Lecture 9: The k-means algorithm

Introduction to Machine Learning

Sophie Robert

L3 MIASHS — Semestre 2

2022-2023

Reminder on previous session

Question

Can anyone remind me what is the definition of **unsupervised learning ?**

Principle

K-Means algorithm

The k-means algorithm* (*MacQueen, 1967*) is a clustering algorithm that partitions the space into k cluster by minimizing the *within-cluster variance*.

Principle

K-Means algorithm

The k-means algorithm* (*MacQueen, 1967*) is a clustering algorithm that partitions the space into k cluster by minimizing the *within-cluster variance*.

Given a set of individuals described by their features $(X_1, \ldots, X_n)$ find k sets to partition the data into by minimzing *the within cluster variance*.

$$\sum_{i=1}^{k} \sum_{X \in S_i} = ||X - \mu_i||^2$$

with:

$$\mu_i = \frac{1}{|S_i|} \sum_{X \in S_i} X$$

(μ_i is the mean or centroid)

Medoids and centroids

Centroids

A centroid* is the arithmetic mean of a cluster, that is most often **not part of the dataset**.

Medoids and centroids

Centroids

A centroid* is the arithmetic mean of a cluster, that is most often **not part of the dataset**.

Medoids

A medoid* is a **member of the dataset** which sum of dissimilarities to all the objects in the cluster is minimal.

The k-means algorithm

In practice, problem is NP-hard, so we rely on **Lloyd's iterative algorithm**:

The k-means algorithm

In practice, problem is NP-hard, so we rely on **Lloyd's iterative algorithm**:
Given a set of k means $m_1^{(1)}, \dots, m_k^{(1)}$, iteratively perform two steps:

The k-means algorithm

In practice, problem is NP-hard, so we rely on **Lloyd's iterative algorithm**:
Given a set of k means $m_1^{(1)}, \ldots, m_k^{(1)}$, iteratively perform two steps:

1. **Assignment step**: Assign each observation to the cluster with the nearest mean using the **Euclidean distance**.

The k-means algorithm

In practice, problem is NP-hard, so we rely on **Lloyd's iterative algorithm**:
Given a set of k means $m_1^{(1)}, \ldots, m_k^{(1)}$, iteratively perform two steps:

1. **Assignment step**: Assign each observation to the cluster with the nearest mean using the **Euclidean distance**.
2. **Update step**: Recalculate the mean for each cluster.

The k-means algorithm

In practice, problem is NP-hard, so we rely on **Lloyd's iterative algorithm**:
Given a set of k means $m_1^{(1)}, \dots, m_k^{(1)}$, iteratively perform two steps:

1. **Assignment step**: Assign each observation to the cluster with the nearest mean using the **Euclidean distance**.
2. **Update step**: Recalculate the mean for each cluster.

Run steps until assignment do not change.

The k-means algorithm

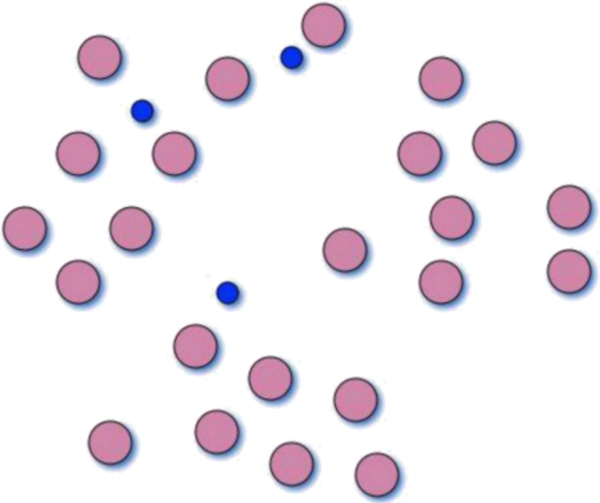
In practice, problem is NP-hard, so we rely on **Lloyd's iterative algorithm**:
Given a set of k means $m_1^{(1)}, \ldots, m_k^{(1)}$, iteratively perform two steps:

1. **Assignment step**: Assign each observation to the cluster with the nearest mean using the **Euclidean distance**.
2. **Update step**: Recalculate the mean for each cluster.

