Machine Learning

Clustering

Karol Przystalski

March 9, 2022

Department of Information Technologies, Jagiellonian University

Agenda

1. Introduction
2. Distributed clustering
3. Density-based clustering
4. Hierachical clustering
5. Quality metrics
6. Other clustering methods

Introduction

What is clustering?

This group of learning methods are also known under different names. It depends on the context where it is used.

Unsupervised learning can be called as learning without a teacher. It is the opposite to learning with a teacher – supervised learning.

Unsupervised learning is also known as partitioning, segmentation, typology, numerical taxonomy or clustering. The last term is one of the most common used aside from unsupervised learning.

Goal

A cluster is a set of elements/objects of the same label. Comparing to supervised methods, the label used here is based on similarities between elements of each cluster. It means that some elements are more similar to some than to other elements.

In other words, the goal of a clustering method is to find groups of objects that are most similar to each other.

Taxonomy

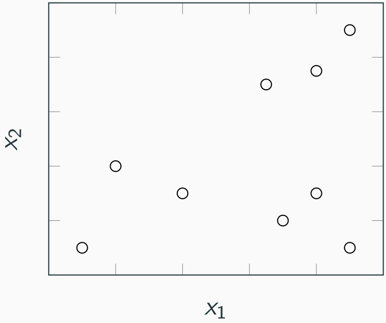
There are three major types of clustering methods:

1. distributed,
2. density-based,
3. hierarchical.

There are more types, but not so popular.

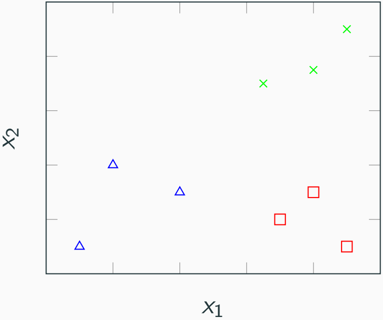
Example

In this figure we have an example data set of elements that we would like to cluster into three groups.



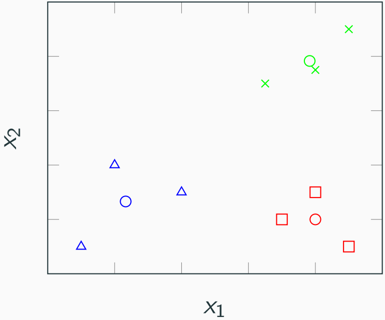
Example

It is obvious that we should have group it as shown in the figure.



Example

The centroids are marked with a circle. Each group has one centroid.



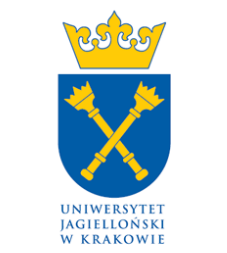
Usage

Clustering methods are widely used. Some application examples are:

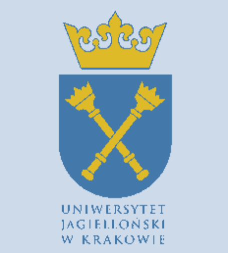
1. customer segmentation,
2. image processing,
3. office localization or city-planning,
4. marketing,
5. and many more.

Image segmentation

Here you can see two examples of how we can use clustering in image segmenatation to distinguish between

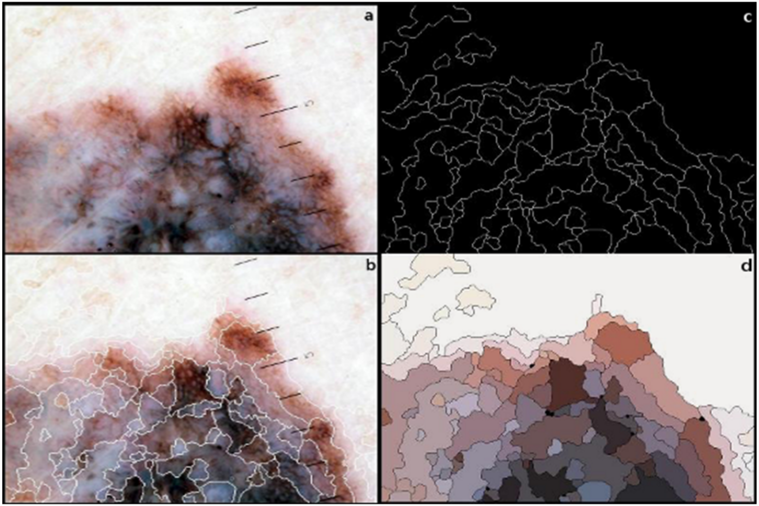


(a)



(b)

More serious about clustering



Distributed clustering

General algorithm

The steps are like following:

1. choose the entrance cluster centroids,
2. calculate the assignation matrix U,
3. calculate new centroids matrix V,
4. calculate the difference between previously assignation matrix U and the new one calculated in current iteration.

