Clustering

May 8, 2016

k-Clustering

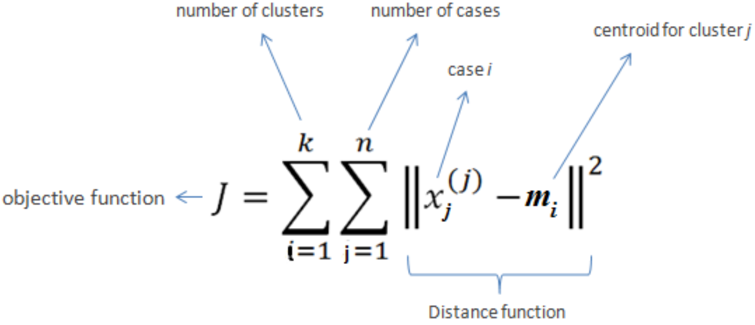
k-Means clustering intends to partition n objects into k clusters in which each object belongs to the cluster with the nearest mean.

- Produces k different clusters of greatest possible distinction.

k-Clustering

k-Means clustering intends to partition n objects into k clusters in which each object belongs to the cluster with the nearest mean.

- Produces k different clusters of greatest possible distinction.
- Works by minimizing the squared error function:



Applications

Used as a preprocessing step for other algorithms, for example to find a starting configuration.

- Market segmentation
- Computer vision
- Wireless sensor networks
  - Clustering algorithm plays the role in finding Cluster heads(or cluster centers) which collects all the data in its respective cluster.
- Astronomy: The Large Synoptic Survey Telescope (LSST) images the full southern sky every few days, requiring more than 30 terabytes to be processed and stored every day during ten years.
  - Select rare and alike targets, identify noise, ...

Algorithm

The algorithm partitions the data into k groups, where k is predefined. Identifying the best number of clusters k leading to the greatest separation is out of this algorithm's scope.

1. Select k points at random as cluster centers.
2. Assign objects to their closest cluster center according to the Euclidean distance function.
3. Calculate the centroid or mean of all objects in each cluster.
4. Repeat steps 2, 3 and 4 until the same points are assigned to each cluster in consecutive rounds.

Algorithm - Textbook “Mathematics of Fuzzy Sets and Fuzzy Logic”

Given a data set $X = \{x_1, \ldots, x_n\}$ and $k \leq n$, find a partition $S_1, \ldots, S_k$ of X such that it minimizes

$$J\left(S_i, m_i\right)_{i=1,\ldots,k} = \sum_{i=1}^{k} \sum_{x_j \in S_i} \|x_j - m_i\|^2$$

where m_i is the mean of S_i.

Algorithm - Textbook "Mathematics of Fuzzy Sets and Fuzzy Logic"

Given a data set $X = \{x_1, \ldots, x_n\}$ and $k \leq n$, find a partition $S_1, \ldots, S_k$ of X such that it minimizes

$$J(S_i, m_i)_{i=1,\ldots,k} = \sum_{i=1}^{k} \sum_{x_j \in S_i} \|x_j - m_i\|^2$$

where m_i is the mean of S_i.

k-means algorithm. (MacQueen)
Assign:

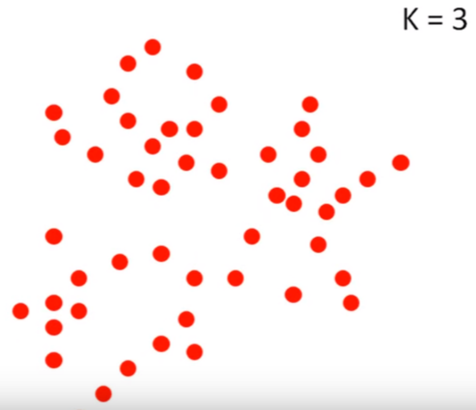
$$S_i = \{x_p \mid \|x_p - m_i\| \leq \|x_p - m_j\|, j = 1, ..., k\}$$

Update:

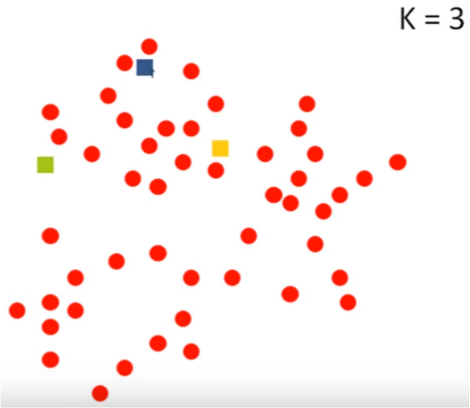
$$m_i = \frac{\sum_{x_j \in S_i} x_j}{|S_i|},$$

where $|S_i| = \sum_{x_j \in S_i} 1$ denotes the cardinality (number of elements) of the finite set.

