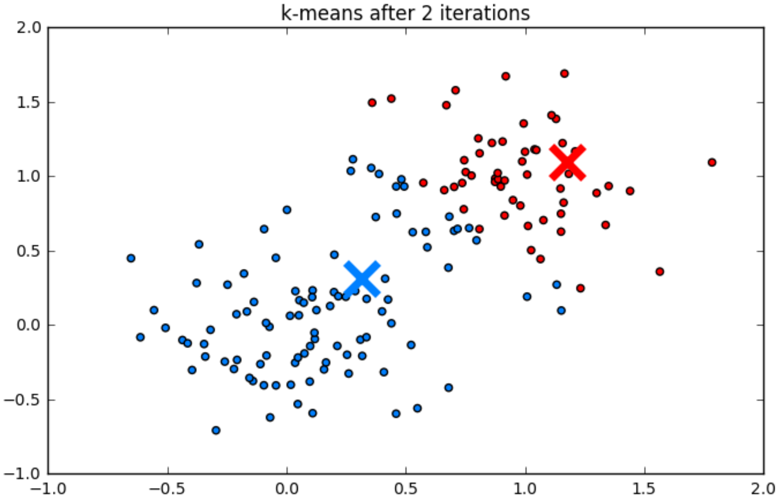
Complexity: $O(NDKI)$

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

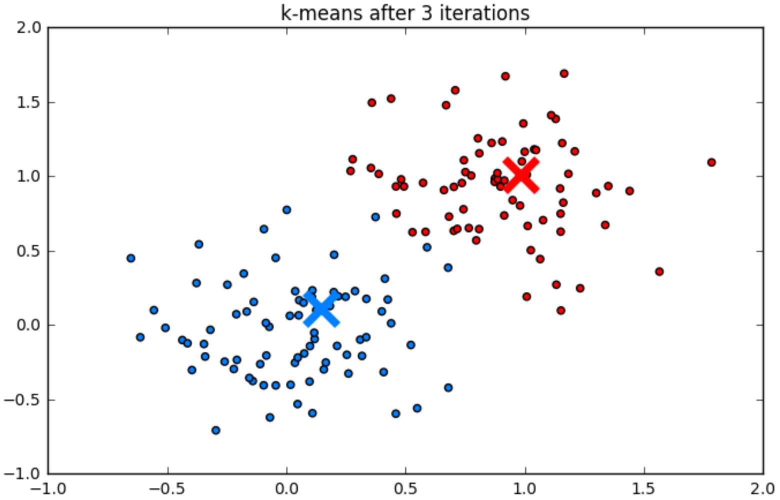
Example of K-means



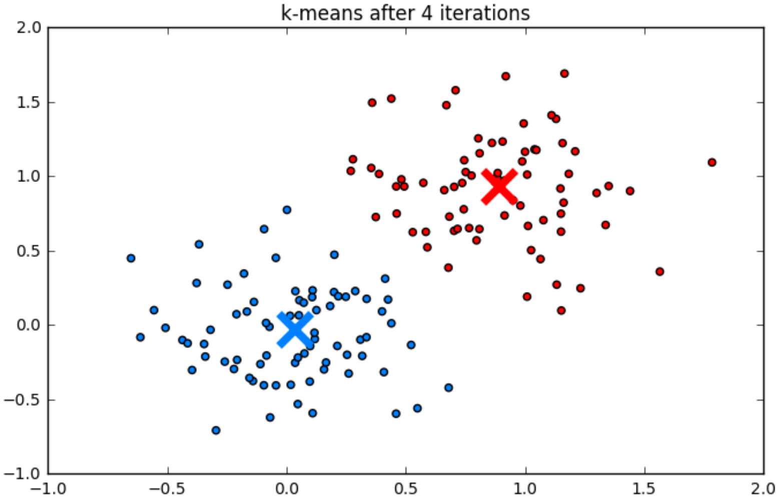
Example of K-means



Example of K-means

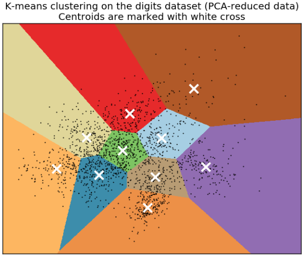


Example of K-means



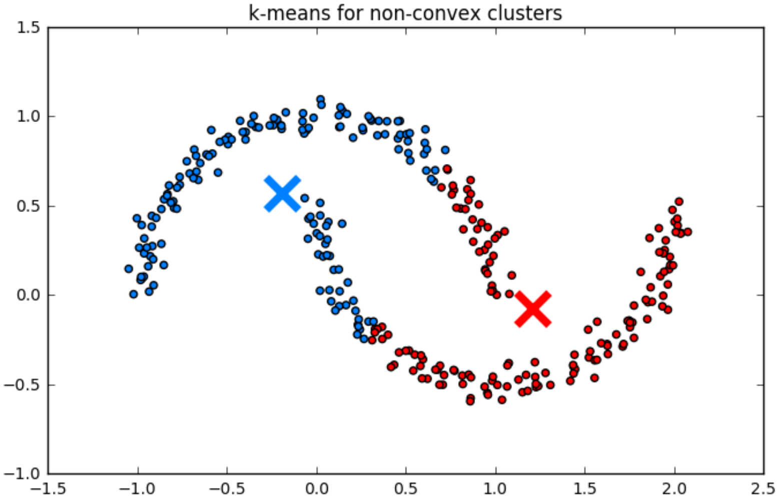
Gotchas

- 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

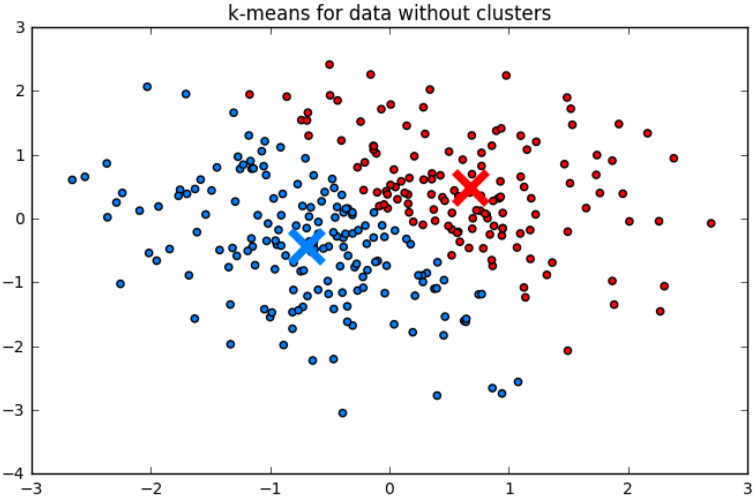


Table of Contents

Motivation

- Number of clusters K not known a priory.
- Clustering is usually not flat, but hierarchical with different levels of granularity:
    - sites in the Internet
    - books in library
    - animals in nature

Hierarchical clustering

Hierarchical clustering may be:

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

3. Hierarchical clustering
   - Top-down hierarchical clustering
   - Bottom-up hierarchical clustering
   - DBScan

Algorithm

INPUT:
data D, flat clustering algorithm A
leaf selection criterion, termination criterion

Initialize tree T to root, containing all data

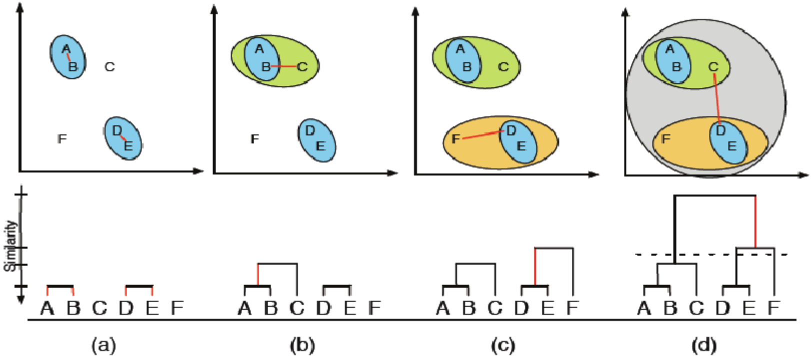
REPEAT
  based on selection criterion, select leaf L
  using algorithm A split L into children $L_1, ...L_K$
  add $L_1, ...L_K$ as child nodes to tree T
UNTIL termination criterion