General algorithm – details

Choose the entrance cluster centroids

This step is done only once. In most methods the center of each cluster needs to be chosen before the algorithm starts. The two most popular ways to do it, is to set it randomly or set fixed values.

Calculate the assignation matrix U

Assignation matrix calculation step is slightly different in each clustering method. Matrix U consist of c rows and k columns, where c is the number of groups/clusters that we want to have and k is the number of elements in training data set.

General algorithm – details

Calculate new centroids matrix V

Centroids are calculated in most methods in a similar way. The number of groups c_i is the same as the number of centers v_i, where $i = 1, \ldots, c$:

$$V = [v_1, v_2, \ldots, v_c]. \tag{1}$$

Calculate the change rate

We calculate the assignation matrix from the previous step as well as new centroids in each iteration until the differences between the changes in both are small enough.

k-means

K-means is also known as hard c-means algorithm (hcm). The It is one of the simplest clustering method.

The goal of this algorithm is to assign each element in the training data set into a cluster in a binary way. It means that an element can be assigned to only one cluster fully. It is a strict (hard) way of assignation.

k-means – assignation matrix

In k-means the assignation matrix can look like following:

$$U = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \end{bmatrix}.$$

An element can be assigned to one of two classes in this case.

The assignation is done in a simple way. For each object x_k we measure the distance from it to each center. The closest distance wins:

$$\mu_{ik}^{(t)} = \begin{cases} 1 & \text{if } d(x_k, v_i) < d(x_k, v_j), \text{for each } j \neq i \\ 0 & \text{in other case} \end{cases}. \tag{2}$$

Centroids

Each group center is calculated separately as following:

$$v_i = \frac{\sum_{k=1}^{M} \mu_{ik}^{(t)} x_k}{\sum_{k=1}^{M} \mu_{ik}^{(t)}}. \tag{3}$$

At the end we should get an array of group centroids:

$$V = [v_1, v_2, \ldots, v_c]. \tag{4}$$

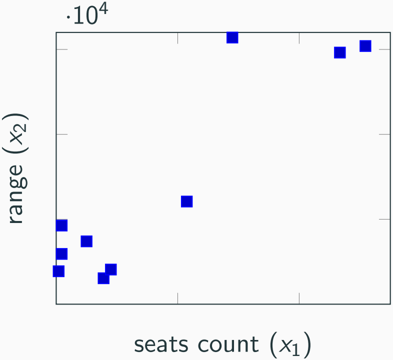
Example – Aircrafts

Let’s take an example of aircrafts

Aircraft name	Distance range (km)	Seats count	Aircraft type
Cesna 510 Mustang	1940	4	private jet
Falcon 10/100	2960	9	private jet
Hawker 900/900XP	4630	9	private jet
ATR 72-600	1528	78	medium size aircraft
Bombardier Dash 8 Q400	2040	90	medium size aircraft
Embraer ERJ145 XR	3700	50	medium size aircraft
Boeing 747-8	14815	467	jet airliner
A380-800	15200	509	jet airliner
Boeing 787-8	15700	290	jet airliner
Boeing 737-900ER	6045	215	jet airliner

Example – Aircrafts

We can draw the example as following:



Normalization

We take the highest values of each column and divide each element by these values. We get a data set X:

$$X = \begin{bmatrix} 0.00785855 & 0.12356688 \\ 0.01768173 & 0.18853503 \\ 0.01768173 & 0.29490446 \\ 0.15324165 & 0.09732484 \\ 0.17681729 & 0.12993631 \\ 0.09823183 & 0.23566879 \\ 0.91748527 & 0.94363057 \\ 1. & 0.96815287 \\ 0.5697446 & 1. \\ 0.42239686 & 0.38503185 \end{bmatrix}$$

Calculation of k-means – one step

Our assignation/membership matrix looks like:

$$U = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

We choose randomly centroids. Let it be

$$V = \begin{bmatrix} 0.9 & 0.5 \\ 0.4 & 0.2 \end{bmatrix}.$$

The next step is to calculate the new membership matrix column by column:

$$\mu_{ik}^{(t)} = \begin{cases} 1 & \text{if } d(x_k, v_i) < d(x_k, v_j), \text{for each } j \neq i \\ 0 & \text{in other case} \end{cases}. \tag{5}$$

Calculation of k-means – one step

One column can be calculated as follows:

$$d(x_1, v_1) = \sqrt{(x_{11}^2 - v_{11}^2)^2 + (x_{12} - v_{12})^2}$$
$$= \sqrt{(0.0078 - 0.9)^2 + (0.1235 - 0.5)^2} =$$
$$\sqrt{0.7960 + 0.1417} = 0.96837$$

$$d(x_1, v_2) = \sqrt{(x_{11}^2 - v_{21}^2)^2 + (x_{12} - v_{22})^2} =$$
$$\sqrt{(0.0078 - 0.4)^2 + (0.1235 - 0.2)^2} =$$
$$\sqrt{0.1036 + 0.0058} = 0.3308$$

