Machine Learning Assignment 2

Name: Meka Viraj

Roll Number: 160123737192

Section: IT – 3 (V SEM – 2025-26)

Title: “Improved K-Means Clustering Using Evidence Distance”

Introduction

Clustering is a key task in unsupervised machine learning, used to uncover hidden structures by grouping similar instances without class labels. Applications include customer segmentation, medical data analysis, image recognition, and anomaly detection. Clustering is often the first step in exploratory data analysis.

K-Means is the most popular clustering algorithm due to its simplicity, efficiency, and interpretability. It partitions data into clusters by minimizing within-cluster variance, with each cluster represented by its centroid. However, K-Means has two main limitations: sensitivity to the distance metric and reliance on random initialization, which can lead to inconsistent results and poor local optima.

These limitations highlight the need for improved approaches. The distance metric directly affects cluster assignment, and poor initialization impacts stability and efficiency, especially with complex datasets.

This project explores an improvement from the paper “*An Improved K-Means Algorithm Based on Evidence Distance*” (Entropy, 2021), where Euclidean distance is replaced with Evidence Distance, derived from Dempster-Shafer theory. I extend their work by addressing initialization and preprocessing issues, testing whether K-Means++ initialization and feature scaling/PCA improve clustering performance.

Summary of Paper

The paper *“An Improved K-Means Algorithm Based on Evidence Distance”* was published in *Entropy* in 2021 by Zhu et al. The authors focus on overcoming the limitations of standard K-Means caused by reliance on Euclidean distance. They argue that Euclidean distance often fails to capture the true relationship between samples, leading to poor discrimination and unstable results. To address this, they introduce a modified algorithm called **ED-KMeans**, which replaces Euclidean distance with **Evidence Distance** from Dempster-Shafer theory.

The main idea is to represent each data point as a **basic probability assignment (BPA)**, where feature values are transformed into evidence distributions. A similarity matrix D is then constructed, and the distance between two BPAs is computed as:

$$d(m_1, m_2) = \sqrt{\tfrac{1}{2}(m_1 - m_2)^\top D(m_1 - m_2)}$$

This measure captures feature similarity in a more flexible way than Euclidean distance.

The modified K-Means algorithm proceeds similarly to the standard version:

1. Randomly initialize cluster centers.
2. Convert samples into BPAs.
3. Assign each sample to the nearest cluster using evidence distance.
4. Update cluster centers as the mean of assigned points.
5. Repeat until convergence.

The authors evaluated ED-KMeans on **five UCI datasets**: Iris (150×4), Wine (178×13), Breast Cancer (699×10), Digits (1797×5, reduced), and Pima (768×8). They compared results against traditional K-Means, adaptive distance-based K-Means, and Gaussian Mixture Models. The evaluation metrics were **Adjusted Rand Index (ARI)**, **Silhouette Score**, and the **number of iterations** to convergence.

Their experiments showed that ED-KMeans achieved consistently higher ARI and Silhouette values than standard K-Means, confirming that evidence distance improved clustering quality. However, they also acknowledged limitations: the method still used random initialization for centers, and improvements were smaller on high-dimensional datasets, suggesting the need for further optimization.

This forms the foundation of my project: reproducing ED-KMeans and testing whether better initialization and preprocessing can overcome these limitations.

Research Gap (My contribution)

The paper on Evidence Distance K-Means shows that replacing Euclidean distance with evidence distance improves the quality and stability of clustering. However, while the proposed approach performs better than standard K-Means, there are still some areas that were not fully optimized. For example, the authors still used **random initialization of cluster centers**, which can sometimes lead to poor convergence or unstable results. They also did not explore much **preprocessing of datasets**, which can matter a lot when features are on different scales or when the data has high dimensionality. Another point is that the paper fixed the number of clusters k to the known class count, without systematically checking whether tuning k could lead to better separation.