Comments

- Leaf selection criterion:
    - split leaf most close to the root
        - result: balanced tree by height
    - split leaf with maximum elements
        - result: balanced tree by cluster size
- Building hierarchy top-down is more natural for a human

3 Hierarchical clustering
- Top-down hierarchical clustering
- Bottom-up hierarchical clustering
- DBScan

Bottom-up clustering demo



Algorithm

initialize distance matrix $M \in \mathbb{R}^{N \times N}$ between
singleton clusters $\{x_1\},...\{x_N\}$

REPEAT:
1) pick closest pair of clusters i and j
2) merge clusters i and j
3) delete rows/columns i, j from M and add
   new row/column for merged cluster

UNTIL 1 cluster is left

RETURN hiearchical clustering of objects

- Early stopping is possible when:
  - K clusters are left
  - distance between most close clusters $\geq$threshold

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

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

  - closest centroid

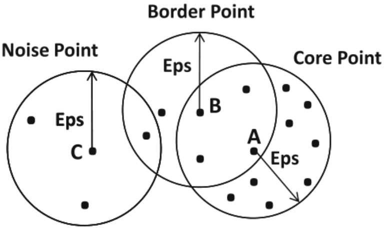
$$\rho(A, B) = \rho(\mu_A, \mu_B)$$

where $\mu_U = \frac{1}{|U|} \sum_{x \in U} x$ or $m_U = median_{x \in U}\{x\}$

3. Hierarchical clustering
   - Top-down hierarchical clustering
   - Bottom-up hierarchical clustering
   - DBScan

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.