After the first row we get the assignation matrix:

$$U = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Calculation of k-means – one step

The full assignation matrix after step one looks like following:

$$U = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Centroids are calculated as following:

$$v_i = \frac{\sum_{k=1}^{M} \mu_{ik}^{(t)} x_k}{\sum_{k=1}^{M} \mu_{ik}^{(t)}}. \tag{6}$$

The first centroid is calculated as following:

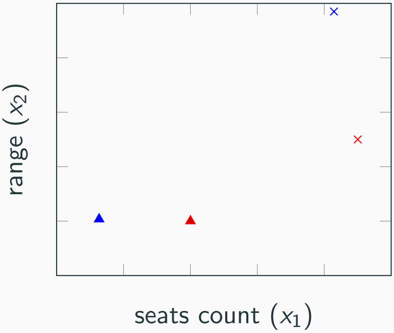
$$\begin{aligned} v_i &= \frac{\begin{bmatrix} 0.917 \\ 0.943 \end{bmatrix} + \begin{bmatrix} 1 \\ 0.968 \end{bmatrix} + \begin{bmatrix} 0.569 \\ 1 \end{bmatrix}}{3} \\ &= \frac{\begin{bmatrix} 2.48722986 \\ 2.911 \end{bmatrix}}{3} = \begin{bmatrix} 0.829 & 0.970 \end{bmatrix} \end{aligned}$$

Calculation of k-means – one step

The centroids after step one are as following:

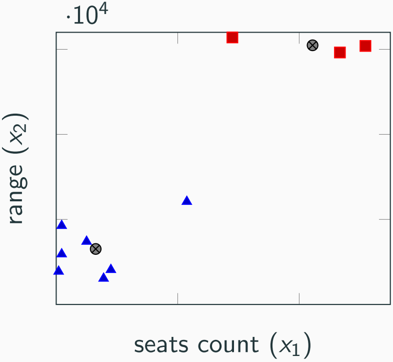
$$V = \begin{bmatrix} 0.829 & 0.970 \\ 0.127 & 0.207 \end{bmatrix}.$$

The move can be drawn as shown in figure below.



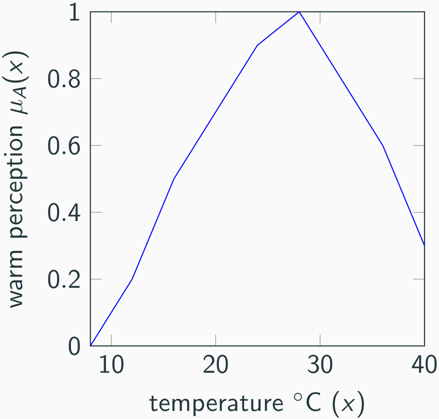
Example

The obvious clustering into two groups would look like in the figure below. The gray circles are the centroids of each group.



Assignation function – fuzzy sets

In fuzzy sets we have an assignation function $\mu_A(x)$ that assigns a value based on the current fuzzy set. If we take the temperature of the weather it could look like following:



Fuzzy

In fuzzy k-means (fcm) the assignation function is different:

$$\mu_{ik} = (\sum_{j=1}^{c}(\frac{d(x_k, v_i)}{d(_k, v_j)})^{\frac{2}{m-1}})^{-1} \tag{7}$$

It assigns a value from 0 to 1 to a group, but the sum of the values cannot be greater than 1.

The centers are also calculated a bit differently:

$$v_i = \frac{\sum_{k=1}^{M}(\mu_{ik}^{(t)})^m x_k}{\sum_{k=1}^{M}(\mu_{ik}^{(t)})^m} \tag{8}$$

. An example assignation matrix can look in case of FCM like following:

$$U = \begin{bmatrix} 0.45 & 0.65 & 0.3 & 0.9 & 0.25 \\ 0.55 & 0.35 & 0.7 & 0.1 & 0.75 \end{bmatrix}.$$

Fuzzy k-means

In case of fuzzy k-means we have a differene in the membership calculation. For the centers as shown in previous example, the two values of membership for the first element in the data set, can be calculated as follows:

$$\mu_{11} = (1 + (\frac{d(x_1, v_1)}{d(x_1, v_2)})^2)^{-1},$$

where $x_1 = \begin{bmatrix} 0.0078 \\ 0.1235 \end{bmatrix}$, $v_1 = \begin{bmatrix} 0.9 \\ 0.4 \end{bmatrix}$ and $v_2 = \begin{bmatrix} 0.5 \\ 0.2 \end{bmatrix}$. It gives us $\mu_{11} = 4.5168^{-1} = 0.2139$ and $\mu_{12} = 0.77860$.