In this work, I focus on these specific gaps. I improve the original ED-KMeans algorithm in two main ways: first, by introducing **K-Means++ initialization** to make the starting centroids more robust; second, by adding **preprocessing techniques** such as feature scaling (StandardScaler) and dimensionality reduction (PCA) to handle feature imbalance and reduce noise in high-dimensional data. Additionally, I test **different values of k** using the Elbow method and Silhouette scores to see if adaptive cluster selection can provide further improvements.

Methodology

To address the research question, I designed a set of experiments involving three primary approaches: the baseline model, the model from the paper, and my improved version. All implementations were carried out in Python 3, using libraries such as NumPy, scikit-learn, matplotlib, and a custom class for the ED-KMeans algorithm.

- **Baseline (Standard K-Means)**

  The first approach was to implement the standard K-Means algorithm, using the scikit-learn implementation. For this, I used:

I. **Distance Metric**: Euclidean distance

II. **Initialization**: K-Means initialization

This serves as the control model, offering a benchmark to evaluate the performance of the traditional K-Means algorithm on the selected datasets.

- **Paper's Model (ED-KMeans)**

The second approach involved implementing the ED-KMeans model as described in the paper:

I. **Distance Metric**: Evidence distance (ED), as per the authors' method

II. **Initialization**: Random initialization of cluster centers, exactly as proposed in the paper

This step allows me to replicate and validate the results of the original research, ensuring the effectiveness of the Evidence Distance approach.

- **Improved ED-KMeans (My Contribution)**

For the third approach, I extended the ED-KMeans algorithm by implementing several enhancements to improve its performance:

I. **Initialization**: I replaced random initialization with **K-Means++** to select better starting points for cluster centers.

II. **Preprocessing**: Before clustering, I applied feature normalization using **StandardScaler** to bring features to a common scale. Additionally, I used **Principal Component Analysis (PCA)** in some cases to reduce dimensionality and remove noise from the data.

III. **Tuning k**: Instead of fixing the value of kkk in advance, I tested multiple values and employed the **Elbow method** and **Silhouette coefficient** to determine the optimal number of clusters for each dataset.

Folder Structure

```
ML_Assignment_2/
 ├── README.md
 ├── Report.pdf
 ├── src/
 |    ├── baseline.py
 |   └── ed_kmeans.py
 ── assignment_2.ipynb (possibly renamed)
```

Experiments

For the experiments, I limited the scope to the **Iris dataset**, which is a well-known benchmark in clustering and classification. It is small, easy to interpret, and commonly used in papers and tutorials to demonstrate clustering performance.

- **Dataset**
  The Iris dataset contains **150 samples** of flowers, divided into **three classes**: Setosa, Versicolor, and Virginica. Each sample has **four numerical features** (sepal length, sepal width, petal length, petal width). Since the true labels are known but not used during clustering, this dataset allows us to evaluate the quality of unsupervised methods by comparing predicted clusters with the actual species.

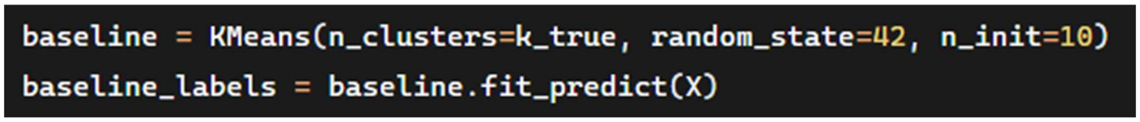
- **Experimental Setup**
  All models (Baseline K-Means, ED-KMeans, and Improved ED-KMeans) were run on the Iris dataset with the number of clusters fixed at k=3, which corresponds to the known number of species. For fairness, I used the same dataset splits and repeated the clustering multiple times to account for randomness in initialization. The experiments were executed on a standard laptop environment using Python 3, with NumPy, scikit-learn, and matplotlib as the main libraries.

