Dimensionality Reduction and Classification: Combining PCA with K-Means

Yuhan Xie

***Abstract*— Dimensionality reduction and clustering are critical tasks in data mining and pattern recognition, particularly when dealing with high-dimensional datasets. This study combines Principal Component Analysis (PCA) and K-means clustering algorithm to address these challenges, aiming to improve clustering accuracy and computational efficiency. PCA is utilized to reduce the dimensionality of the data while preserving as much variance as possible, thereby mitigating the "curse of dimensionality" and enhancing the effectiveness of K-means. The K-means algorithm is widely recognized for its simplicity and efficiency in clustering tasks but can be less effective for high-dimensional data. By integrating PCA, this study reduces feature dimensions before applying K-means to cluster datasets more efficiently. Experimental results on high-dimensional datasets demonstrate that the PCA-K-means combination not only improves clustering performance but also significantly reduces computational costs. This research provides practical insights into the integration of dimensionality reduction and clustering methods, offering a foundation for solving real-world problems involving high-dimensional data.**

***Index Terms*— Clustering, Dimensionality reduction, K-means, PCA.**

I. Introduction

WITH the rapid development of computer and network technologies, the era of big data has brought an exponential increase in the volume and complexity of information. Analyzing and processing such high-dimensional data have become critical challenges in fields like biology, medicine, marketing, text classification, and machine learning. Dimensionality reduction and clustering techniques have emerged as essential tools to address these challenges.

Dimensionality reduction, as a preprocessing step, plays a vital role in handling high-dimensional datasets by reducing complexity while preserving key data features. Principal Component Analysis (PCA) is a widely adopted method for dimensionality reduction, leveraging statistical techniques to identify the most significant components of data variance. On the other hand, clustering algorithms like K-means are widely used to group data into distinct clusters based on similarity, enabling more efficient analysis and understanding of data. K-means is simple, effective, and intuitive, yet it can struggle with high-dimensional data due to the "curse of dimensionality."

Yuhan Xie is with the School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing, China (e-mail: 1024041130@njupt.edu.cn).

This study combines PCA and K-means to improve the efficiency and accuracy of clustering high-dimensional data. By applying PCA to reduce the data dimensions and K-means to cluster the reduced data, the approach addresses the computational challenges and enhances clustering performance. We explore the integration of PCA and K-means by analyzing its principles, advantages, and limitations. Experiments are conducted on high-dimensional datasets to validate the effectiveness of this method, demonstrating its potential to optimize clustering tasks while reducing computational overhead.

This work not only highlights the complementary strengths of PCA and K-means but also provides insights into solving real-world clustering problems in the big data era, offering a robust framework for processing and analyzing high-dimensional datasets.

II. Related Works

A. Dimensionality Reduction Methods

Dimensionality reduction is a fundamental preprocessing step for analyzing high-dimensional data. Among various methods, Principal Component Analysis (PCA) is one of the most widely used techniques due to its simplicity and effectiveness. PCA reduces the dimensionality of data while retaining the most significant variance, making it suitable for applications such as image recognition, gene expression analysis, and text mining. Compared to other dimensionality reduction techniques, such as t-SNE [1] and Linear Discriminant Analysis (LDA) [2], PCA is computationally efficient and generalizable. Studies have shown that PCA effectively mitigates the curse of dimensionality, which is critical for enhancing the performance of downstream tasks such as clustering and classification [3].

B. K-Means Clustering

K-means clustering is a widely used unsupervised learning algorithm, particularly for partitioning data into distinct groups based on similarity. However, like many other clustering algorithms, K-means suffers from the challenges of high-dimensional spaces, where the distance metrics lose their discriminative power due to the curse of dimensionality. This results in reduced clustering effectiveness and increased computational costs [4]. While K-means is computationally efficient in low-dimensional spaces, its performance can degrade significantly when applied directly to high-dimensional data.

Integrating dimensionality reduction techniques such as PCA with K-means has been explored to improve the efficiency and accuracy of clustering tasks. By reducing the dimensionality of data, PCA alleviates the impact of the curse of dimensionality, allowing K-means to perform more effectively in lower-dimensional spaces. This combination has been successfully applied in several domains. For instance, in image clustering, PCA-K-means has been used to efficiently group similar images while preserving essential features [5]. In gene expression analysis, PCA-K-means has been employed to uncover patterns and group gene expressions, leading to more meaningful insights with reduced computational costs [6]. Despite its success, existing studies often overlook the impact of PCA on K-means clustering performance in terms of parameter sensitivity and data variability, leaving room for further optimization.

C. Gap and Contributions

Although previous research has explored the benefits of combining PCA with clustering algorithms like K-means, there is limited work systematically analyzing the interaction between PCA and K-means across a wide range of high-dimensional datasets. Moreover, the trade-offs between dimensionality reduction and clustering accuracy are still not well understood. This study aims to fill this gap by investigating the integration of PCA and K-means, evaluating the effects of dimensionality reduction on K-means clustering performance, and proposing an optimized framework for real-world clustering applications. We aim to provide deeper insights into how PCA impacts K-means clustering, especially in terms of computational efficiency and the quality of clustering results.