Possibilistic

The possibilistic k-means (PCM) is a bit more complex and the assignation function takes

$$\mu_{ik} = (1 + (\frac{D_{ikA}}{\eta_i})^{\frac{2}{m-1}})^{-1}, \tag{9}$$

where possibilistic distribution measure:

$$\eta_i = \frac{\sum_{k=1}^{M} (\mu_{ik})^m D_{ikA}^2}{\sum_{k=1}^{M} (\mu_{ik})^m}, \tag{10}$$

and

$$D_{ikA}^2 = ||x_k - v_i||_A^2 = (x_k - v_i)^T A (x_k - v_i), \tag{11}$$

where A is a diagonal matrix.

An example assignation matrix for PCM can look like following:

$$U = \begin{bmatrix} 0.15 & 0.85 & 0.1 & 0.5 & 0.35 \\ 0.75 & 0.05 & 0.8 & 0.15 & 0.55 \end{bmatrix}.$$

Density-based clustering

Density clustering method steps

DBScan is one of density-based clustering methods. It consist of the following steps:

1. calculate distance matrix,
2. get closest element,
3. merge into a cluster if the distance is small enough.

It can be also used to find noise in the dataset.

Neighborhood

The density is calculated using the neightborhood elements:

$$N_\epsilon : q|d(p,q) \leq \epsilon, \tag{12}$$

where p and q are two elements of the training data set and ϵ is the neighborhood distance.

Distance matrix

Distance matrix D for aircrafts example looks like following:

$$D = \begin{bmatrix} 0.0 & 0.06 & 0.17 & 0.14 & 0.16 & 0.14 & 1.22 & 1.30 & 1.04 & 0.49 \\ 0.06 & 0.0 & 0.10 & 0.16 & 0.16 & 0.09 & 1.17 & 1.25 & 0.98 & 0.44 \\ 0.14 & 0.16 & 0.23 & 0.0 & 0.04 & 0.14 & 1.14 & 1.21 & 0.99 & 0.39 \\ 0.16 & 0.16 & 0.22 & 0.04 & 0.0 & 0.13 & 1.10 & 1.17 & 0.95 & 0.35 \\ 0.14 & 0.09 & 0.09 & 0.14 & 0.13 & 0.0 & 1.08 & 1.16 & 0.89 & 0.35 \\ 1.22 & 1.17 & 1.10 & 1.14 & 1.10 & 1.08 & 0.0 & 0.08 & 0.35 & 0.74 \\ 1.30 & 1.25 & 1.19 & 1.21 & 1.17 & 1.16 & 0.08 & 0.0 & 0.43 & 0.82 \\ 1.04 & 0.98 & 0.89 & 0.99 & 0.95 & 0.89 & 0.35 & 0.43 & 0.0 & 0.63 \\ 0.49 & 0.44 & 0.41 & 0.39 & 0.35 & 0.35 & 0.74 & 0.82 & 0.63 & 0.0 \end{bmatrix}$$

DBScan – choose the first object

Next, we choose one randomly chosen element and get the lowest distances. Let the chosen object is x_3:

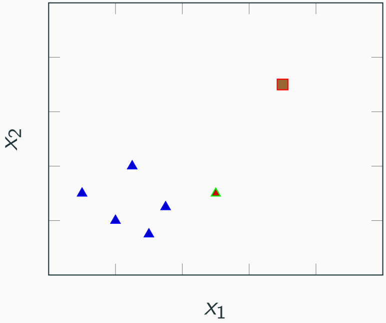
$$d_3 = \begin{bmatrix} 0.14 & 0.16 & 0.23 & 0.0 & 0.04 & 0.14 & 1.14 & 1.21 & 0.99 & 0.39 \end{bmatrix}.$$

We include all object where the distance is below the `max_distance` into a cluster. If the `max_distance` (ϵ) is 0.2 than we have four other objects within this cluster.

We take an object as border object if it's below `min_points`. All other objects are treated as noise – does not belong to any cluster.

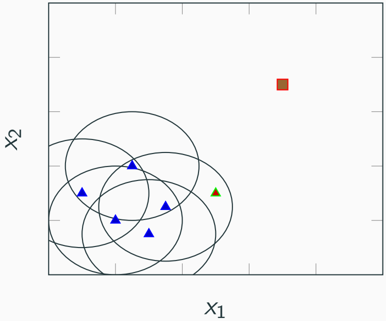
Differences between Density-based and hiearchical

In the figure below we have the core points marked with blue, the border points marked with green and outliers marked with red.



