Introduction to Machine Learning

K-Means, K-Meoids, K-Centers and Variations

Andres Mendez-Vazquez

September 8, 2025

Outline

Outline

The Hardness of K-means clustering

Definition

- Given a multiset $S \subseteq \mathbb{R}^d$, an integer k and $L \in \mathbb{R}$, is there a subset $T \subset \mathbb{R}^d$ with $|T| = k$ such that

$$\sum_{\boldsymbol{x} \in S} \min_{\boldsymbol{t} \in T} \|\boldsymbol{x} - \boldsymbol{t}\|^2 \leq L?$$

The Hardness of K-means clustering

Definition

- Given a multiset $S \subseteq \mathbb{R}^d$, an integer k and $L \in \mathbb{R}$, is there a subset $T \subset \mathbb{R}^d$ with $|T| = k$ such that

$$\sum_{\boldsymbol{x} \in S} \min_{\boldsymbol{t} \in T} \|\boldsymbol{x} - \boldsymbol{t}\|^2 \leq L?$$

Theorem

- The k-means clustering problem is NP-complete even for $d = 2$.

Reduction

The reduction to an NP-Complete problem

- Exact Cover by 3-Sets problem

Reduction

The reduction to an NP-Complete problem

- Exact Cover by 3-Sets problem

Definition

- Given a finite set U containing exactly $3n$ elements and a collection $\mathcal{C} = \{S_1, S_2, ..., S_l\}$ of subsets of U each of which contains exactly 3 elements, Are there n sets in $\mathcal{C}$ such that their union is U?

However

There are efficient heuristic and approximation algorithms

- Which can solve this problem

Outline

K-Means - Stuart Lloyd(Circa 1957)

History

Invented by Stuart Loyd in Bell Labs to obtain the best quantization in a signal data set.

K-Means - Stuart Lloyd(Circa 1957)

History

Invented by Stuart Loyd in Bell Labs to obtain the best quantization in a signal data set.

Something Notable

The paper was published until 1982

K-Means - Stuart Lloyd(Circa 1957)

History

Invented by Stuart Loyd in Bell Labs to obtain the best quantization in a signal data set.

Something Notable

The paper was published until 1982

Basically given N vectors $\boldsymbol{x}_1, ..., \boldsymbol{x}_N \in \mathbb{R}^d$

It tries to find k points $\boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_k \in \mathbb{R}^d$ that minimize the expression (i.e. a partition S of the vector points):

$$\sum_{k=1}^{N} \sum_{i:\boldsymbol{x}_i \in C_k} \left\| \boldsymbol{x}_i - \boldsymbol{\mu}_k \right\|^2 = \sum_{k=1}^{N} \sum_{i:\boldsymbol{x}_i \in C_k} \left(\boldsymbol{x}_i - \boldsymbol{\mu}_k \right)^T \left(\boldsymbol{x}_i - \boldsymbol{\mu}_k \right)$$

K-means clustering

K-means

It is a partitional clustering algorithm.

K-means clustering

K-means

It is a partitional clustering algorithm.

Definition

Let the set of data points (or instances) D be $\{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$ where $\mathbf{x}_i = (x_{i1}, \cdots, x_{ir})^T$:

K-means clustering

K-means

It is a partitional clustering algorithm.

Definition

Let the set of data points (or instances) D be $\{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$ where $\mathbf{x}_i = (x_{i1}, \cdots, x_{ir})^T$:

- The K-means algorithm partitions the given data into K clusters.

K-means clustering

K-means

It is a partitional clustering algorithm.

Definition

Let the set of data points (or instances) D be $\{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$ where $\mathbf{x}_i = (x_{i1}, \cdots, x_{ir})^T$:

- The K-means algorithm partitions the given data into K clusters.
- Each cluster has a cluster center, called centroid.

K-means clustering

K-means

It is a partitional clustering algorithm.

Definition

Let the set of data points (or instances) D be $\{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$ where $\mathbf{x}_i = (x_{i1}, \cdots, x_{ir})^T$:

- The K-means algorithm partitions the given data into K clusters.
- Each cluster has a cluster center, called centroid.
- K is specified by the user.