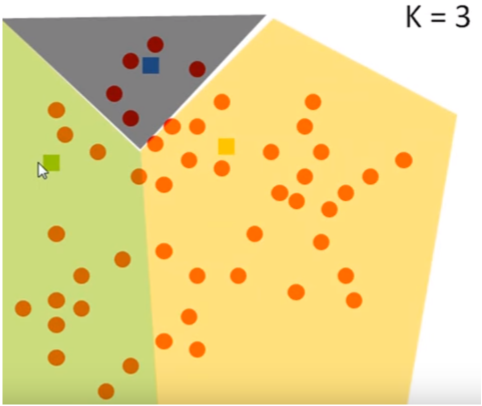
k-Clustering - Illustrated



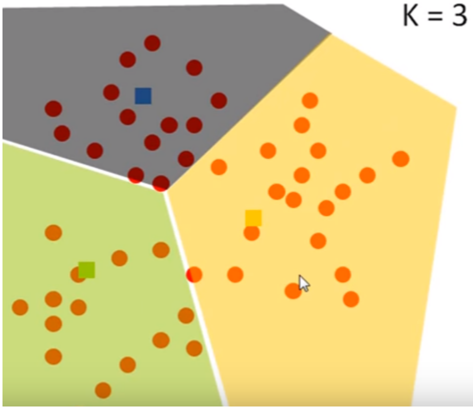
k-Clustering - Illustrated



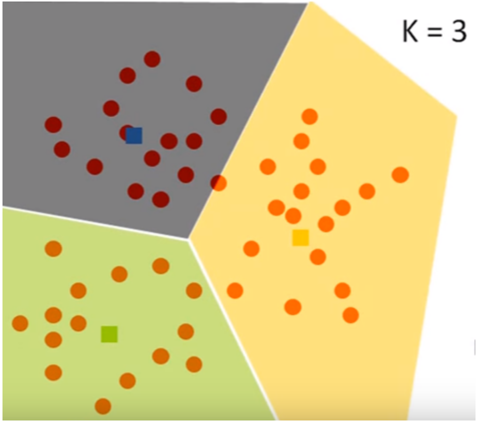
k-Clustering - Illustrated



k-Clustering - Illustrated



k-Clustering - Illustrated



Example

Suppose we want to group the visitors to a website using just their age into $k = 2$ clusters. Age data points: 15, 15, 16, 19, 19, 20, 20, 21, 22, 28, 35, 40, 41, 42, 43, 44, 60, 61, 65

Example

Suppose we want to group the visitors to a website using just their age into $k = 2$ clusters. Age data points: 15, 15, 16, 19, 19, 20, 20, 21, 22, 28, 35, 40, 41, 42, 43, 44, 60, 61, 65

Initial clusters:

Centroid (C1) = 16 [16]

Centroid (C2) = 22 [22]

Example

Suppose we want to group the visitors to a website using just their age into $k = 2$ clusters. Age data points: 15, 15, 16, 19, 19, 20, 20, 21, 22, 28, 35, 40, 41, 42, 43, 44, 60, 61, 65

Initial clusters:

Centroid (C1) = 16 [16]

Centroid (C2) = 22 [22]

1. Iteration 1:

   C1 = 15.33 [15,15,16]

   C2 = 36.25 [19,19,20,20,21,22,28,35,40,41,42,43,44,60,61,65]

Example

Suppose we want to group the visitors to a website using just their age into $k = 2$ clusters. Age data points: 15, 15, 16, 19, 19, 20, 20, 21, 22, 28, 35, 40, 41, 42, 43, 44, 60, 61, 65

Initial clusters:

Centroid (C1) = 16 [16]

Centroid (C2) = 22 [22]

1. Iteration 1:
   C1 = 15.33 [15,15,16]
   C2 = 36.25 [19,19,20,20,21,22,28,35,40,41,42,43,44,60,61,65]
2. Iteration 2:
   C1 = 18.56 [15,15,16,19,19,20,20,21,22]
   C2 = 45.90 [28,35,40,41,42,43,44,60,61,65]

Example

3. Iteration 3:
   C1 = 19.50 [15,15,16,19,19,20,20,21,22,28]
   C2 = 47.89 [35,40,41,42,43,44,60,61,65]

Example

3. Iteration 3:
   C1 = 19.50 [15,15,16,19,19,20,20,21,22,28]
   C2 = 47.89 [35,40,41,42,43,44,60,61,65]
4. Iteration 4:
   C1 = 19.50 [15,15,16,19,19,20,20,21,22,28]
   C2 = 47.89 [35,40,41,42,43,44,60,61,65]