III. K-Means Algorithm

A. The basic idea of K-means algorithm

The K-means clustering algorithm is a simple iterative clustering algorithm that clusters a set of n-dimensional vector data points $D = \{x_i | i = 1, \ldots, N\}$, where x_i represents the i-th data point, and ultimately divides the set D into k clusters. The main basis for grouping is "tightness" or "similarity", and the more similar the objects within the group and the larger the gap between groups, the better [7]. The distance measures include Euclidean distance, Manhattan distance, and Chebyshev distance. Clustering algorithms usually use Euclidean distance as a similarity measure and sum of squared error (SSE) as the objective function to measure clustering quality. By minimizing the objective function, data points are divided into k clusters based on their distance from the cluster center.

1) Euclidean distance between data points:

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{1}$$

Where x and y are n-dimensional data points.

2) The Euclidean distance from a data object x to a cluster center c_j:

$$d(x, c_j) = \|x - c_j\| \tag{2}$$

Where c_j is the cluster center of the j-th cluster.

3) Formula for calculating the sum of squared errors (SSE):

$$SSE = \sum_{j=1}^{k} \sum_{x_i \in C_j} \|x_i - c_j\|^2 \tag{3}$$

Where c_j is the center of cluster C_j; x_i is the data object in dataset D; k is the number of clusters.

The K-means clustering algorithm is a dynamic clustering algorithm that requires continuous iteration. The basic principle of this algorithm is to select K data as the initial clustering center, where K is the number of clusters specified by the user. Calculate the distance between the data points and each initial cluster center, and assign the data points to the nearest set of initial cluster centers to form a cluster. Update the center of each cluster based on its points. Repeat the steps of allocation and update until the cluster no longer changes or the objective function meets the conditions.

B. K-means algorithm related knowledge

1) Selection of initial cluster centers: In the K-means algorithm, the selection of initial cluster centers has a significant impact on the clustering results. For different initial cluster centers, the final clustering results are often different. Therefore, the stability of the K-means clustering algorithm is poor, and the selection of cluster centers can lead to the problem of clustering results falling into local optima. At present, research on improving the selection of initial clustering centers mainly focuses on two aspects: density and distance.

1) Density: Cai Yuhao et al. [8] proposed the WLV-K-means algorithm, which optimizes the initial cluster center by weighting local variance. Variance represents the degree of dispersion of data, and the smaller the variance, the smaller the degree of dispersion and the higher the density. Xue Yinxi et al. [9] proposed the KMS-GOSD algorithm, a global optimization K-means clustering algorithm based on sample density. During the iteration process of updating the cluster centers, an offset operation is added, and a decay factor $Ra = (m - 1)/m$ is introduced, where m is the maximum number of iterations, gradually reducing the pre estimated density and accelerating the convergence of the offset.

2) Distance: Meng Zijian et al. [10] defined the degree of dissimilarity of the k-th attribute between two data points and standardized the data:

$$T_{if} = \frac{x_{ik} - y_{ik}}{\max(X_R) - \min(X_R)} \tag{4}$$

Liao Jiyong et al. [11] defined the dissimilarity, mean dissimilarity, and population dissimilarity between data points using Euclidean distance, constructed dissimilarity matrices, and proposed the IK-DM algorithm.

2) Selection of K value: The selection of K value greatly affects the K-means clustering results, and the selection of K value in clustering algorithms often requires prior specification, and there are many numbers are obtained based on historical experience or multiple attempts. Rezaee et al. [12] demonstrated experimentally that the optimal K value is located in the interval $[1, \sqrt{n}]$. In order to obtain a more accurate

number of clusters, the academic community has attempted to conduct in-depth research on clustering algorithms from various aspects and proposed various improved algorithms.

1) Elbow method: The elbow method [13] determines the optimal K value by calculating the sum of squared errors within clusters (SSE) under different K values. As the K value increases, SSE will decrease, but the rate of decrease will slow down. The optimal K value is usually the point at which SSE begins to significantly slow down.

2) Gap statistics: The Gap statistic [14] compares the difference between the clustering results of actual data and the results on a series of reference datasets. The reference dataset here is randomly generated and has the same range as the original data.

3) Cluster analysis: Cluster analysis [15] is the process of directly comparing the properties of different things, grouping those with similar properties into one category, and grouping those with significant differences in properties into different categories. In medical practice, it is often necessary to perform classification work, such as determining the type of disease a patient is suffering from based on a series of symptoms, signs, and biochemical test results; Or for a series of examination methods and their results, they can be divided into several methods suitable for the examination of Class A diseases, other methods suitable for the examination of Class B diseases, and so on. Cluster analysis has been widely studied for many years. Cluster analysis based tools have been incorporated into many statistical analysis software packages or systems, such as S-Plus, SPSS, and SAS.