K-means algorithm

The K-means algorithm works as follows

Given k as the possible number of cluster:

K-means algorithm

The K-means algorithm works as follows

Given k as the possible number of cluster:

1. Randomly choose K data points (seeds) to be the initial centroids, cluster centers,
   - $\{\mathbf{v}_1, \cdots, \mathbf{v}_k\}$

K-means algorithm

The K-means algorithm works as follows

Given k as the possible number of cluster:

1. Randomly choose K data points (seeds) to be the initial centroids, cluster centers,
   - $\{\mathbf{v}_1, \cdots, \mathbf{v}_k\}$
2. Assign each data point to the closest centroid
   - $c_i = \arg\min_j \{dist(\mathbf{x}_i - \mathbf{v}_j)\}$

K-means algorithm

The K-means algorithm works as follows

Given k as the possible number of cluster:

1. Randomly choose K data points (seeds) to be the initial centroids, cluster centers,
   - $\{\mathbf{v}_1, \cdots, \mathbf{v}_k\}$
2. Assign each data point to the closest centroid
   - $c_i = \arg\min_j \{dist(\mathbf{x}_i - \mathbf{v}_j)\}$
3. Re-compute the centroids using the current cluster memberships.
   - $\mathbf{v}_j = \dfrac{\sum_{i=1}^{n} I(c_i = j)\mathbf{x}_i}{\sum_{i=1}^{n} I(c_i = j)}$

K-means algorithm

The K-means algorithm works as follows

Given k as the possible number of cluster:

1. Randomly choose K data points (seeds) to be the initial centroids, cluster centers,
   - $\{\mathbf{v}_1, \cdots, \mathbf{v}_k\}$
2. Assign each data point to the closest centroid
   - $c_i = \arg\min_j \{dist(\mathbf{x}_i - \mathbf{v}_j)\}$
3. Re-compute the centroids using the current cluster memberships.
   - $\mathbf{v}_j = \dfrac{\displaystyle\sum_{i=1}^{n} I(c_i = j)\mathbf{x}_i}{\displaystyle\sum_{i=1}^{n} I(c_i = j)}$
4. If a convergence criterion is not met, go to 2.

What is the code trying to do?

It is trying to find a partition S

K-means tries to find a partition S such that it minimizes the cost function:

$$\min_{S} \sum_{k=1}^{N} \sum_{i:\boldsymbol{x}_i \in C_k} \left(\boldsymbol{x}_i - \boldsymbol{\mu}_k\right)^T \left(\boldsymbol{x}_i - \boldsymbol{\mu}_k\right) \tag{1}$$

What is the code trying to do?

It is trying to find a partition S

K-means tries to find a partition S such that it minimizes the cost function:

$$\min_{S} \sum_{k=1}^{N} \sum_{i:\boldsymbol{x}_i \in C_k} \left(\boldsymbol{x}_i - \boldsymbol{\mu}_k\right)^T \left(\boldsymbol{x}_i - \boldsymbol{\mu}_k\right) \tag{1}$$

What is the code trying to do?

It is trying to find a partition S

K-means tries to find a partition S such that it minimizes the cost function:

$$\min_{S} \sum_{k=1}^{N} \sum_{i:\boldsymbol{x}_i \in C_k} \left(\boldsymbol{x}_i - \boldsymbol{\mu}_k\right)^T \left(\boldsymbol{x}_i - \boldsymbol{\mu}_k\right) \tag{1}$$

Where μ_k is the centroid for cluster C_k

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{i:\boldsymbol{x}_i \in C_k} \boldsymbol{x}_i \tag{2}$$

Where N_k is the number of samples in the cluster C_k.

Outline

What Stopping/convergence criterion should we use?

First

No (or minimum) re-assignments of data points to different clusters.

What Stopping/convergence criterion should we use?

First

No (or minimum) re-assignments of data points to different clusters.

Second

No (or minimum) change of centroids.

