Neural Networks 1 - Self-organization

18NES1 - Lecture 11, Summer semester 2024/25

Zuzana Petříčková

April 29th, 2025

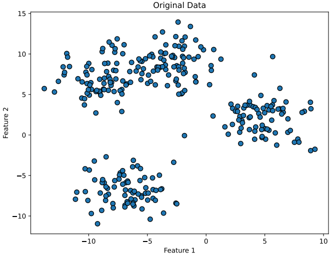
This Week

Unsupervised Learning (Self-Organization)

- General Overview
- Classification and the k-Nearest Neighbors Algorithm
- Clustering and the k-Means Algorithm
- Demonstrations and examples

Unsupervised Learning and Self-Organization

- Training set T in the form $T = \{\vec{x}_1, \dots, \vec{x}_N\}$ (only inputs)
- $\vec{x}_i \in \mathbb{R}^n$ is the i-th training input pattern, target outputs are unknown
- **Idea:** the model itself decides which response is best for a given input and adjusts its weights accordingly $\rightarrow$ self-organization

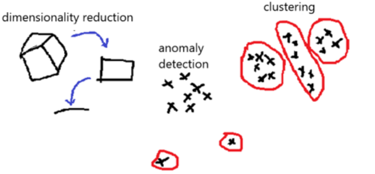


- We have data but no knowledge of its internal structure
- The goal is to uncover the structure and patterns within the data

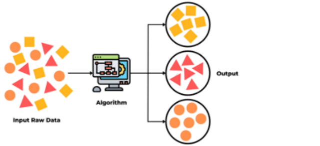
Unsupervised Learning and Self-Organization

- **Goal:** to discover structure or patterns within the data
- **Applications:**
  - **Dimensionality reduction** (data compression, visualization)
  - **Anomaly detection** (e.g., in banking transactions)
  - **Clustering** (e.g., customer segmentation, plagiarism detection)
  - E-commerce: recommendation systems

Types of Tasks:



https://towardsdatascience.com/unsupervised-learning-algorithms-cheat-sheet-d391a39de44a



https://eastgate-software.com/what-is-unsupervised-learning/#ftoc-heading-7

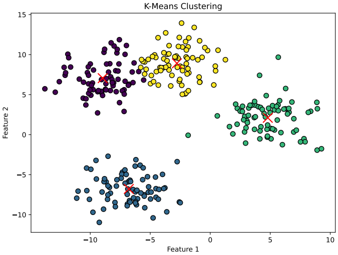
Unsupervised Learning and Self-Organization

Cluster

- A group of samples with **high similarity among themselves** and **low similarity to samples in other clusters**
- In simplified terms: **similarity = proximity**

Clustering

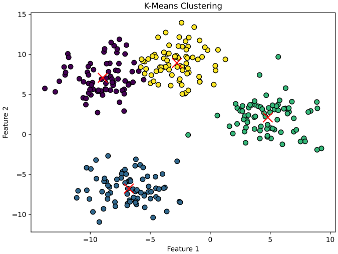
- Disjoint partitioning of data into clusters



Clustering

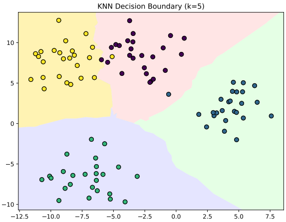
Challenges:

- How to determine the number and distribution of clusters in the feature space?
- How to choose the representative(s) of a cluster?
  - Appropriately selected training samples belonging to a cluster
  - Example: the centroid of a cluster



Detour: k-Nearest Neighbors Algorithm

- A classification method using supervised learning: Training patterns are stored and classified into one of l different classes
- An unknown input vector is assigned to the class most common among the k nearest vectors from the stored set



- Simplest variant: 1-nearest neighbor, the classification model consists of the stored data

Detour: k-Nearest Neighbors Algorithm

How to compute the distance (similarity) between numerical vectors?

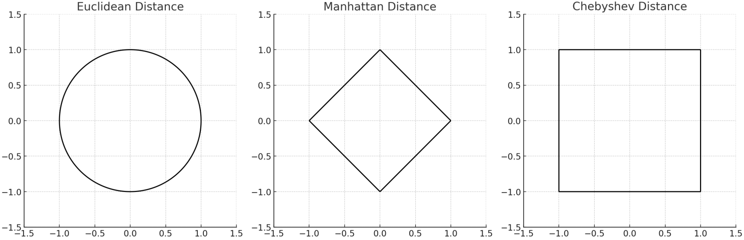
- **Euclidean distance:** $d(\vec{p}, \vec{q}) = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2}$
- When only comparing distances (for efficiency), it is common to use the squared distance: $d(\vec{p}, \vec{q}) = \sum_{i=1}^{n}(p_i - q_i)^2$

Other distance metrics include:

- **Manhattan (city block) distance:** $d(\vec{p}, \vec{q}) = \sum_{i=1}^{n} |p_i - q_i|$
- **Chebyshev distance:** $d(\vec{p}, \vec{q}) = \max_i |p_i - q_i|$ *"What is the biggest problem?"*
- **Minkowski distance:** $d(\vec{p}, \vec{q}) = \left(\sum_{i=1}^{n} |p_i - q_i|^r\right)^{\frac{1}{r}}$ *Generalizes the previous metrics ($r = 2$, $r = 1$, $r \to \infty$)*
- **Cosine similarity:** $\cos(\vec{p}, \vec{q}) = \frac{\vec{p} \cdot \vec{q}}{\|\vec{p}\| \|\vec{q}\|}$ *We focus on the direction, not the magnitude (useful for text processing)*
- (and others)