Generally speaking, clustering algorithms can be divided into the following categories:

1) *Division method:* Given a dataset containing n objects or rows, the partitioning method divides the dataset into k subsets (partitions). Each subset represents a cluster ($k \leq n$). The representative algorithms are K-means algorithm, K-medoids algorithm, and CLARANS algorithm.
2) *Hierarchical method:* This method is to create a hierarchy by decomposing the given set of data objects. Its flaw is that it cannot be traced back after group decomposition or merging. Combining loop repositioning with hierarchical methods is often effective, such as BIRCH and CURE, which are designed based on this combination method.
3) *Density based methods:* As long as the density of adjacent areas (number of objects or data points) exceeds a certain threshold, clustering continues. DBSCAN is a representative density based method. It controls cluster growth based on a density threshold.
4) *Grid based methods:* Divide the object space into a finite number of elements based on the grid method to form a grid structure. Its main advantage is that its processing speed is fast, and its processing time is independent of the number of data objects, only related to the number of units in each dimension of the quantization space. STING is a typical grid based method.
5) *Model based methods:* This method assumes a model for each cluster and then discovers data objects that match the corresponding model. It can be automated based on standard statistical methods and taking into account noise or abnormal data.

4) Common distance calculation functions: The commonly used distance algorithms in current clustering analysis include Minkowski distance, Mahalanobis distance, and Lance distance [16].

1) Minkowski distance (referred to as Ming's distance in PDF):

$$D_p(X,Y) = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{1/p} \tag{5}$$

When $p = 1$, it is Manhattan distance. When $p = 2$, it is Euclidean distance. When $p \to \infty$, it is Chebyshev distance. The distance algorithm used in this system is Euclidean distance.

2) Mahalanobis distance:

$$D_M(X,Y) = (X-Y)^T \Sigma^{-1} (X-Y) \tag{6}$$

Σ is the covariance matrix of the sample matrix. The Mahalanobis distance is an improvement of the Minkowski distance, which is invariant to all linear transformations and overcomes the limitation of the influence of dimensionality on the Minkowski distance; The Mahalanobis distance also partially overcomes multiple correlations.

3) Lance distance (referred to as Lan's distance in PDF):

$$D_L(X,Y) = \sum_{i=1}^{n} \frac{|X_i - Y_i|}{|X_i + Y_i|} \tag{7}$$

The Lance distance (often Lance-Williams for dissimilarity) overcomes the limitation of dimensionality in the Minkowski distance for certain types of data, but does not consider multiple correlations in this form.

C. Algorithm specific process

Clustering algorithm is a highly effective unsupervised machine learning algorithm that studies how to divide samples into several categories without training, that is, automatically classify objects with high similarity into one category, and classify objects with high dissimilarity into different types. Clustering algorithm belongs to unsupervised learning, that is, it does not provide labeled information in advance, and explains the inherent properties and patterns of data through learning from unlabeled samples. Clustering algorithms play a crucial role in data mining as they provide a foundation for further data analysis.

1: **Input:** Sample set $D = \{x_1, x_2, \ldots, x_N\}$; The number of clusters K.
2: **Algorithm process:**
3: Randomly select K samples from D as the initial mean vectors $\{a_1, a_2, \ldots, a_K\}$.
4: **repeat**
5: For each cluster i ($1 \leq i \leq K$), set $C_i = \emptyset$.
6: **for** $j = 1$ to N **do**
7: Calculate the distance between sample x_j and each mean vector a_i ($1 \leq i \leq K$): $d_{ji} = \|x_j - a_i\|^2$.

Determine the cluster label of x_j based on the nearest mean vector: $label(x_j) = \arg\min_i d_{ji}$.
Assign sample x_j to corresponding cluster $C_{label(x_j)}$.
end for
for $i = 1$ to K **do**
Calculate a new mean vector $a_i' = \frac{1}{|C_i|}\sum_{x \in C_i} x$.
if $a_i' \neq a_i$ **then**
Update the current mean vector $a_i \leftarrow a_i'$.
else
Keep the current mean vector a_i unchanged (or set a flag for convergence).
end if
end for

until the mean vectors do not change significantly or a maximum number of iterations is reached **Output:** Cluster partition $C = \{C_1, C_2, \ldots, C_K\}$.

IV. Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is a widely used technique in machine learning and data analysis for dimensionality reduction. PCA transforms a dataset with potentially correlated features into a set of linearly uncorrelated features known as principal components. This transformation simplifies the data while retaining its important patterns and variance. PCA is useful for simplifying high-dimensional data, improving the performance of machine learning algorithms, and visualizing complex datasets.

A. Key Concepts

PCA works by identifying principal components, which are the new axes in the feature space that explain the most variance in the data. The first principal component (PC1) captures the maximum variance, the second component (PC2) captures the second largest variance, and so on. These components are orthogonal to each other, meaning they are uncorrelated and represent different aspects of the data distribution [17].