What Stopping/convergence criterion should we use?

First

No (or minimum) re-assignments of data points to different clusters.

Second

No (or minimum) change of centroids.

Third

Minimum decrease in the sum of squared error (SSE),

What Stopping/convergence criterion should we use?

First

No (or minimum) re-assignments of data points to different clusters.

Second

No (or minimum) change of centroids.

Third

Minimum decrease in the sum of squared error (SSE),

- C_k is cluster k.

What Stopping/convergence criterion should we use?

First

No (or minimum) re-assignments of data points to different clusters.

Second

No (or minimum) change of centroids.

Third

Minimum decrease in the sum of squared error (SSE),

- C_k is cluster k.
- $\mathbf{v}_k$ is the centroid of cluster C_k.

$$SSE = \sum_{k=1}^{K} \sum_{x \in c_k} dist\left(\mathbf{x}, \mathbf{v}_k\right)^2$$

Outline

The distance function

Actually, we have the following distance functions:

Euclidean

$$dist(\mathbf{x}, \mathbf{y}) = ||\mathbf{x} - \mathbf{y}|| = \sqrt{(\mathbf{x} - \mathbf{y})^T (\mathbf{x} - \mathbf{y})}$$

The distance function

Actually, we have the following distance functions:

Euclidean

$$dist(\mathbf{x}, \mathbf{y}) = ||\mathbf{x} - \mathbf{y}|| = \sqrt{(\mathbf{x} - \mathbf{y})^T (\mathbf{x} - \mathbf{y})}$$

Manhattan

$$dist(\mathbf{x}, \mathbf{y}) = ||\mathbf{x} - \mathbf{y}||_1 = \sum_{i=1}^{n} |x_i - y_i|$$

The distance function

Actually, we have the following distance functions:

Euclidean

$$dist(\mathbf{x}, \mathbf{y}) = ||\mathbf{x} - \mathbf{y}|| = \sqrt{(\mathbf{x} - \mathbf{y})^T (\mathbf{x} - \mathbf{y})}$$

Manhattan

$$dist(\mathbf{x}, \mathbf{y}) = ||\mathbf{x} - \mathbf{y}||_1 = \sum_{i=1}^{n} |x_i - y_i|$$

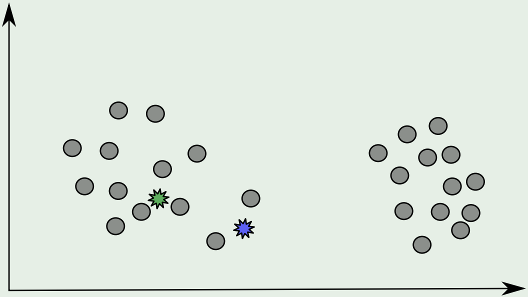
Mahalanobis

$$dist(\mathbf{x}, \mathbf{y}) = ||\mathbf{x} - \mathbf{y}||_A = \sqrt{(\mathbf{x} - \mathbf{y})^T A (\mathbf{x} - \mathbf{y})}$$