Example

3. Iteration 3:
   C1 = 19.50 [15,15,16,19,19,20,20,21,22,28]
   C2 = 47.89 [35,40,41,42,43,44,60,61,65]
4. Iteration 4:
   C1 = 19.50 [15,15,16,19,19,20,20,21,22,28]
   C2 = 47.89 [35,40,41,42,43,44,60,61,65]

No change between iterations 3 and 4 has been noted. By using clustering, 2 groups have been identified 15-28 and 35-65. The initial choice of centroids can affect the output clusters, so the algorithm is often run multiple times with different starting conditions in order to get a fair view of what the clusters should be.

Demo

Correctness of the Algorithm

Proposition (convergence): After a finite number of steps, neither the Assignment nor the Update steps modify the output of the algorithm (S_i's).

Correctness of the Algorithm

Proposition (convergence): After a finite number of steps, neither the Assignment nor the Update steps modify the output of the algorithm (S_i's).

Proof:

1. "Assignment" decreases J.

Correctness of the Algorithm

Proposition (convergence): After a finite number of steps, neither the Assignment nor the Update steps modify the output of the algorithm (S_i's).
Proof:

1. "Assignment" decreases J.If a point x_q was misplaced in S_k, i.e., $||x_q - m_k|| \geq ||x_q - m_j||$, for some $j \in \{1, \ldots, k\}$ then it will be placed in S_j and this way $J(S_i, m_i)$ will decrease.

Correctness of the Algorithm

Proposition (convergence): After a finite number of steps, neither the Assignment nor the Update steps modify the output of the algorithm (S_i's).
Proof:

1. "Assignment" decreases J.If a point x_q was misplaced in S_k, i.e., $||x_q - m_k|| \geq ||x_q - m_j||$, for some $j \in \{1, \ldots, k\}$ then it will be placed in S_j and this way $J(S_i, m_i)$ will decrease.
2. "Update" decreases J.

Correctness of the Algorithm

Proposition (convergence): After a finite number of steps, neither the Assignment nor the Update steps modify the output of the algorithm (S_i's).
Proof:

1. "Assignment" decreases J.If a point x_q was misplaced in S_k, i.e., $||x_q - m_k|| \geq ||x_q - m_j||$, for some $j \in \{1, \ldots, k\}$ then it will be placed in S_j and this way $J(S_i, m_i)$ will decrease.
2. "Update" decreases J.Notice that J is convex and thus a local minimum is a global minimum.

$$\frac{\partial J}{\partial m_i} = 2 \sum_{x_j \in S_i} (m_i - x_j), i = 1, \ldots, k$$

The global minimum is obtained at J's critical points
$2 \sum_{x_j \in S_i} (m_i - x_j) = 0 \rightarrow$

Correctness of the Algorithm

Proposition (convergence): After a finite number of steps, neither the Assignment nor the Update steps modify the output of the algorithm (S_i's).

Proof:

1. "Assignment" decreases J.If a point x_q was misplaced in S_k, i.e., $||x_q - m_k|| \geq ||x_q - m_j||$, for some $j \in \{1, \ldots, k\}$ then it will be placed in S_j and this way $J(S_i, m_i)$ will decrease.

2. "Update" decreases J.Notice that J is convex and thus a local minimum is a global minimum.

$$\frac{\partial J}{\partial m_i} = 2 \sum_{x_j \in S_i} (m_i - x_j), i = 1, \ldots, k$$

   The global minimum is obtained at J's critical points $2\sum_{x_j \in S_i}(m_i - x_j) = 0 \rightarrow \sum_{x_j \in S_i} m_i = \sum_{x_j \in S_i} x_j \rightarrow$

Correctness of the Algorithm

Proposition (convergence): After a finite number of steps, neither the Assignment nor the Update steps modify the output of the algorithm (S_i's).

Proof:

1. "Assignment" decreases J.If a point x_q was misplaced in S_k, i.e., $||x_q - m_k|| \geq ||x_q - m_j||$, for some $j \in \{1, \ldots, k\}$ then it will be placed in S_j and this way $J(S_i, m_i)$ will decrease.

2. "Update" decreases J.Notice that J is convex and thus a local minimum is a global minimum.

$$\frac{\partial J}{\partial m_i} = 2 \sum_{x_j \in S_i} (m_i - x_j), i = 1, \ldots, k$$

The global minimum is obtained at J's critical points $2 \sum_{x_j \in S_i} (m_i - x_j) = 0 \rightarrow \sum_{x_j \in S_i} m_i = \sum_{x_j \in S_i} x_j \rightarrow$

$$m_i = \frac{\sum_{x_j \in S_i} x_j}{\sum_{x_j \in S_i} 1} = \text{mean of } S_i$$

Correctness of the Algorithm

2. Since the point is a global minimum, the update step will decrease the value of $J(S_i, m_i)$.
3. Finally since the search space is finite and since every step decreases the value of the $J(S_i, m_i)$, the algorithm will be convergent.