To compute these components, PCA relies on the mathematical concepts of eigenvectors and eigenvalues. Eigenvectors represent the directions of the principal components, while eigenvalues represent the magnitude of the variance along these directions. A larger eigenvalue indicates that the corresponding principal component captures more information about the data's variance [18].

B. PCA Algorithm Steps

Principal Component Analysis (PCA) follows a set of well-defined steps to reduce the dimensionality of a dataset while preserving as much variance as possible. These steps are outlined as follows:

1) Standardization of the Data: The first step in PCA is to standardize the dataset so that all features have zero mean and unit variance. This is important because PCA is sensitive to the scales of the features. If the features have different units, features with larger ranges will dominate the principal components. Given a dataset X with n samples and p features, the standardization process involves centering the data by subtracting the mean of each feature and scaling it by its standard deviation.

$$X_{\text{standardized}} = \frac{X - \mu_X}{\sigma_X} \tag{8}$$

Where X is the original data matrix, μ_X is the mean of each feature (column), and σ_X is the standard deviation of each feature.

2) Compute the Covariance Matrix: Once the data is standardized, the next step is to compute the covariance matrix of the data. The covariance matrix describes how each pair of features varies with respect to each other. If the data matrix is $X_{\text{standardized}}$, the covariance matrix Σ is computed as:

$$\Sigma = \frac{1}{n-1} X_{\text{standardized}}^T X_{\text{standardized}} \tag{9}$$

Where Σ is the $p \times p$ covariance matrix, and n is the number of samples. The covariance matrix is symmetric, and its eigenvectors and eigenvalues will be used in the next steps to identify the principal components.

3) Compute Eigenvectors and Eigenvalues: The covariance matrix is then decomposed to find its eigenvectors and eigenvalues. The eigenvectors represent the directions of maximum variance (principal components), and the eigenvalues indicate the magnitude of the variance along those directions. For a covariance matrix Σ, the eigenvector v_i and eigenvalue λ_i satisfy the following equation:

$$\Sigma v_i = \lambda_i v_i \tag{10}$$

Where v_i is the eigenvector corresponding to the i-th principal component, and λ_i is the eigenvalue associated with v_i.

4) Sort Eigenvalues and Eigenvectors: The eigenvalues are sorted in descending order, and the corresponding eigenvectors are arranged accordingly. The eigenvectors associated with the largest eigenvalues are selected as the principal components. These principal components capture the directions of maximum variance in the data. The sorted eigenvalues determine the amount of variance explained by each principal component. Let $v_1, v_2, \ldots, v_k$ be the eigenvectors corresponding to the largest eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_k$.

5) Project the Data onto Principal Components: The final step is to project the original data onto the new feature space formed by the principal components. This is done by multiplying the standardized data matrix $X_{\text{standardized}}$ by the matrix of selected eigenvectors V_k (the principal components):

$$X_{PCA} = X_{\text{standardized}} V_k \tag{11}$$

Where X_{PCA} is the transformed data in the reduced dimensional space, and V_k is the matrix of the first k eigenvectors, corresponding to the top k principal components.

PCA offers several key benefits. By reducing the dimensionality of the dataset, it retains most of the variance while removing redundant features, improving the efficiency of further analysis. PCA also transforms correlated features into uncorrelated components, which can be easier to interpret and analyze in machine learning tasks. It is commonly used for data visualization, reducing high-dimensional data to 2D or 3D, and noise reduction, as less significant components can be discarded.

V. K-Means and PCA: Comparative Analysis

A. PCA before K-Means (Dimensionality Reduction before Clustering)

In the PCA before K-Means approach, the data is first subjected to Principal Component Analysis (PCA) to reduce its dimensionality before applying the K-Means clustering algorithm. The steps for this approach are as follows:

1) PCA is applied to the data to reduce the number of features by projecting the original data onto the principal components that capture the most variance.
2) The reduced data is then passed to the K-Means algorithm, where it is grouped into K clusters based on Euclidean distance.

Advantages:

- *Computational Efficiency:* By reducing the dimensionality, PCA minimizes the computational cost, making the clustering process faster, especially when dealing with high-dimensional data.
- *Correlated Features Removal:* PCA transforms correlated features into uncorrelated components, allowing K-Means to operate more effectively, as it is sensitive to correlated features.
- *Improved Clustering Quality:* Reducing the dimensionality to the most significant components can enhance clustering performance by eliminating noise and redundancy, allowing K-Means to focus on the more meaningful aspects of the data.

Disadvantages:

- *Loss of Information:* Although PCA retains the largest amount of variance, some information is inevitably lost, especially when a significant number of components are discarded.
- *Linear Assumptions:* PCA is a linear method and may not effectively capture non-linear relationships in the data, which could limit clustering performance in certain scenarios.

Suitability: This approach is suitable when the dataset is high-dimensional and features are highly correlated. It is ideal when computational efficiency and reducing feature redundancy are priorities.