Outline

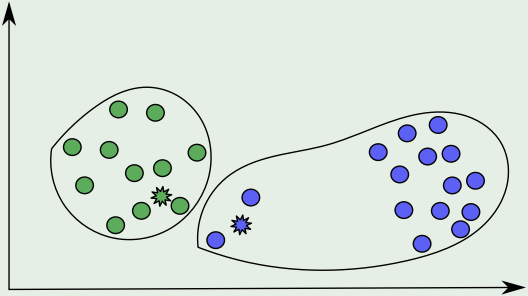
An example

Dropping two possible centroids



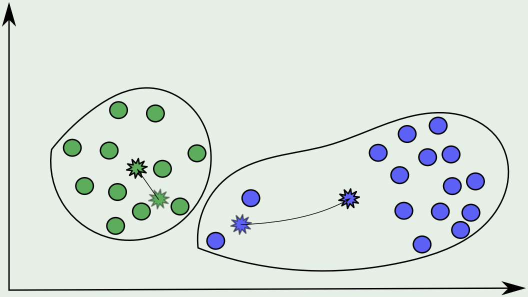
An example

Calculate the memberships



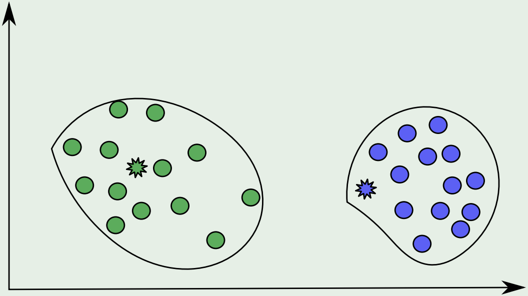
An example

We re-calculate centroids



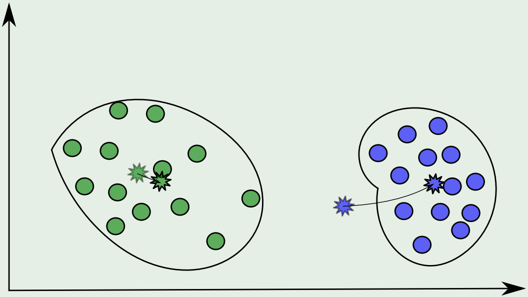
An example

We re-calculate memberships



An example

We re-calculate centroids and keep going



Outline

Strengths of K-means

Strengths

- Simple: easy to understand and to implement

Strengths of K-means

Strengths

- Simple: easy to understand and to implement
- Efficient: Time complexity: $O(tKN)$, where N is the number of data points, K is the number of clusters, and t is the number of iterations.

Strengths of K-means

Strengths

- Simple: easy to understand and to implement
- Efficient: Time complexity: $O(tKN)$, where N is the number of data points, K is the number of clusters, and t is the number of iterations.
- Since both K and t are small. K-means is considered a linear algorithm.

Strengths of K-means

Strengths

- Simple: easy to understand and to implement
- Efficient: Time complexity: $O(tKN)$, where N is the number of data points, K is the number of clusters, and t is the number of iterations.
- Since both K and t are small. K-means is considered a linear algorithm.

Popularity

K-means is the most popular clustering algorithm.

Strengths of K-means

Strengths

- Simple: easy to understand and to implement
- Efficient: Time complexity: $O(tKN)$, where N is the number of data points, K is the number of clusters, and t is the number of iterations.
- Since both K and t are small. K-means is considered a linear algorithm.

Popularity

K-means is the most popular clustering algorithm.

Note that

It terminates at a local optimum if SSE is used. The global optimum is hard to find due to complexity.

Weaknesses of K-means

Important

The algorithm is only applicable if the mean is defined.

Weaknesses of K-means

Important

The algorithm is only applicable if the mean is defined.

- For categorical data, K-mode - the centroid is represented by most frequent values.

Weaknesses of K-means

Important

The algorithm is only applicable if the mean is defined.

- For categorical data, K-mode - the centroid is represented by most frequent values.

In addition

The user needs to specify K.

Weaknesses of K-means

Important

The algorithm is only applicable if the mean is defined.

- For categorical data, K-mode - the centroid is represented by most frequent values.

In addition

The user needs to specify K.

Outliers

The algorithm is sensitive to outliers.

Weaknesses of K-means

Important

The algorithm is only applicable if the mean is defined.

- For categorical data, K-mode - the centroid is represented by most frequent values.

In addition

The user needs to specify K.

Outliers

The algorithm is sensitive to outliers.

- Outliers are data points that are very far away from other data points.

Weaknesses of K-means

Important

The algorithm is only applicable if the mean is defined.

- For categorical data, K-mode - the centroid is represented by most frequent values.

In addition

The user needs to specify K.