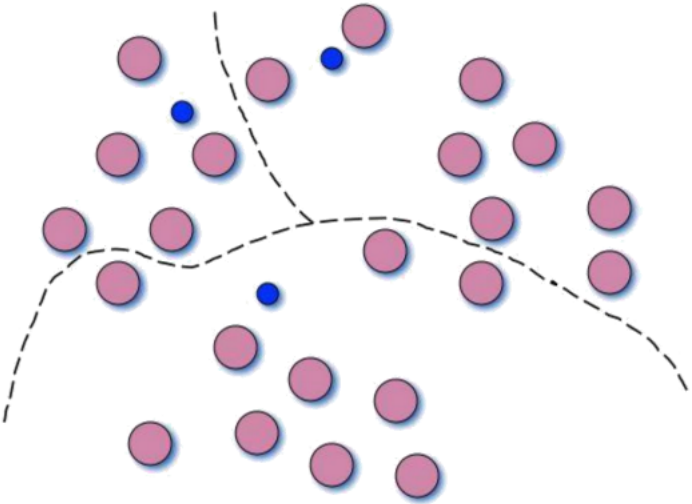
Run steps until assignment do not change.

There is no garantee to find the optimum (but efficient in practice).

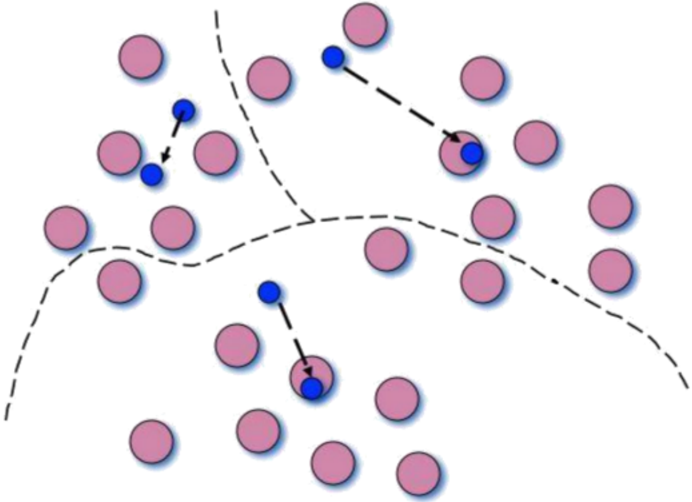
Initialization



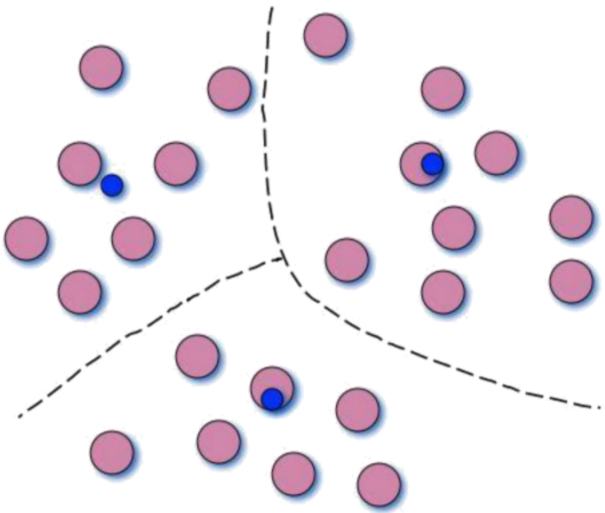
Assign each individual to a cluster



Compute new medoids



Repeat until stable



Initialization

Most common initialization for the algorithm:

- **Fully random approach**: randomly choose k vectors in the feature space.

Initialization

Most common initialization for the algorithm:

- **Fully random approach**: randomly choose k vectors in the feature space.
- **Forgy partition**: Randomly choose k observations from the dataset.

Initialization

Most common initialization for the algorithm:

- **Fully random approach**: randomly choose k vectors in the feature space.
- **Forgy partition**: Randomly choose k observations from the dataset.
- **Random partition**: Randomly assign a cluster to each observation.

Example: k-means algorithm

Questions

1. Perform the k-means algorithm on the following dataset for k=2
2. Assign each individual to a cluster
3. Give coordinates of each centroid

ID	Sepal length	Sepal width
1	5	2
2	5	3
3	4	3
4	7	4
5	6	5

Hyperparameters

Question

What are the hyperparameters of the algorithm ?

Hyperparameters

Question

What are the hyperparameters of the algorithm ?

k !

Selecting the number of clusters using the elbow method

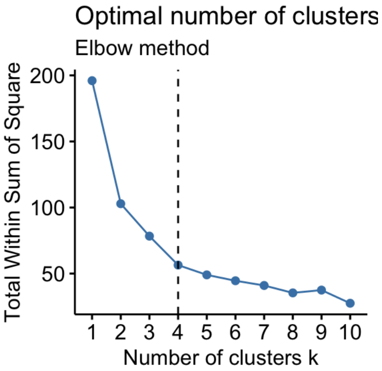
Elbow method

An elbow plot* is a visual method by plotting the *within cluster variance* against the number of clusters and selecting the number of clusters before the curve flattens.

