WIRTSCHAFTS
UNIVERSITÄT
WIEN VIENNA
UNIVERSITY OF
ECONOMICS
AND BUSINESS

Data Science and Machine Learning 2187 & 2087: Unsupervised Learning

Max Thomasberger,

December 4, 2020

Goals of this lecture

- Understand the difference between unsupervised- and supervised learning
- Understand the definition of hard clustering
- Understand clustering cost from different similarity measures
- Understand the K-means algorithm
- Understand the K-medeoids algorithm

Some notation

Feature vectors x, labels y

$$x \in \mathbb{R}^d$$
$$y \in \{-1, 1\}$$

Training set

$$S_n = \{(x^{(i)}, y^{(i)}), i = 1, ..., n\}$$

Classifier

$$h : \mathbb{R}^d \to \{-1, 1\}$$

Supervised Learning vs. Unsupervised Learning

- In supervised learning we have labeled data: $S_n = \{(x^{(i)}, y^{(i)}), i = 1, ..., n\}$ and want to learn to correctly classify unseen data
  - Think of:
  - A gazillion of photos with a "cat" and "not cat" classification.
  - etc.
- In clustering we only have feature vectors: $S_n = \{x^{(i)} | i = 1, \cdots, n\}$ and want to find structures in unlabeled data
  - Think of:
  - Clustering a data set of customer into groups
  - Find spatial patterns, e.g. crime hotspots
  - Find similar news stories
  - Recommend products to customers "like you"
  - Create labels for supervised learning algorithms
  - Exploratory data analysis
- Types
  - Hard clustering
  - Soft clustering
  - Hierarchical clustering

Clustering as Partitioning

Clustering input: $S_n = \{x^{(i)} | i = 1, \cdots, n\}, K$

Number of clusters: K

The output of the clustering algorithm are indexes that partition the data: $C_1, \cdots, C_k$; where $C_1 \cup C_2 \cup ... \cup C_K = \{1, 2, ..., n\}$ and $C_i \cap C_j = \emptyset$ {for any $i \neq j$ in $\{1, ..., k\}$.

In other words: the union of all C_j 's is the original set and the intersection of any C_i and C_j is an empty set.

In plain English:

We want to assign each element of the training data set S_n into K separate clusters in a way that each element only belongs to one cluster.

Clustering as selecting representatives

Clustering input: $S_n = \{x^{(i)} | n = 1, \cdots, n\}, K$

Number of clusters: K

Select the best representatives of each cluster: $z^{(1)}, \cdots, z^{(k)}$.

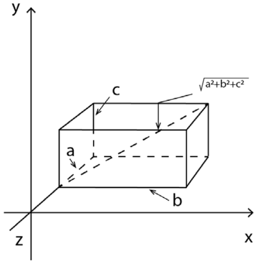
Similarity Measures-Cost functions

- Cost of partitioning
  - Sum of costs of all individual clusters: $cost(C_1, \cdots, C_k) \sum_{j=1}^{k} cost(C_j)$.
- Cost of a single cluster
  - Sum of distances from data points to the representative of the cluster: $Cost(C, z) = \sum_{i \in C} distance(x^{(i)}, z)$
- Total Cost to be minimized
  - $\text{Cost}(C_1, ..., C_K) = \sum_{j=1}^{K} \text{Cost}(C_j) = \sum_{j=1}^{K} \sum_{i \in C_j} \|x_i - z_j\|^2$

Two common distance measures

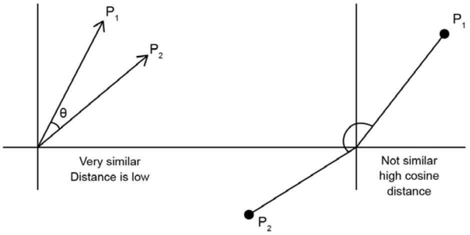
- Cosine similarity: $cos(x^{(i)}, x^{(j)}) = \frac{x^{(i)}.x^{(j)}}{||x^{(i)}||||x^{(j)}||}$
  - Is not sensitive of magnitude of vector (will not react to length).
- Euclidean squared distance: $dist(x^{(i)}, x^{(j)}) = ||x^{(i)} - x^{(j)}||^2$.
  - Will react to length of the vectors