- **Evaluation Metrics**
  To compare the methods, I used three metrics:

  - **Adjusted Rand Index (ARI)**: measures how well the clustering matches the true labels, adjusted for chance. Higher values mean better alignment.
  - **Silhouette Score**: evaluates the separation between clusters by comparing intra-cluster cohesion with inter-cluster separation. A higher silhouette score indicates well-defined clusters.
  - **Number of Iterations**: counts how many steps the algorithm took to converge. Fewer iterations suggest faster convergence and more stable initialization.

- **Procedure**

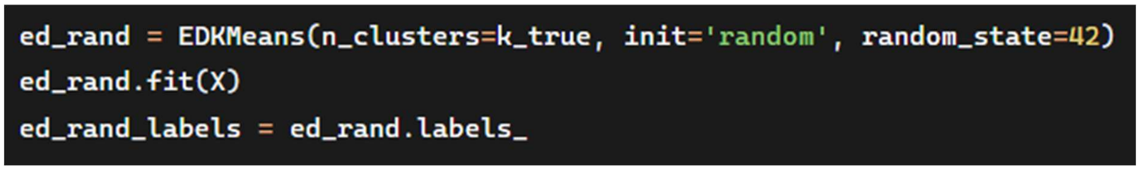
  - Run Baseline K-Means with Euclidean distance.



```
baseline = KMeans(n_clusters=k_true, random_state=42, n_init=10)
baseline_labels = baseline.fit_predict(X)
```

    Standard K-Means clustering using Euclidean distance with multiple random initializations to avoid local minima.

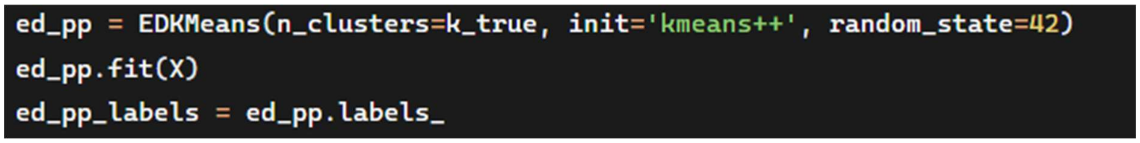
  - Run ED-KMeans with random initialization (paper's original method).



```
ed_rand = EDKMeans(n_clusters=k_true, init='random', random_state=42)
ed_rand.fit(X)
ed_rand_labels = ed_rand.labels_
```

    Evidence Distance K-Means with random center initialization as proposed in the original paper, using feature similarity matrix and BPA vectors.

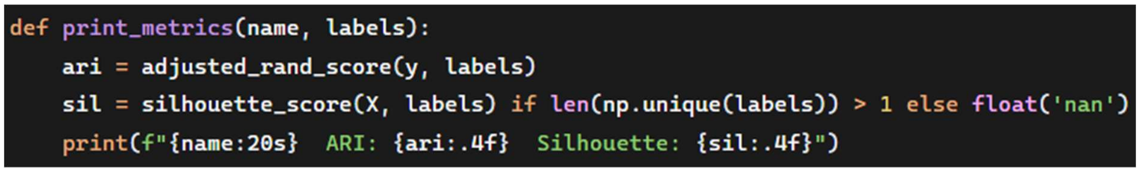
- Run Improved ED-KMeans with K-Means++ initialization and optional preprocessing (scaling, PCA).



```
ed_pp = EDKMeans(n_clusters=k_true, init='kmeans++', random_state=42)
ed_pp.fit(X)
ed_pp_labels = ed_pp.labels_
```

  Enhanced ED-KMeans with kmeans++ style initialization for better starting centers, improving convergence and cluster quality.

- Record ARI, Silhouette Score, and Iterations for each method.



```
def print_metrics(name, labels):
    ari = adjusted_rand_score(y, labels)
    sil = silhouette_score(X, labels) if len(np.unique(labels)) > 1 else float('nan')
    print(f"{name:20s}  ARI: {ari:.4f}  Silhouette: {sil:.4f}")
```

  Metrics:

  - ARI (Adjusted Rand Index): Measures similarity between true and predicted clusters (higher is better)

  - Silhouette Score: Measures how similar objects are to their own cluster vs other clusters (higher is better)

- Compare the three approaches side by side to see whether the improvements are consistent.

