K-means clustering

Applied Machine Learning in Engineering - Exercise 03

TU Berlin, Summer Term 2025

Prof. Dr.-Ing. Merten Stender – `merten.stender@tu-berlin.de`

In this exercise, you will implement the basic K-means algorithm from scratch. You can choose between two levels of implementation complexity:

- **Option A: procedural (lower complexity)**: Complete a partially implemented procedural version of K-means provided in the file `my_kmeans.py` (available in ISIS).
- **Option B: object-oriented (higher complexity, recommended)**: Implement K-means clustering from scratch using object-oriented programming. You can use the procedural code as inspiration for your class structure.

Learning Objectives

- Understand the core components of the K-means clustering algorithm.
- Apply different distance metrics (L_1, L_2) for clustering.
- Visualize clustering results and interpret centroid positions and assignments.
- Handle edge cases such as empty clusters and monitor clustering metrics over time.
- Practice procedural and object-oriented programming.
- Validate your implementation against a standard library (scikit-learn).

Data Set, Variables, and Helper Functions

The basic structure of the code is given (`my_kmeans.py`), and students need to fill in gaps when Option A was chosen. Please use the following variables in the implementation:

- `x`: a NumPy array of shape `(N, n)` containing the data where `N` is the number of data points and `n` is the dimensionality of the feature space.
- `K`: denotes the number of clusters.
- `labels`: a NumPy array of shape `(N,)` containing cluster assignments (indices starting from 0)
- `centroids`: a NumPy array of shape `(K, n)` storing the centroid positions

For a visual test, you can use the function `plot_clusters` from `utils_clustering.py`. Plot data points and clusters (no labels) using `plot_clusters(x=x, centroids=centroids)` and plot data points with cluster assignment using `plot_clusters(x=x, labels=labels, centroids=centroids)`. You can import functions from a different file using `from <otherFile> import <function>`.

Task 1: Cluster Assignment Function

(a) Implement the function `assign_cluster()` using both L_1 and L_2 norms.

(b) Test your function using a small set of data points and two centroids. Inspect the output labels and visualize the result.

Task 2: Centroid Update Function

(a) Implement the function `update_centroids()` using both L_1 and L_2 norms. Follow the code comments for guidance.

(b) Test your implementation on a small, hand-crafted set of data points. Plot the result and verify that the centroids match your expectations.

Task 3: Convergence Check

Study the function `is_converged()` and annotate each line with a comment describing what it does. Discuss your understanding with a peer.

Task 4: Main K-means Loop

You now have the building blocks to implement the full clustering routine:
`kmeans_clustering(x: np.ndarray, K: int, norm: str = 'L2', init_centroids: np.ndarray = None)`

(a) Write the condition for continuing the `while` loop.

(b) Use the functions from Problems 1 to 3 to perform:

- Cluster assignment
- Centroid updates
- Convergence checking

(c) Test your implementation using the sample data provided at the bottom of `my_kmeans.py`.

Task 5: Validation Against scikit-learn

Compare your implementation with `sklearn.cluster.KMeans` using the dataset `example_data_Kmeans.csv`.

- Do your cluster labels and centroids match those from scikit-learn?
- If there are differences, can you explain why?

Task 6 (Optional): Handling Empty Clusters

Enhance the `update_centroids()` function to detect and handle empty clusters. If a cluster has no assigned points, call `relocate_empty_centroid()`.

Task 7 (Optional): Track Clustering Quality

Modify `kmeans_clustering()` to compute and return the clustering metrics SSE (Sum of Squared Errors) and BSS (Between-Cluster Sum of Squares) at each iteration using: `sse()` and `bss()` from `utils_clustering.py`. Plot how these metrics evolve over the iterations of your K-means algorithm for the sample dataset.