Euclidean Distance and Euclidean Squared Distance



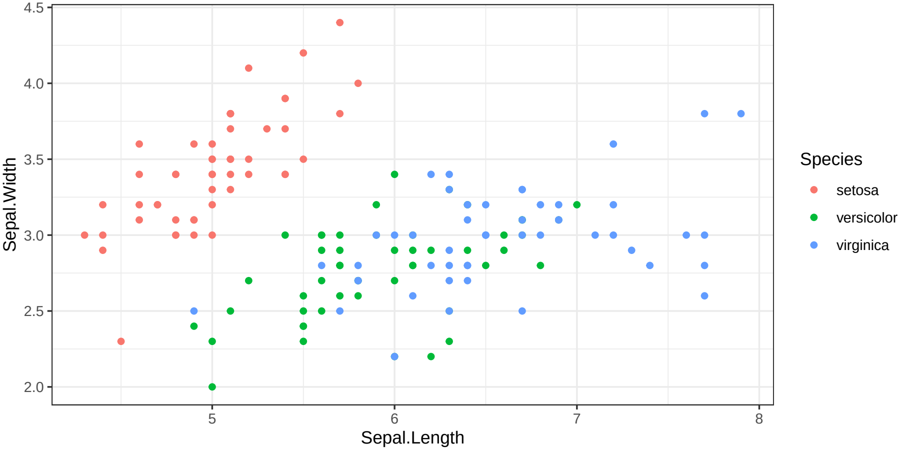
- The Euclidean distance between any two points is the square root of the sum of squares of differences between the coordinates. Straight line distance between any two points (pythagorean theorem)
- In two dimensions: $dist(p, q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2}$.
- Squared Euclidean distance is the sum of squares: $dist^2(p, q) = (p_1 - q_1)^2 + (p_2 - q_2)^2$
- Generalizes to n-dimensions but looses meaning in very high dimensional data

Cosine distance and Cosine similarity

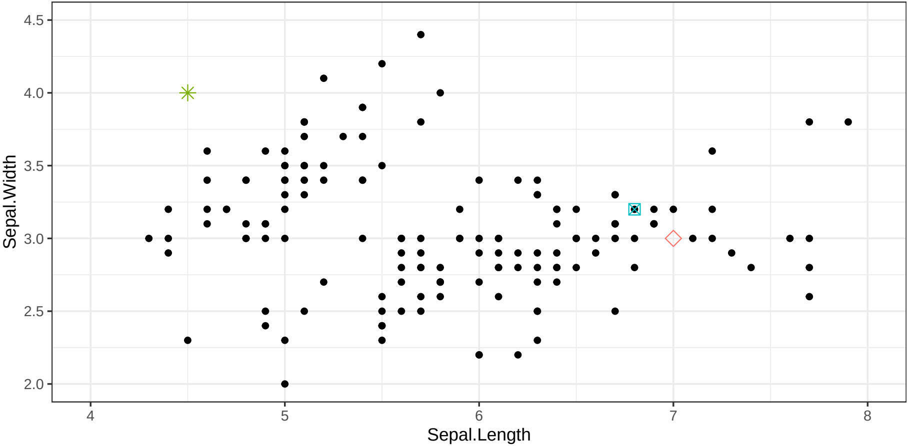


- Cosine similarity between any two points is defined as the cosine of the angle between any two points with the origin as its vertex.
- Cosine distance is defined as: 1 - cosine similarity
- Cosine distance varies from 0 to 2, whereas cosine similarity varies between -1 to 1.