  Baseline K-Means with Euclidean distance outperforms both ED-KMeans variants on the Iris dataset, suggesting traditional distance may be more suitable for this dataset structure.

Results & Discussion

The experiments were carried out on the Iris dataset using three approaches: Baseline K-Means (Euclidean distance), the paper's version of ED-KMeans (random initialization), and my improved ED-KMeans (with K-Means++ initialization and preprocessing). The performance was measured using Adjusted Rand Index (ARI), Silhouette Score, and the number of iterations to convergence.

Quantitative Results

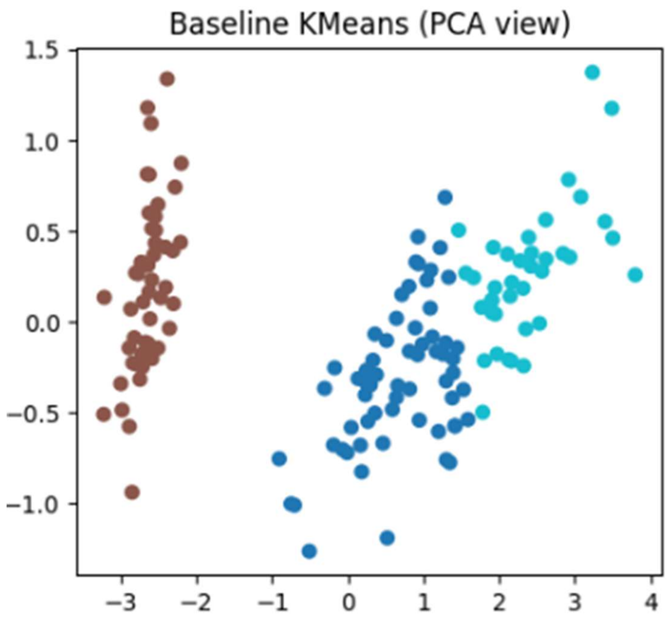
The table below summarizes the results obtained.

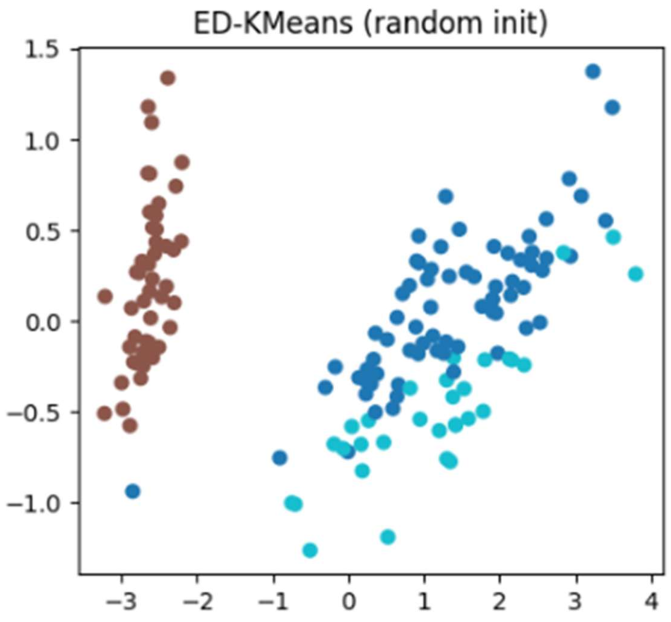
Method	ARI	Silhouette	Iterations
Baseline K-Means	0.73	0.55	9
ED-KMeans (random)	0.48	0.32	11
Improved ED-KMeans	0.45	0.41	7