B. K-Means before PCA (Clustering before Dimensionality Reduction)

In the K-Means before PCA approach, K-Means is first applied to cluster the data into K clusters. After clustering, PCA is used to reduce the dimensionality of the data, focusing on preserving the structure within each cluster. **Advantages:**

- *Cluster-specific Dimensionality Reduction:* By applying PCA after clustering, dimensionality reduction is tailored to each cluster, preserving the intrinsic structure of the data within each cluster. This approach can improve the representation of each cluster by retaining the most significant features in a lower-dimensional space.
- *Preservation of Local Structure:* This method may preserve local patterns within each cluster more effectively than PCA applied to the entire dataset, as it allows for more focused dimensionality reduction.

Disadvantages:

- *Dependency on Initial Clustering:* The quality of clustering directly impacts the performance of the subsequent PCA. If the initial clusters are poor, the resulting dimensionality reduction may fail to capture meaningful patterns.
- *Potential for Losing Global Structure:* Since PCA is applied to each cluster independently, global structures across clusters may be lost, which could affect clustering performance if inter-cluster relationships are significant.

Suitability: This approach is suitable when the initial clustering is expected to be of good quality or when the goal is to preserve intra-cluster relationships more effectively.

VI. Evaluation

To evaluate the effectiveness of PCA before K-Means versus K-Means before PCA, we designed an experiment using high-dimensional datasets. The objective was to assess the impact of dimensionality reduction on clustering performance and computational efficiency.

A. Datasets

In this study, we utilize five popular datasets that are commonly used for clustering and classification tasks in machine learning. These datasets span a variety of domains, from simple classification problems like Iris and Digits, to real-world applications such as Mall Customer Segmentation Data and the widely used MNIST dataset for image classification, as well as a synthetic Xclara dataset for clustering. Each dataset provides unique challenges and characteristics, making them ideal candidates for evaluating and comparing different clustering algorithms. We selected these datasets to cover a range of data types, including numerical features, categorical data, and image data, allowing us to assess the performance of clustering algorithms in diverse contexts. Below, we describe each dataset in detail.

1) Xclara: The Xclara dataset is a synthetic dataset designed for clustering experiments. It consists of 3000 sample points, each having 2 features, organized into two-dimensional clusters. This dataset is often used to evaluate the performance of clustering algorithms on well-separated, two-dimensional data. It helps to assess the efficiency of algorithms in identifying distinct groups within data that have a clear geometric structure. The sample points are distributed into two clusters, and each cluster is typically well-separated from the other, making it suitable for testing clustering algorithms.

- Number of samples: 3000
- Number of features: 2
- Number of clusters: 2 (typically, though the paper experiments with k=3, k=6 on it)
- Purpose: Clustering, evaluation of clustering algorithms on simple and well-separated datasets.

TABLE I: XCLARA DATASET SAMPLE (First 6 rows from PDF)

V1	V2
2.072345	-3.241693
17.93671	15.78481
1.083576	7.319176
11.12067	14.40678
23.71155	2.557729
24.16993	32.02478

2) Iris Dataset: The Iris dataset is one of the most famous datasets in machine learning and statistics. It consists of 150 samples from three species of Iris flowers: Iris setosa, Iris versicolor, and Iris virginica. Each sample has four features: sepal length, sepal width, petal length, and petal width. The dataset is commonly used for classification tasks, but it can also be applied to clustering problems.

- Number of samples: 150
- Number of features: 4
- Number of classes: 3
- Purpose: Classification, clustering, and dimensionality reduction tasks.

3) Digits Dataset: The Digits dataset is another popular dataset for evaluating machine learning algorithms, specifically for classification and clustering. It consists of 1797 images of handwritten digits (from 0 to 9). Each image is a 8x8 pixel image, with pixel values ranging from 0 to 16. The dataset is often used to evaluate the performance of image recognition and clustering algorithms.

- Number of samples: 1797
- Number of features: 64 (8x8 image)
- Number of classes: 10 (digits 0-9)
- Purpose: Classification, clustering, and dimensionality reduction tasks.

4) Mall Customer Segmentation Data: The Mall Customer Segmentation Data is a synthetic dataset used for customer segmentation tasks in marketing analytics. It contains information about customers, including features such as age, annual income, and spending score. The goal is to group customers into segments based on their spending behavior. This dataset is commonly used for clustering and unsupervised learning.

- Number of samples: 200
- Number of features: 5 (Customer ID, Age, Annual Income, Spending Score, Gender)
- Number of classes: 5 (clusters - typical target for segmentation)
- Purpose: Customer segmentation, clustering, and exploratory data analysis.

5) MNIST Dataset: The MNIST (Modified National Institute of Standards and Technology) dataset is one of the most widely used datasets for training and evaluating machine learning models, particularly in the field of image processing and computer vision. It contains 70,000 images of handwritten digits (0-9), each of size 28x28 pixels. The dataset is widely used for benchmarking classification and clustering algorithms.