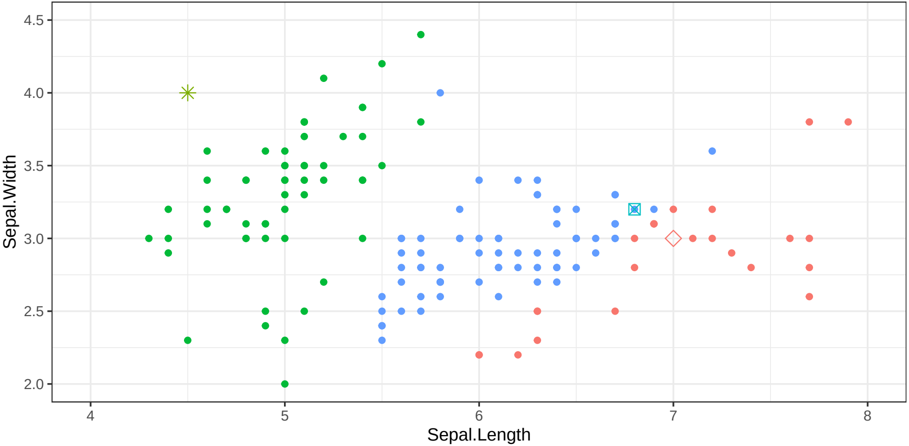
Iris Dataset



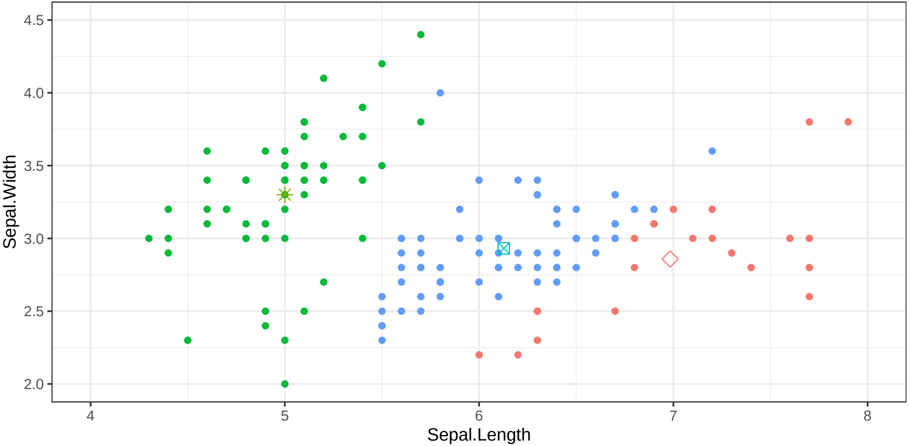
Assign random initialization points (representatives z_j)



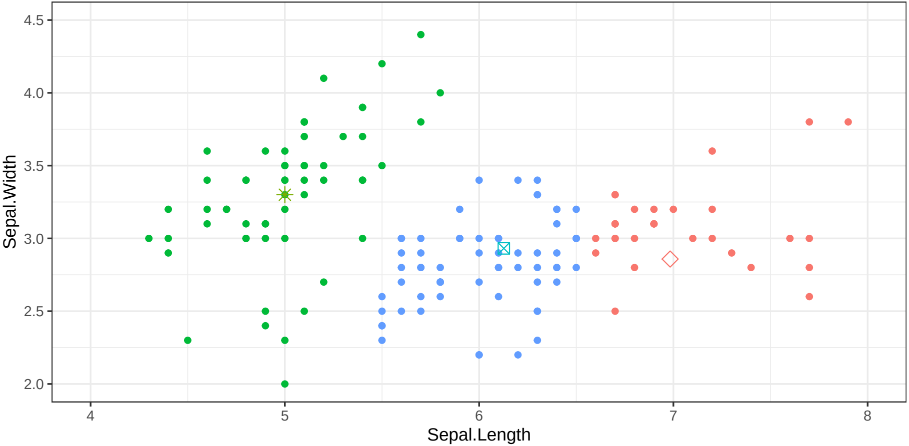
Calculate nearest points to representatives z_j (1st round)



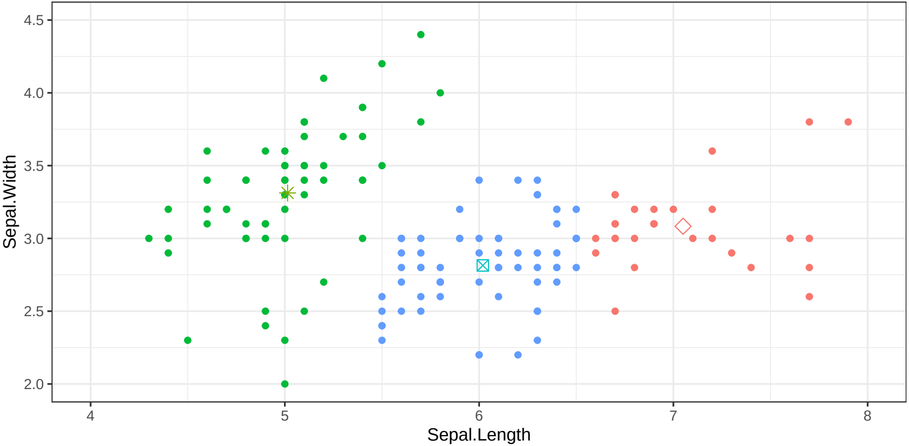
Recalculate representatives z_j as centroids of the new cluster (1st round)