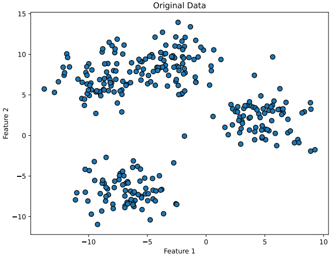
Detour: k-Nearest Neighbors Algorithm

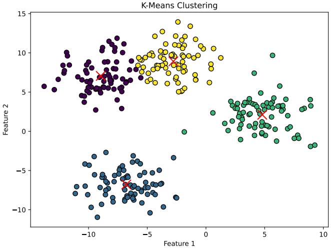
How to compute the distance (similarity)?



The k-Means Clustering Algorithm

- Unsupervised learning
- Input patterns are classified into k different clusters, each cluster l is represented by its centroid $\vec{c}_l$
- A new vector $\vec{x}$ is assigned to the cluster i whose centroid $\vec{c}_l$ is closest to it





The k-Means Clustering Algorithm

1. Given a training set $T = \{\vec{x}_1, \ldots, \vec{x}_N\}, \vec{x}_i \in \mathbb{R}^n$
2. Select k random vectors $\vec{c}_l, l = 1, \ldots, k$ (from $\mathbb{R}^n$ or from T) as initial cluster centroids
3. Repeat:
   - Assign each vector from T to the nearest cluster centroid
   - Recalculate the cluster centroids based on assigned patterns:

$$\vec{c}_l = \frac{1}{n_l} \sum_{l_i=1}^{n_l} (\vec{x}_{l_i})$$

     where n_l is the number of vectors assigned to cluster l, and l_i indexes vectors assigned to cluster l
   - Repeat the above steps until the cluster memberships of training patterns no longer change

Initialization in k-Means

- The result of k-means depends heavily on the initial choice of centroids.
- Poor initialization $\rightarrow$ poor clustering, slow convergence, getting stuck in a local minimum.
- Initialization options:
  - Random vectors in $\mathbb{R}^n$ or within the range of the data
  - Random selection of points from the training set T
  - **k-means++**:
    - First centroid is chosen randomly
    - Subsequent centroids are chosen with probability proportional to the square of the distance to the nearest already chosen centroid
  - Running the algorithm multiple times and selecting the best solution (lowest sum of squared distances)

Parameters of the k-Means Algorithm

- Number of clusters k (usually specified by the user)
- Distance metric (default: Euclidean)
- Initialization method (random, k-means++, custom choice)
- Stopping criteria:
  - until cluster memberships stop changing
  - until centroids stop changing
  - reaching a maximum number of iterations
- Further improvements: If a centroid has no points assigned, reinitialize it

Example (Demonstration)

kmeans_clustering.ipynb

- Demonstration of a custom implementation of the k-means algorithm with visualization of the learning process
- Several datasets and different initialization options

Questions:

- How does centroid initialization affect learning?
- How long does learning take for larger datasets?
- How to find the optimal number of clusters?
- How to evaluate the quality of the clusters formed?

The k-Means Clustering Algorithm

Advantages

- Fast algorithm, easy to implement
- Suitable for quick insight into data structure

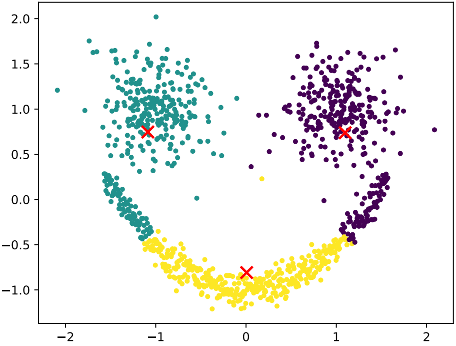
Disadvantages

- The number of clusters must be specified in advance
- Batch processing (problematic for large data or online learning)
- High sensitivity to the initial choice of centroids
- Sensitive to outliers
- May fail for complex data structures: seeks spherical clusters
- Problematic for high-dimensional data (*curse of dimensionality*), or strongly correlated features

The k-Means Clustering Algorithm

Examples of More Complex Tasks:

kmeans_clustering.ipynb



The k-Means Clustering Algorithm

Disadvantages and Their Solutions

- The number of clusters must be specified in advance
  $\rightarrow$ try different values of k and choose the best one
- Batch processing (problematic for large datasets or online learning)
  $\rightarrow$ minibatch or online k-Means
- High sensitivity to the initial choice of centroids
  $\rightarrow$ enhanced initialization
- Sensitivity to outliers
  $\rightarrow$ data normalization:
  - also ensures invariance to scaling and translation
  - but may not always help

The k-Means Clustering Algorithm

Disadvantages and Their Solutions

- May fail for complex data structures: tends to find spherical clusters
  $\rightarrow$ use a different distance metric
- Problems with high-dimensional input data (*curse of dimensionality*), or strongly correlated features
  $\rightarrow$ apply PCA (Principal Component Analysis) for input preprocessing