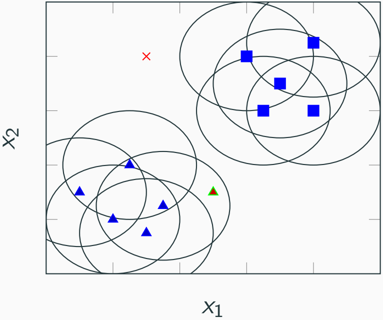
Differences between Density-based and hiearchical

The same figure, but with the neighborhoods:



Differences between Density-based and hiearchical

The same figure, but with the neighborhoods:



Hierachical clustering

Hierchical clustering types

We have two types of hierchical clustering methods:

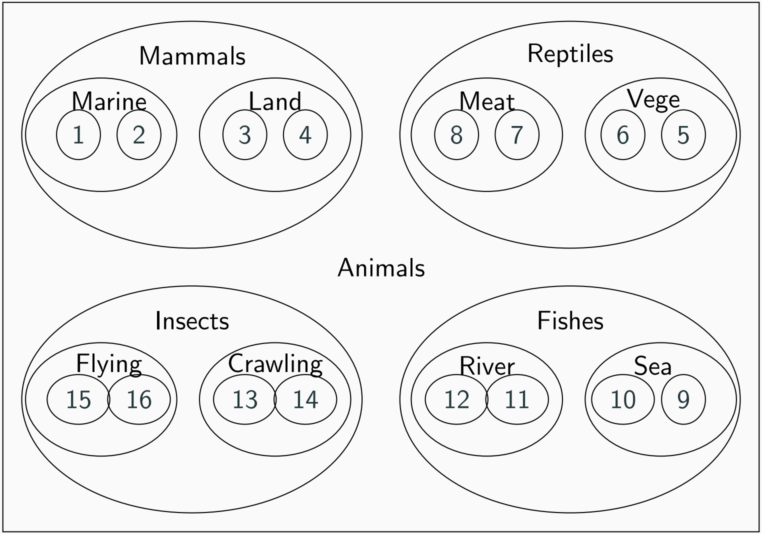
1. agglomerative,
2. divisive.

Both a doing almost the same, but in the opposite direction. Let's take an example to show the differences:

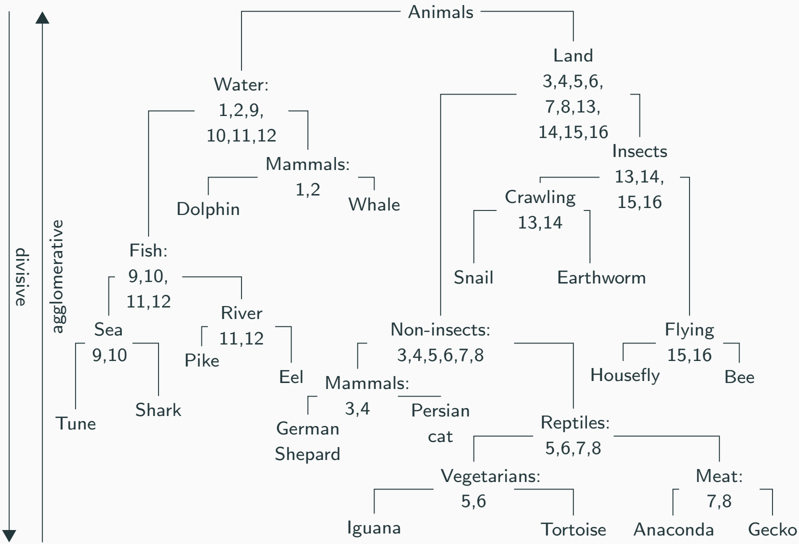
Id	Group	Subgroup	Animal name	Id	Group	Subgroup	Animal name
1	Mammals	Marine	Dolphin	9	Fishes	Sea	Shark
2	Mammals	Marine	Whale	10	Fishes	Sea	Tune
3	Mammals	Land	German Sheppard	11	Fishes	River	Pike
4	Mammals	Land	Persian cat	12	Fishes	River	Eel
5	Reptiles	Vegetarian	Iguana	13	Insects	Crawling	Ladybug
6	Reptiles	Vegetarian	Tortoise	14	Insects	Crawling	Earthworm
7	Reptiles	Meat eaters	Anaconda	15	Insects	Flying	Bee
8	Reptiles	Meat eaters	Gecko	16	Insects	Flying	Housefly

Nested sets

As nested sets the data set would look like following:



Dendrogram



divisive
agglomerative
Animals
Water:
1,2,9,
10,11,12
Land
3,4,5,6,
7,8,13,
14,15,16
Mammals:
1,2
Dolphin
Whale
Insects
13,14,
15,16
Crawling
13,14
Snail
Earthworm
Fish:
9,10,
11,12
Sea
9,10
River
11,12
Pike
Eel
Tune
Shark
Non-insects:
3,4,5,6,7,8
Flying
15,16
Housefly
Bee
Mammals:
3,4
German
Shepard
Persian
cat
Reptiles:
5,6,7,8
Vegetarians:
5,6
Iguana
Tortoise
Meat:
7,8
Anaconda
Gecko

Agglomerative clustering

The agglomerative clustering method is divided into three steps:

1. calculate current dendrogram distance matrix,
2. get lowest distance from matrix,
3. merge clusters/elements into clusters.

It is repeated until we have one cluster or expected clusters number.