Calculate nearest points to representatives z_j (2nd round)



Recalculate representatives z_j as centroids of the new cluster (2nd round)



K-means Algorithm

Given a set of feature vectors $S_n = \{x^{(i)} | i = 1, ..., n\}$ and the number of clusters K we can find cluster assignments $C_1, \cdots, C_K$ and the representatives of each of the K clusters $z_1, \cdots, z_K$:

1. Randomly select $z_1, \cdots, z_K$
2. Iterate until no change in cost
   2.1 Given $z_1, \cdots, z_K$, assign each data point $x^{(i)}$ to the closest z_j, such that $\text{Cost}(z_1, ...z_K) = \sum_{i=1}^{n} \min_{j=1,...,K} \left\| x^{(i)} - z_j \right\|^2$
   2.2 Given $C_1, \cdots, C_K$ find the best representatives $z_1, \cdots, z_K$, i.e. find $z_1, \cdots, z_K$ such that $z_j = \text{argmin}_z \sum_{i \in C_j} \|x^{(i)} - z\|^2$

Minimizing the cost in K-means

K-means only works with Euclidean square distance!

The best representative is found by optimization (gradient with respect to $z^{(j)}$, setting to zero and solving for $z^{(j)}$).

$\nabla_{z_j} \left(\sum_{i \in \mathbb{C}_j} \|x^{(i)} - z_j\|^2 \right) = 0$

$\sum_{i \in \mathbb{C}_j} -2(x^{(i)} - z_j) = 0$

$z^{(j)} = \frac{\sum_{i \in C_j} x^{(i)}}{|C_j|}$

It is the centroid of the cluster, where C_j is the size of the respective cluster.

The clustering output that the K-Means algorithm converges to depends on the intialization! Suboptimal initializations are possible.

K-means in pseudo code for two clusters

1. Choose any two random coordinates, z_1 and z_2, on the scatter plot as initial cluster centers.
2. Calculate the distance of each data point in the scatter plot from coordinates z_1 and z_2
3. Assign each data point to a cluster based on whether it is closer to z_1 or z_2
4. Find the mean coordinates of all points in each cluster and update the values of z_1 and z_2 to those coordinates respectively.
5. Start again from Step 2 until the coordinates of z_1 and z_2 stop moving significantly, or after a certain pre-determined number of iterations of the process.

Limitations of the K-Means Algorithm

- Algorithm is only guaranteed to converge to local minimum
- Initialization matters
  - Bad initalization can lead to suboptimal clusters in pathological cases
- Unlclear how many cluster we should plug into the algo (more about that next lecture)
- Only works with eucledian distance

K-Medoids

Any distance measure possible!

Gives actual data points as representatives.

Finds the cost-minimizing representatives $z_1, \cdots, z_K$ for any distance measure. Uses real data points for initialization.

1. Randomly select $\{z_1, ..., z_K\} \subseteq \{x_1, ..., x_n\}$
2. Iterate until no change in cost
   2.1 Given $z_1, \cdots, z_K$, assign each data point $x^{(i)}$ to the closest z_j, so that $\text{Cost}(z_1, ...z_K) = \sum_{i=1}^{n} \min_{j=1,...,K} \left\| x^{(i)} - z_j \right\|^2$
   2.2 Given $C_j \in \{C_1, ..., C_K\}$ find the best representative $z_j \in \{x_1, ..., x_n\}$ such that $\sum_{x^{(i)} \in C_j} \text{dist}(x^{(i)}, z_j)$ is minimal

K-Mediods pseudocode

1. Choose k data points from the scatter plot as starting points for cluster centers.
2. Calculate their distance from all the points in the scatter plot.
3. Classify each point into the cluster whose center it is closest to.
4. Select a new point in each cluster that minimizes the sum of distances of all points in that cluster from itself.
5. Repeat Step 2 until the centers stop changing.

Literature