Selecting the number of clusters using the elbow method

Elbow method

An elbow plot* is a visual method by plotting the *within cluster variance* against the number of clusters and selecting the number of clusters before the curve flattens.

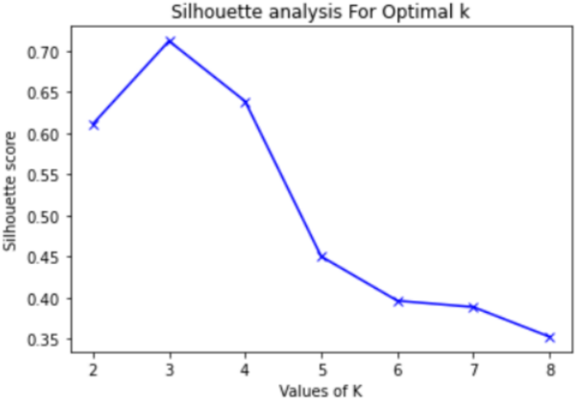


Selecting the number of clusters using silhouette score

The silhouette score (see previous lecture) reaches its global maximum for the optimum number of k.

Selecting the number of clusters using silhouette score

The silhouette score (see previous lecture) reaches its global maximum for the optimum number of k.



Line plot between K and Silhouette score

Advantages and drawbacks

Advantages

- Fast to compute
- Easy to understand
- Work very well when clusters have a spherical shape

Advantages and drawbacks

Advantages

- Fast to compute
- Easy to understand
- Work very well when clusters have a spherical shape

Limits

- Random algorithm
- No garantee to not be in a local optimum
- k must be chosen beforehand
- Class representative does not exist making it harder to interprete

Similar algorithm: PAM

Partitioning Around Medoids

Partitioning Around Medoids* (PAM) (*Leonard Kaufman and Peter J. Rousseeuw*) is a clustering algorithm that attempts to minimize the distance between points labeled to be in a cluster and a point designated as the center of that cluster.

Similar algorithm: PAM

Partitioning Around Medoids

Partitioning Around Medoids* (PAM) (*Leonard Kaufman and Peter J. Rousseeuw*) is a clustering algorithm that attempts to minimize the distance between points labeled to be in a cluster and a point designated as the center of that cluster.

Fixes one of the problem of k-means: the *medoid* (instead of centroid) exists in the dataset.

PAM algorithm

The PAM algorithm is iterative:
Given k and a cost function $\sum_{i=1}^{k} \sum_{X \in S_i} = d(X, x^{(i)})$ with $x^{(i)}$ the medoid of cluster i and d a dissimilarity,

PAM algorithm

The PAM algorithm is iterative:
Given k and a cost function $\sum_{i=1}^{k} \sum_{X \in S_i} = d(X, x^{(i)})$ with $x^{(i)}$ the medoid of cluster i and d a dissimilarity,
Initialize: greedily select k of the n data points as the medoids to minimize the cost.

PAM algorithm

The PAM algorithm is iterative:
Given k and a cost function $\sum_{i=1}^{k} \sum_{X \in S_i} = d(X, x^{(i)})$ with $x^{(i)}$ the medoid of cluster i and d a dissimilarity,
Initialize: greedily select k of the n data points as the medoids to minimize the cost.

Until the cost function does not decrease anymore:

1. Associate each non-medoid data point to the closest medoid
2. For each medoid m, and for each non-medoid data point o
   1. Swap m and o
   2. Compute the cost change
   3. If the cost decreases, the store the value for the cost decrease
3. Perform the swap of o and m that decreases the most the cost function

Example: PAM algorithm

Questions

1. Perform the PAM algorithm on the following dataset for k=2
2. Assign each individual to a cluster
3. Give coordinates of each centroid

ID	Sepal length	Sepal width
1	5	2
2	5	3
3	4	3
4	7	4
5	6	5

Advantages and limits

Advantages:

- The medoid is part of the dataset and can easily be interpreted.
- Selected dissimilarity can be customized.

Advantages and limits

Advantages:

- The medoid is part of the dataset and can easily be interpreted.
- Selected dissimilarity can be customized.

Limits:

- We need to decide a value for k
- Algorithm initialization is random

Questions

Questions ?