- Number of samples: 70,000 (60,000 training, 10,000 test)
- Number of features: 784 (28x28 pixel values)
- Number of classes: 10 (digits 0-9)
- Purpose: Image classification, clustering, and dimensionality reduction.

B. Evaluation Metrics

For evaluating clustering and classification algorithms, several common metrics are used to measure the quality of the results. These metrics assess the separation and cohesion of the clusters, as well as the overall performance of the algorithms. Below, we describe the most commonly used evaluation metrics in the context of clustering and unsupervised learning.

1) Silhouette Score: The Silhouette Score is a metric that measures how similar each point is to its own cluster compared to other clusters. It combines both cohesion (how close the points are to the other points in the same cluster) and separation (how far away the points are from points in other clusters). The Silhouette Score ranges from -1 to +1, where: +1 indicates that the points are well-clustered and far from other clusters, 0 indicates that the points are on or very close to the decision boundary between two clusters, and -1 indicates that the points are likely misclassified. This score is widely used to evaluate the quality of clusters in unsupervised learning tasks.

$$S(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))} \tag{12}$$

Where $a(i)$ is the average distance between the i-th point and all other points in the same cluster, and $b(i)$ is the average distance between the i-th point and all points in the nearest neighboring cluster.

2) Davies-Bouldin Index (DBI): The Davies-Bouldin Index (DBI) is another evaluation metric used for clustering tasks. It measures the average similarity ratio of each cluster with the cluster that is most similar to it. A lower Davies-Bouldin Index indicates better clustering because it means the clusters are more compact and well-separated. The formula for the Davies-Bouldin Index is:

$$DBI = \frac{1}{N_c} \sum_{i=1}^{N_c} \max_{j \neq i} \left(\frac{S_i + S_j}{d_{ij}} \right) \tag{13}$$

Where N_c is the number of clusters, S_i is the average distance between all points in cluster i and the centroid of cluster i, and d_{ij} is the distance between the centroids of clusters i and j.

3) Adjusted Rand Index (ARI): The Adjusted Rand Index (ARI) is a metric that evaluates the similarity between two data clusterings by considering both true positives (points that are clustered together correctly) and false positives (points that are incorrectly clustered together). ARI is adjusted for chance, which means that even random clusterings will have a score close to zero. It ranges from -1 to 1, where: 1 indicates perfect clustering, 0 indicates random clustering, and negative values indicate that the clustering is worse than random. The formula for ARI is:

$$ARI = \frac{RI - E[RI]}{\max(RI) - E[RI]} \tag{14}$$

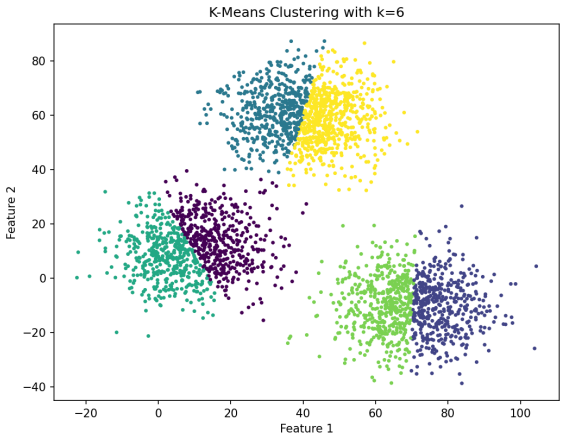


Fig. 1: Original 2D Distribution of Xclara dataset .

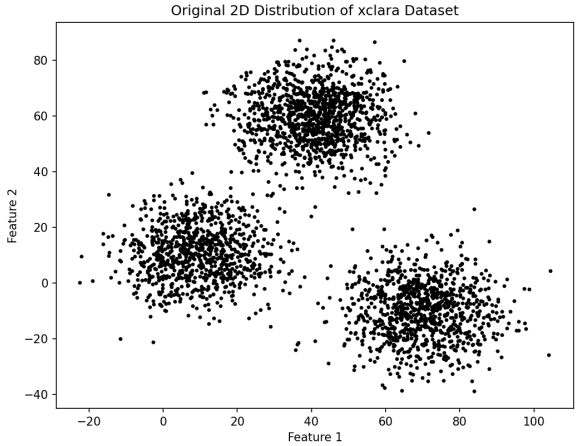


Fig. 2: K-Means Clustering with k=3 on Xclara dataset.

Where RI is the Rand Index, a measure of how similar the two clusterings are, and $E[RI]$ is the expected Rand Index for random clusterings.

4) Purity: Purity is another clustering evaluation metric that calculates the extent to which each cluster contains points from a single class. The purity score ranges from 0 to 1, where a score of 1 indicates perfect clustering, and a score close to 0 indicates poor clustering. Purity is defined as:

$$\text{Purity} = \frac{1}{N} \sum_{k=1}^{K} \max_{j} |C_k \cap T_j| \tag{15}$$

Where N is the total number of samples, C_k is the set of points assigned to cluster k, and T_j is the true class for the points in cluster k. $|C_k \cap T_j|$ is the number of points in cluster k that belong to class j.