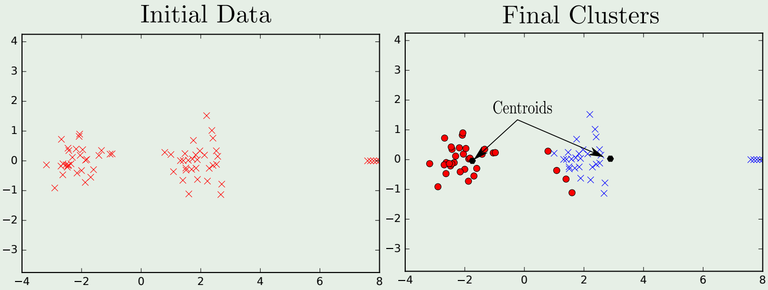
Outliers

The algorithm is sensitive to outliers.

- Outliers are data points that are very far away from other data points.
- Outliers could be errors in the data recording or some special data points with very different values.

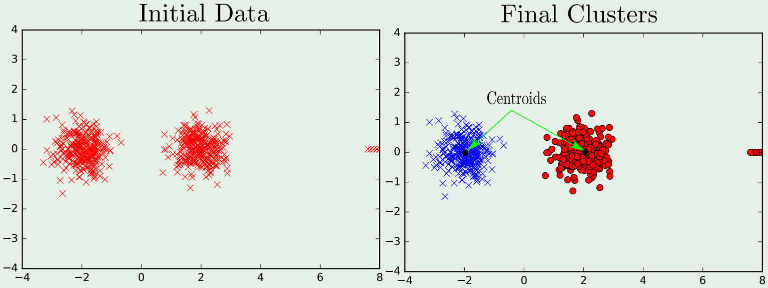
Weaknesses of K-means: Problems with outliers

A series of outliers



Weaknesses of K-means: Problems with outliers

Nevertheless, if you have more dense clusters



Weaknesses of K-means: How to deal with outliers

One method

To remove some data points in the clustering process that are much further away from the centroids than other data points.

Weaknesses of K-means: How to deal with outliers

One method

To remove some data points in the clustering process that are much further away from the centroids than other data points.

- To be safe, we may want to monitor these possible outliers over a few iterations and then decide to remove them.

Weaknesses of K-means: How to deal with outliers

One method

To remove some data points in the clustering process that are much further away from the centroids than other data points.

- To be safe, we may want to monitor these possible outliers over a few iterations and then decide to remove them.

Another method

To perform random sampling.

Weaknesses of K-means: How to deal with outliers

One method

To remove some data points in the clustering process that are much further away from the centroids than other data points.

- To be safe, we may want to monitor these possible outliers over a few iterations and then decide to remove them.

Another method

To perform random sampling.

- Since in sampling we only choose a small subset of the data points, the chance of selecting an outlier is very small.

Weaknesses of K-means: How to deal with outliers

One method

To remove some data points in the clustering process that are much further away from the centroids than other data points.

- To be safe, we may want to monitor these possible outliers over a few iterations and then decide to remove them.

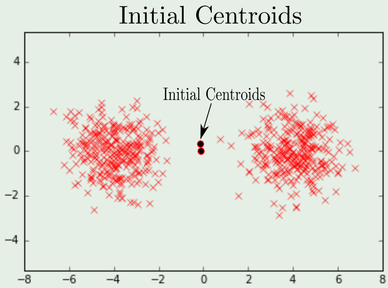
Another method

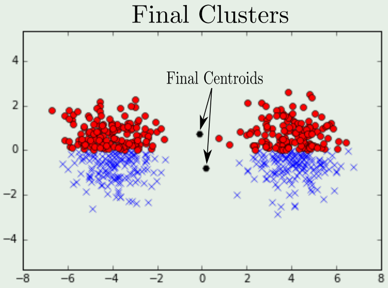
To perform random sampling.

- Since in sampling we only choose a small subset of the data points, the chance of selecting an outlier is very small.
- Assign the rest of the data points to the clusters by distance or similarity comparison, or classification.

Weaknesses of K-means (cont...)