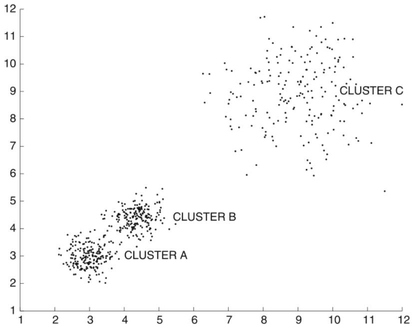
Algorithm

```
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
```

Failure for varying density



- Large k: cluster C is missed
- Small k: clusters A and B get merged

Comments

- Connecting core points - agglomerative clustering with single linkage, stopping at distance ε.
- Advantages:
    - Resistant to outliers by ignoring noise points.
    - automatically determines the number of clusters
- Disadvantages:
    - works badly for density varying clusters
- Complexity $O(N^2 Dk)$
    - can be reduced to $O(N \ln NDk)$ for small D with spatial indexing.

Mean shift clustering

INPUT: training set $x_1, ...x_N$, step size η,
kernel $K(\cdot)$, bandwidth h.

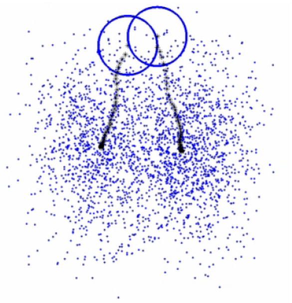
FOR $n = 1, ...N$:
$z_0 = x_n, i = 0$
REPEAT until convergence:
$$z_{i+1} = \frac{\sum_{k=1}^{N} K(\rho(z_i, x_k)/h)x_k}{\sum_{k=1}^{N} K(\rho(z, x_k)/h)}$$
$i = i + 1$
assosiate x_n to peak z_i

Merge almost identical peak positions $z_1, ...z_N$

RETURN clusters of data points, converging to the same peak.

Comments

Mean shift convergence process



- Mean shift clustering is equivalent to steepest gradient clustering.
- Usually RBF kernel $K(\rho(x,x')/h) = e^{-\rho(x,x')^2/h^2}$ is used
- Efficient to discard objects that are outside some ε-neighbourhood of z_i in z_i recalculation.

Clustering evaluation: Silhuette coefficient[1]

For each object x_i define:

- s_i-mean distance to objects in the same cluster
- d_i-mean distance to objects in the next nearest cluster

Silhouette coefficient for x_i:

$$Silhouette_i = \frac{d_i - s_i}{\max\{d_i, s_i\}}$$

Silhouette coefficient for $x_1, ...x_N$:

$$Silhouette = \frac{1}{N}\sum_{i=1}^{N} \frac{d_i - s_i}{\max\{d_i, s_i\}}$$

[1] Peter J. Rousseeuw (1987). "Silhouettes: a Graphical Aid to the Interpretation and Validation of Cluster Analysis". Computational and Applied Mathematics 20: 53–65.

Discussion

- Advantages
  - The score is bounded between -1 for incorrect clustering and +1 for highly dense clustering.
  - Scores around zero indicate overlapping clusters.
  - The score is higher when clusters are dense and well separated.
- Disadvantages
  - complexity $O(N^2D)$
    - use feature space indexing or random subsampling
  - favours convex clusters

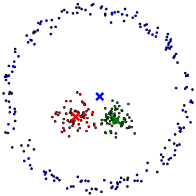


Table of Contents

Isolation forest

- Isolation tree splitting algorithm:

while nodes with ≥ 2 observations exist:
    take node with ≥ 2 observations
    select random non-constant feature f for that node
    select random threshold $t \in [f_{min}, f_{max})$
    split current node into 2 nodes depending on $f \leq t$ rule

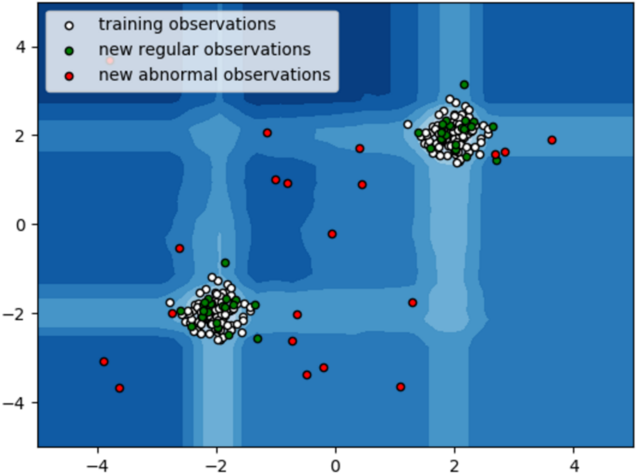
Isolation forest

- Isolation tree splitting algorithm:

while nodes with ≥ 2 observations exist:
    take node with ≥ 2 observations
    select random non-constant feature f for that node
    select random threshold $t \in [f_{min}, f_{max})$
    split current node into 2 nodes depending on $f \leq t$ rule

  - Typicalness of object≈depth of the node containing only that object
    - outliers are easier to separate
    - but depends too much on randomness
- Isolation forest - collection of M independent isolation trees.
  - Typicalness of object=average depth of the node of that object in M trees.
  - outlier score = - typicalness.

Example


training observations
new regular observations
new abnormal observations
4
2
0
−2
−4
−4
−2
0
2
4