Agglomerative clustering – lowest distance

Where is Wally?

$$D = \begin{bmatrix} 0.0 & 0.06 & 0.17 & 0.14 & 0.16 & 0.14 & 1.22 & 1.30 & 1.04 & 0.49 \\ 0.06 & 0.0 & 0.10 & 0.16 & 0.16 & 0.09 & 1.17 & 1.25 & 0.98 & 0.44 \\ 0.14 & 0.16 & 0.23 & 0.0 & 0.04 & 0.14 & 1.14 & 1.21 & 0.99 & 0.39 \\ 0.16 & 0.16 & 0.22 & 0.04 & 0.0 & 0.13 & 1.10 & 1.17 & 0.95 & 0.35 \\ 0.14 & 0.09 & 0.09 & 0.14 & 0.13 & 0.0 & 1.08 & 1.16 & 0.89 & 0.35 \\ 1.22 & 1.17 & 1.10 & 1.14 & 1.10 & 1.08 & 0.0 & 0.08 & 0.35 & 0.74 \\ 1.30 & 1.25 & 1.19 & 1.21 & 1.17 & 1.16 & 0.08 & 0.0 & 0.43 & 0.82 \\ 1.04 & 0.98 & 0.89 & 0.99 & 0.95 & 0.89 & 0.35 & 0.43 & 0.0 & 0.63 \\ 0.49 & 0.44 & 0.41 & 0.39 & 0.35 & 0.35 & 0.74 & 0.82 & 0.63 & 0.0 \end{bmatrix}$$

Agglomerative clustering – lowest distance

Where is Wally?

$$D = \begin{bmatrix} 0.0 & 0.06 & 0.17 & 0.14 & 0.16 & 0.14 & 1.22 & 1.30 & 1.04 & 0.49 \\ 0.06 & 0.0 & 0.10 & 0.16 & 0.16 & 0.09 & 1.17 & 1.25 & 0.98 & 0.44 \\ 0.14 & 0.16 & 0.23 & 0.0 & \mathbf{0.04} & 0.14 & 1.14 & 1.21 & 0.99 & 0.39 \\ 0.16 & 0.16 & 0.22 & 0.04 & 0.0 & 0.13 & 1.10 & 1.17 & 0.95 & 0.35 \\ 0.14 & 0.09 & 0.09 & 0.14 & 0.13 & 0.0 & 1.08 & 1.16 & 0.89 & 0.35 \\ 1.22 & 1.17 & 1.10 & 1.14 & 1.10 & 1.08 & 0.0 & 0.08 & 0.35 & 0.74 \\ 1.30 & 1.25 & 1.19 & 1.21 & 1.17 & 1.16 & 0.08 & 0.0 & 0.43 & 0.82 \\ 1.04 & 0.98 & 0.89 & 0.99 & 0.95 & 0.89 & 0.35 & 0.43 & 0.0 & 0.63 \\ 0.49 & 0.44 & 0.41 & 0.39 & 0.35 & 0.35 & 0.74 & 0.82 & 0.63 & 0.0 \end{bmatrix}$$

Divisive clustering

The divisive clustering method is divided into three steps:

1. calculate distance matrix in each cluster,
2. get highest distance average,
3. split clusters.

It is repeated until we have no cluster to be divided or expected clusters number is reached.

Disimmilarity measure

There are few popular dissimilarity measures that we can use to generate the distance marix.

Measure name	Equation	
Manhattan distance	$\rho_{Man}(x_r, x_s) = \sum_{i=1}^{n} \lvert x_{ri} - x_{si} \rvert$	(13)
Chebyshew distance	$\rho_{Ch}(x_r, x_s) = max_{1 \leq i \leq n} \lvert x_{ri} - x_{si} \rvert$	(14)
Frecht distance	$\rho(x_r, x_s) = \sum_{i=1}^{d} \frac{\lvert x_{ri} - x_{si} \rvert}{1 + \lvert x_{ri} + x_{si}} \frac{1}{2^i}$	(15)
Canberra distance	$\rho(x_r, x_s) = \sum_{i=1}^{d} \frac{\lvert x_{ri} - x_{si} \rvert}{\lvert x_{ri} + x_{si} \rvert}$	(16)
Post office distance	$\rho_{pos}(x_r, x_s) = \begin{cases} \rho_{Min}(x_r, 0) + \rho_{Min}(0, x_s), \text{ for } x_r \neq x_s, \\ 0, \text{ for } x_r = x_s \end{cases}$	(17)
Bray-Curtis distance	$\rho_{bc}(x_r, x_s) = \frac{\sum_{i=1}^{d} \lvert x_{ri} - x_{si} \rvert}{\sum_{i=1}^{d} (x_{ri} - x_{si})}$	(18)

Agglomerative distance measure types

Based on the approach, we can use few methods to measure the distance between clusters:

Method name	Equation	
Single Linkage	$d_{12} = \min_{i,j} d(X_i, Y_j)$	(19)
Complete Linkage	$d_{12} = \max_{i,j} d(X_i, Y_j)$	(20)
Average Linkage	$d_{12} = \frac{1}{kl} \sum_{i=1}^{k} \sum_{j=1}^{j} d(X_i, Y_j)$	(21)
Centroid Method	$d_{12} = d(\bar{x}, \bar{y})$	(22)

Quality metrics

Quality metric types

Finding the best clustering method is not an easy task. To make this task easier we can use multiple validation methods. The most important factors are **homogeneity** and **heterogeneity** of a clustering method.