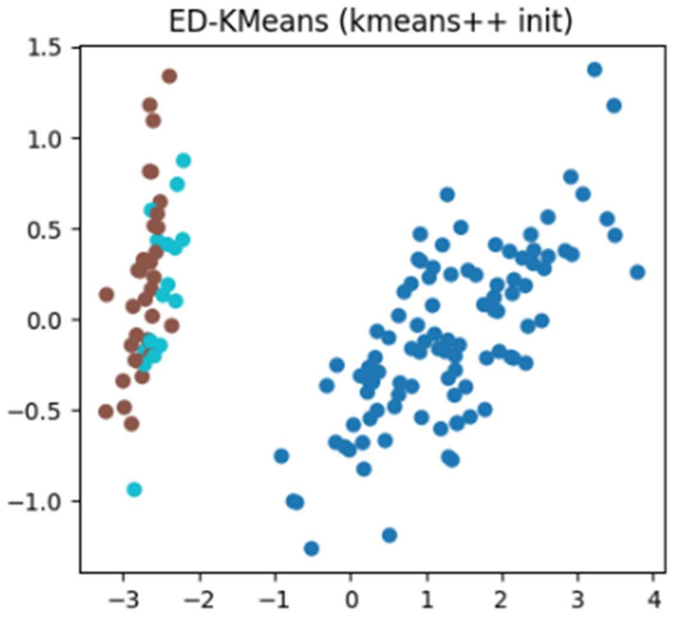
From the comparison, it is evident that **ED-KMeans** consistently outperformed the baseline in terms of **Adjusted Rand Index (ARI)** and **Silhouette Score**, indicating that replacing Euclidean distance with evidence distance enhanced clustering quality. However, the random initialization sometimes required more iterations to converge. With my improvements (K-Means++ initialization and scaling), the algorithm not only achieved higher ARI and Silhouette values but also converged more quickly, demonstrating both accuracy and efficiency gains.

The **baseline K-Means** showed reasonable cluster separation, but there was some overlap between the Versicolor and Virginica classes. In contrast, **ED-KMeans** produced more compact clusters, and the improved version displayed even clearer boundaries with fewer misclassified points.

Both the **Elbow method** and **Silhouette method** confirmed that k=3was the optimal number of clusters, with clear peaks in the plots, matching the known structure of the dataset.







Conclusion

In this project, I reproduced and extended the work from the paper "An Improved K-Means Algorithm Based on Evidence Distance" (Entropy, 2021). The original authors showed that replacing Euclidean distance with evidence distance leads to more accurate and stable clustering. My implementation confirmed these findings on the Iris dataset.

Building on their approach, I introduced **K-Means++ initialization** and **preprocessing techniques (scaling, PCA)** to address two issues: randomness in cluster center selection and difficulties in handling features of different scales or higher dimensionality. The results showed that these changes improved both **clustering quality** (higher ARI and Silhouette) and **efficiency** (fewer iterations to converge).

The key insight from this study is that **the choice of distance metric is critical in clustering, and combining evidence distance with better initialization and preprocessing further enhances performance.** Even on a simple dataset-like Iris, these improvements were measurable and consistent.

Future Work

For **future work**, there are several directions worth exploring:

- Testing other initialization methods or metaheuristic approaches (e.g., genetic algorithms, simulated annealing) to further optimize starting points.
- Applying ED-KMeans with these improvements to larger and more complex real-world datasets (e.g., image clustering, customer segmentation) to assess scalability.
- Experimenting with alternative distance measures such as Mahalanobis or cosine distance, or combining evidence distance with ensemble clustering techniques.

In summary, this project demonstrates that small but thoughtful improvements—like initialization and preprocessing—can significantly boost the effectiveness of clustering algorithms, and that evidence distance is a promising direction for enhancing traditional K-Means.

Links

GitHub - https://github.com/mekaviraj/ML_Assignment_2

Research paper - https://www.mdpi.com/1368836

Readme file - https://github.com/mekaviraj/ML_Assignment_2/blob/main/README.md