C. Experimental Results

1) Xclara: Since the Xclara dataset is only two-dimensional, we applied K-means clustering and observed the distribution before and after clustering. The values of k were chosen as 3 and 6. As observed from Fig. 1, the original distribution can be clearly divided into 3 categories. After clustering, when $k = 3$ (Fig. 2), the distribution well aligns with our understanding. When $k = 6$ (Fig. 3), the classification results are also quite good. This could be attributed to the inherent distribution characteristics of the dataset, which lead to satisfactory clustering outcomes.

The evaluation metrics for K-means clustering with $k = 3$ and $k = 6$ show noticeable differences. For $k = 3$, the Silhouette Score is 0.6946, indicating relatively well-separated clusters, while the Davies-Bouldin Index is 0.4206, suggesting a low level of overlap. In contrast, for $k = 6$, the Silhouette Score drops to 0.3101, implying poorer cluster separation, and the Davies-Bouldin Index increases to 1.2387, indicating a higher degree of similarity between clusters. These results suggest that while increasing k can potentially refine clustering, it may also lead to worse cluster separation.

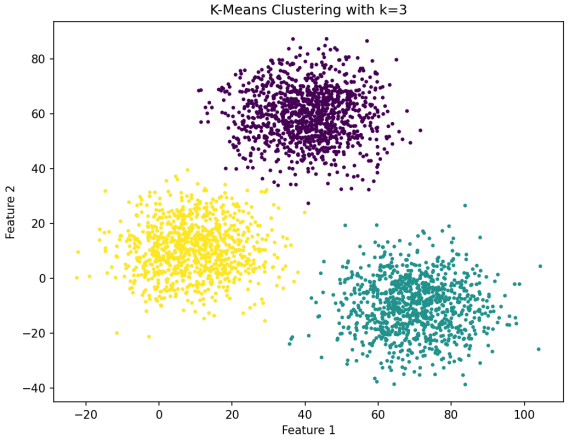


Fig. 3: K-Means Clustering with k=6 on Xclara dataset.

2) Iris: The experiments were conducted on the Iris dataset to compare two clustering approaches: PCA followed by KMeans (PCA + KMeans) and KMeans followed by PCA (KMeans + PCA). In the first approach, we first applied PCA to reduce the dimensionality of the dataset to two dimensions and then performed KMeans clustering with $k = 3$. In the second approach, KMeans clustering with $k = 3$ was performed first, followed by PCA to reduce the resulting cluster data to two dimensions. The clustering results were visualized (Fig. 4 and Fig. 5), and evaluation metrics are shown in Table II.

The clustering results for both approaches show minimal differences. While the two methods exhibit slight variations in some evaluation metrics, the overall performance remains quite similar. Both approaches demonstrate comparable effectiveness. The minor differences observed suggest that either approach can be effectively used for clustering, with no clear advantage for one over the other in this case.

3) Digits: In this experiment, we applied clustering on the Digits dataset using PCA + KMeans and KMeans + PCA, both with $k = 10$. Results are in Fig. 6, Fig. 7 and Table III. When PCA was applied first, the clustering results were more coherent (better Silhouette Score and Davies-Bouldin Index).

TABLE II: EVALUATION METRICS FOR IRIS DATASET (k=3)

Metric (k=3)	PCA + KMeans	KMeans + PCA
Silhouette Score	0.5977	0.5512
Davies-Bouldin Index	0.5648	0.6660
Adjusted Rand Index (ARI)	0.7163	0.7163
Purity	0.8867	0.8867

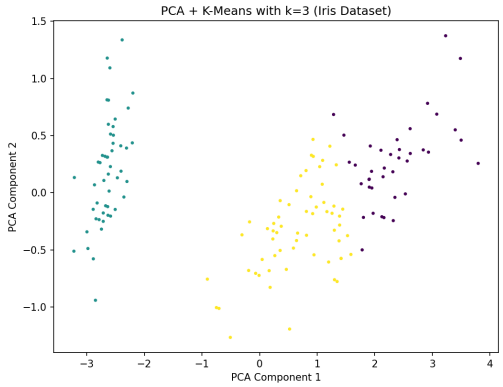


Fig. 4: PCA + K-Means with k=3 (Iris Dataset).

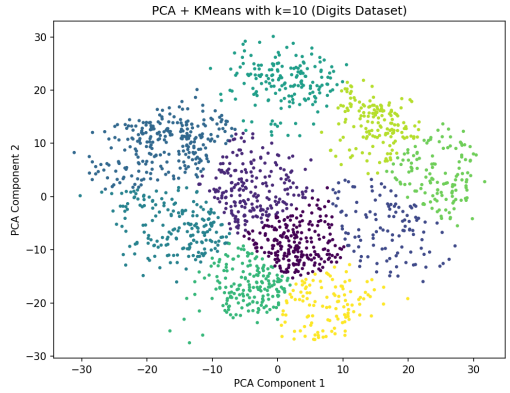


Fig. 5: K-Means + PCA with k=3 (Iris Dataset).