Other possibility is to use one of a commonly known **validation methods**.

How many clusters should we have? Real world clustering problems can be complex and it can be hard to choose the **best number of clusters**.

Good clustering

We say that the clustering method done well when we have:

- high intra-class similarity,
- low inter-class similarity.

Both values are also known as homogeneity and heterogeneity.

Homogeneity

Two metrics of homogeneity that we explain here are marked as σ_1 and σ_2. Both are related to the differences within each cluster. The differences are known as a dispersion measures within a cluster.

As we do some calculation within a cluster we need to refer to a cluster center. The equation of the average objects dispersion looks like following:

$$\sigma_1(c_i) = \frac{1}{m} \sum_{x_1, x_2 \in c_i} d^2(x_1, x_2), \tag{23}$$

where the m is defined as

$$m = \frac{(n_i - 1)n_i}{2}. \tag{24}$$

The n_i is the count of objects within i cluster. If we have two clusters we calculate two dispersion measures σ_1, one for each cluster.

Homogeneity – σ_1

Let's take the aircrafts example again. We have two clusters, one with seven objects and second with three objects. For c_1, the value of m is easy to calculate:

$$m = \frac{(7-1)*7}{2} = 21.$$

Based on that we can calculate the value of $\sigma_1(c_1)$:

$$\sigma_1(c_1) = \frac{1}{21} * (0.065 + 0.17 + \ldots + 0.35) = 2.77.$$

The same can be calculated for the second cluster:

$$\sigma_1(c_2) = \frac{1}{3} * (0.086 + \ldots + 0.43) = 0.968$$

. The second cluster has lower dispersion.

Homogeneity

A second dispersion measure is marked as σ_2. We calculate here the distance power between each object x within a cluster and the cluster center c_i. We divide the result by the count of objects within the cluster:

$$\sigma_2(c_i) = \frac{1}{n_i} \sum_{x \in c_i} d^2(x, c_i). \tag{25}$$

It looks a bit simpler comparing to σ_1. In both cases the smallest value testifies a better clustering result.

Homogeneity – σ_2

This measure shows the dispersion between object and the center of object's cluster.

$$\sigma_2(c_2) = \frac{1}{3} = 0.092 + \ldots + 0.1117 = 0.1643.$$

The same measure for the first centroid is 0.4392. The first has a lower dispersion.

Homogeneity

Similar measures as are the total dispersion measures. Those metrics give a better understanding of recurrence of objects within a cluster and feature space. Both metrics are just sums of dispersion measures σ_1 and σ_2:

$$r(\sigma_1) = \sum_{i=1}^{K} \sigma_1(c_i), \tag{26}$$

$$r(\sigma_2) = \sum_{i=1}^{K} \sigma_2(c_i). \tag{27}$$

Small values of $r(\sigma_1)$ and $r(\sigma_2)$ means high recurrence of objects within the feature space. Higher values mean exactly the opposite.

Heterogeneity

We have four separation measures $s_1(c_i, c_j)$, $s_2(c_i, c_j)$, $s(s_1)$ and $s(s_2)$. The first two separation measures explain how far from each other the clusters are. We measure it for each pair of centroids.

$$s_1(c_i, c_j) = \frac{1}{n_i n_j} \sqrt{\sum_{x_1, \in c_i, x_2 \in c_j} d^2(x_1, x_2)}. \tag{28}$$

We take two objects, each from different cluster and calculate the power distance measure. Next, we sum all the distances from object of two clusters and calculate a square root of it. The value is then divided by the multiplication of the counts of objects in both clusters.

Heterogeneity – $s_1(c_i, c_j)$

The dispersion measure s_1 for the aircraft example can be calculated as follows:

$$s_1(c_1, c_2) = \frac{1}{7 * 3}\sqrt{1.2247^2 + \ldots + 0.6323} = 0.1006. \tag{29}$$

It's quite low.

Heterogeneity

The second separation measure is about the distance between two centroids:

$$s_2(c_i, c_j) = d(c_i, c_j). \tag{30}$$

It is the simplest measure so far as it is not even a sum. We need two loops to go through the centroids

Homogeneity and heterogeneity

The third separation measure uses dispersion measure σ_1. It is a simple sum of a division of s_1 for two centroids and σ_1 for centroid c_i:

$$s(s_1) = \sum_{i,j=1;j\neq i}^{K} \frac{s_1(c_i, c_j)}{\sigma_1(c_i)}. \tag{31}$$

This measure is taking the whole feature space into consideration. The sums can be easily calculated if we already have s_1 and σ_1.