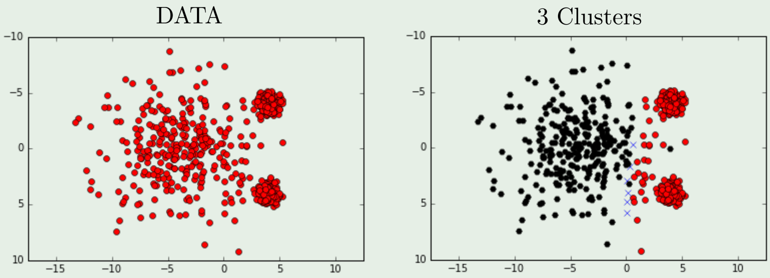
The algorithm is sensitive to initial seeds





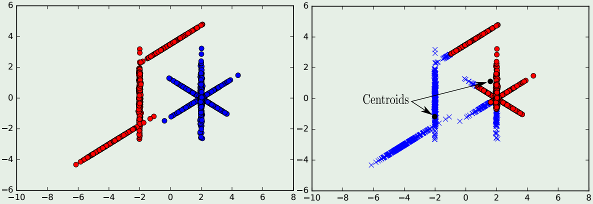
Weaknesses of K-means : Different Densities

We have three cluster nevertheless



Weaknesses of K-means: Non-globular Shapes

Here, we notice that K-means may only detect globular shapes

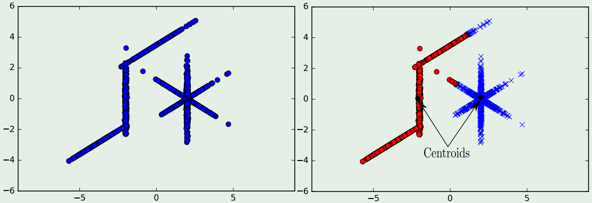


NO GLOBULAR DATA

K-Means Results

Weaknesses of K-means: Non-globular Shapes

However, it sometimes work better than expected



NO GLOBULAR DATA

K-Means Results

Outline

Until now, we have assumed a Euclidean metric space

Important step

- The cluster representatives $m_1, ..., m_k$ in are taken to be the means of the currently assigned clusters.

Until now, we have assumed a Euclidean metric space

Important step

- The cluster representatives $m_1, ..., m_k$ in are taken to be the means of the currently assigned clusters.

We can generalize this by using a dissimilarity $D\left(\boldsymbol{x}_i, \boldsymbol{x}_{i'}\right)$

- By using an explicit optimization with respect to $m_1, ..., m_k$

Outline

Algorithm K-meoids

Step 1

- For a given cluster assignment C find the observation in the cluster minimizing total distance to other points in that cluster:

$$i_k^* = \arg \min_{\{i|C(i)=k\}} \sum_{C(i')=k} D\left(\boldsymbol{x}_i, \boldsymbol{x}_{i'}\right)$$

  - Then $m_k = \boldsymbol{x}_{i_k^*}$ $k = 1, ..., K$ are the current estimates of the cluster centers.

Now

Step 2

- Given a current set of cluster centers $m_1, ..., m_k$, minimize the total error by assigning each observation to the closest (current) cluster center:

$$C(i) = \arg\min_{1 \leq k \leq K} D(\boldsymbol{x}_i, m_k)$$

Now

Step 2

- Given a current set of cluster centers $m_1, ..., m_k$, minimize the total error by assigning each observation to the closest (current) cluster center:

$$C(i) = \arg \min_{1 \leq k \leq K} D(\boldsymbol{x}_i, m_k)$$

Iterate over steps 1 and 2

- Until the assignments do not change.

Outline

Complexity

Problem, solving the first step has a complexity for $k = 1, ..., K$

$$O\left(N_k^2\right)$$

Complexity

Problem, solving the first step has a complexity for $k = 1, ..., K$

$$O\left(N_k^2\right)$$

Given a set of cluster "centers," $\{i_1, i_2, ..., i_K\}$

- Given the new assignments

$$C(i) = \arg\min_{1 \leq k \leq K} D(\boldsymbol{x}_i, m_k)$$

  - It requires a complexity of $O(KN)$ as before.

Therefore

We have that

- K-medoids is more computationally intensive than K-means.