Fuzzy c-Means

- A crisp partition of a set does not allow partial membership degrees of a point in a cluster.
- Often it is convenient to have soft boundaries for clusters because e.g., a given point cannot be harshly categorized as belonging to a cluster or another.
- To allow partial membership to the clusters, these will need to be fuzzy sets.
- Based on this idea and based on the classical k-means algorithm, Bezdek proposed the fuzzy c-means algorithm.

Fuzzy Partitions

Let $X = \{x_1, \ldots, x_n\} \subseteq \mathbb{R}^n$. We say that $u_1, \ldots, u_c, (c \leq n)$ is a fuzzy partition of X if, for each $x_k \in X$,

$$\sum_{i=1}^{c} u_{ik} = 1$$

for each where $u_{ik} = u_i(x_k)$ is the membership degree of x_k in partition u_i.

Fuzzy Clustering Problem

Let $X = \{x_1, \ldots, x_n\} \subseteq \mathbb{R}^n$ be a data set. Find a fuzzy partition $u_1, \ldots, u_c, (c \leq n)$ of X such that

$$J(u, c) = \sum_{i=1}^{c} \sum_{k=1}^{n} u_{ik}^m ||x_k - c_i||^2$$

is minimized, where

$$c_i = \frac{\sum_{k=1}^{n} u_{ik}^m x_k}{\sum_{k=1}^{n} u_{ik}^m}$$

is the center of the ith cluster, $i = 1, \ldots, c$. Note that u_{ik} can be treated as a matrix.

Fuzzy c-Mean Algorithm - Textbook "Mathematics of Fuzzy Sets and Fuzzy Logic"

Fuzzy c-means algorithm
Assignment:

$$u_{ik} = \frac{1}{\sum_{j=1}^{c} \left(\frac{d_{ik}}{d_{jk}}\right)^{\frac{2}{m-1}}}, i = 1, ..., c$$

where

$$d_{ik} = \|x_k - c_i\|, i = 1, ..., c, k = 1, ..., n$$

and the norm is the Euclidean norm in $\mathbb{R}^n$ (however other norms could be also considered).

Update:

$$c_i = \frac{\sum_{k=1}^{n} (u_{ik})^m x_k}{\sum_{k=1}^{n} (u_{ik})^m}, i = 1, ..., c.$$

- Points on the edge of a cluster can be thought of as belonging to the cluster to a lesser degree than points in the center of the cluster.
- m is called the "fuzzifier".
- In the assignment step

$$u_{ik} = \frac{1}{\sum_{j=1}^{c} \left(\frac{||x_k - c_i||}{||x_k - c_j||} \right)^{\frac{2}{m-1}}}$$

  a large m results in a smaller membership u_{ik} and hence "fuzzier" clusters.
- In the absence of domain knowledge, m is commonly set to 2 (Wikipedia).

Demo

Continuous Fuzzy c-Means

Fuzzy Clustering Problem. Given $\Omega \subseteq \mathbb{R}^n$, a region, and $c \in \mathbb{N}$, and $m > 1$, find a fuzzy partition $A_1, ..., A_c$ of X, i.e., fuzzy sets on X that fulfill the property

$$\sum_{i=1}^{c} A_i(x) = 1,$$

such that the functional

$$J(u, \boldsymbol{c}) = \int_{\Omega} \sum_{i=1}^{c} (A_i(x))^m \|x - \boldsymbol{c}_i\|^2 \, dx,$$

is minimized, where

$$\boldsymbol{c}_i = \frac{\int_{\Omega} (A_i(x))^m \, x dx}{\int_{\Omega} (A_i(x))^m \, dx}.$$

is the center of the $i - th$ cluster, $i = 1, ..., c$.

Continuous Fuzzy c-Means Algorithm

Assignment:

$$A_i(x) = \frac{1}{\sum_{j=1}^{c}\left(\frac{\|x-v_i\|}{\|x-v_j\|}\right)^{\frac{2}{m-1}}}, i = 1, ..., c.$$

The norm can be any norm in $\mathbb{R}^n$ (however in the present discussion we use the Euclidean norm).

Update:

$$v_i = \frac{\int_\Omega (A_i(x))^m \, x dx}{\int_\Omega (A_i(x))^m}, i = 1, ..., c.$$