The last measure is also simple. It is a sum of measures s_2:

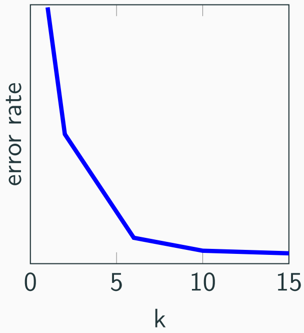
$$s(s_2) = \sum_{i,j=1;j\neq i}^{K} s_2(c_i, c_j) \tag{32}$$

Homogeneity and heterogeneity

We do not need to calculate all measures to get to know if our clustering method is performing well. In most cases we can use just few or one. Especially $r(\sigma_2)$ is used very often.

Clusters number

The number of clusters can be chosen using the elbow method. The goal of this method is to choose many numbers k and calculate the error rate. Based on the result of each execution we can plot a chart that can look like



At some point in this chart increasing the number k gives a a lower error rate. The error rate is next on a similar level for the next values of k.

Internal and external indices

Internal and external indices are other type of measures that shows how good our clustering method is. The difference between internal and external indices depends on the information used to calculate the index.

Internal indices are based only on training data set.

External indices uses the labels and testing data set.

Dunn index

The Dunn index can be easily calculated as a quotient of two distances:

$$C = \frac{d_{min}}{d_{max}}, \tag{33}$$

where the equations of d_{max} and d_{min} are like following:

$$d_{\max} = \max_{1 \leq k \leq K} D_k, \tag{34}$$

$$d_{\min} = \min_{k \neq k'} d_k. \tag{35}$$

Both distances are just the minimum and maximum euclidean distances between objects. The minimum distance is a measure of two object that are in different clusters:

$$d_k = \min_{i,j \in I_k; i \neq j} d(x_i^{(k)} - x_j^{(k')}) \tag{36}$$

.

Dunn index

The clusters are marked with k and k'. The maximum distance takes the distance of two objects within a cluster:

$$D_k = \max_{i,j \in I_k; i \neq j} d(x_i^{(k)} - x_j^{(k)}). \tag{37}$$

D_k and d_k values are calculated for each cluster k, but in Dunn index we take only the highest value of D_k and the lowest value of d_k.

Dunn index

The Dunn index for aircraft examples is as follows:

$$C = \frac{0.6323}{1.3029} = 0.4853.$$

During the classes we calculate the Dunn index for three clusters.

Czekanowski-Dice index

Czekanowski-Dice index is taking labels of the testing data set to measure the quality of a clustering method. It is calculated as following:

$$C = 2\frac{P \times R}{R + R}. \tag{38}$$

Precision R that is also known as Positive Predictive Value (PPV):

$$PPV = \frac{\#TP}{\#TP + \#FP}. \tag{39}$$

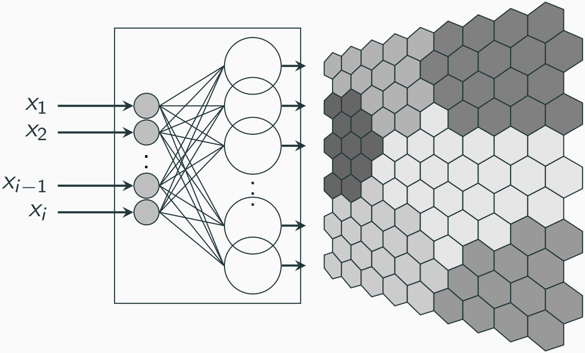
TPR is also called sensitivity or recall and is a measure of good predictions within a set of cases:

$$TPR = \frac{\#TP}{\#TP + \#FN}. \tag{40}$$

Other clustering methods

Kohonen network – SOM

Kohonen neural networks or Self-Organizing Maps are unsupervised neural networks where the goal is create a representation of the feature space.



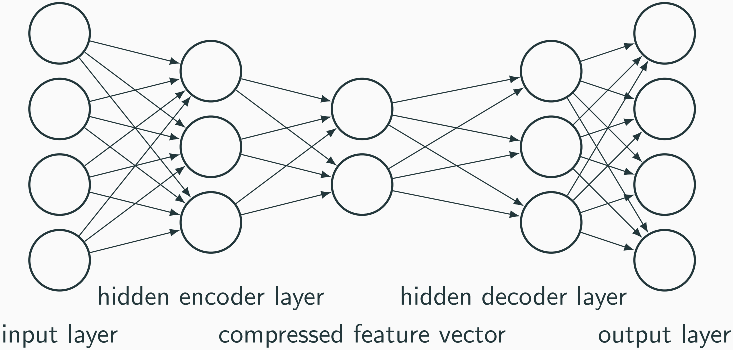
Kohonen network

There are a few strategies how to change the weights of each feature vector:

1. WTA – Winner Takes All,
2. WTM – Winner Takes Most.

The map has a fixed size.

Autoencoders



Questions?