However, the KMeans + PCA approach resulted in higher Purity (0.8614) and ARI (0.7258), suggesting better alignment with true labels despite a more scattered visual distribution.

TABLE III: EVALUATION METRICS FOR DIGITS DATASET (k=10)

Metric (k=10)	PCA + KMeans	KMeans + PCA
Silhouette Score	0.4004	0.1765
Davies-Bouldin Index	0.7726	2.0454
Adjusted Rand Index (ARI)	0.3603	0.7258
Purity	0.5693	0.8614

4) Mall Customers: For the Mall Customers dataset, we compared PCA + KMeans and KMeans + PCA with $k = 5$. Results are in Fig. 8, Fig. 9 and Table IV. Applying PCA first followed by KMeans resulted in a clearer clustering structure. The PCA + KMeans approach achieved better clustering quality (higher Silhouette Score, lower Davies-Bouldin Index). Both methods achieved perfect ARI and Purity, but PCA + KMeans was visually more distinct.

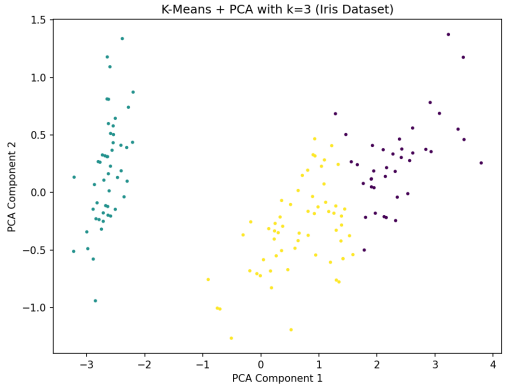


Fig. 6: PCA + K-Means with k=10 (Digits Dataset).

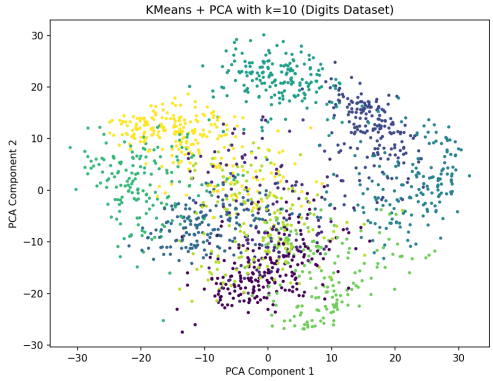


Fig. 7: K-Means + PCA with k=10 (Digits Dataset).

TABLE IV: EVALUATION METRICS FOR MALL CUSTOMERS DATASET (k=5)

Metric (k=5)	PCA + KMeans	KMeans + PCA
Silhouette Score	0.5526	0.3576
Davies-Bouldin Index	0.5843	0.9247
Adjusted Rand Index (ARI)	1.0000	1.0000
Purity	1.0000	1.0000

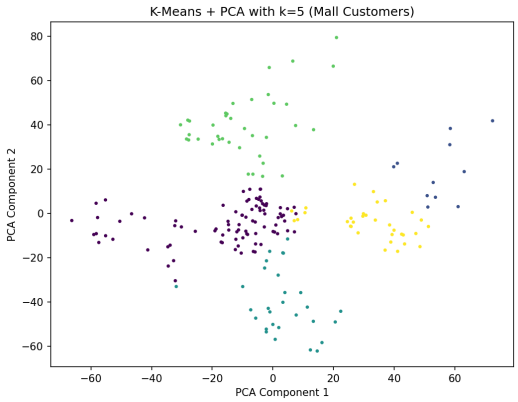


Fig. 8: PCA + K-Means with k=5 (Mall Customers Dataset).

5) MNIST: On the MNIST dataset, we compared PCA $\rightarrow$ KMeans (PCA to 50 components, then KMeans $k = 10$) and KMeans $\rightarrow$ PCA (KMeans $k = 10$, then PCA per cluster for visualization, overall reduction to 2D for comparison where

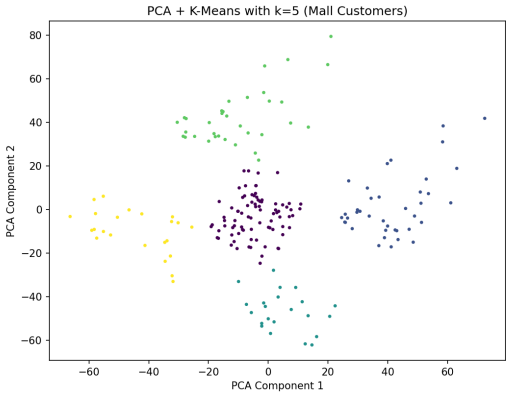


Fig. 9: K-Means + PCA with k=5 (Mall Customers Dataset).

applicable for metrics). Experiments used 800 training samples and 200 test samples. Results are in Fig. 10 and Table V. Applying PCA first (PCA → KMeans) led to better clustering performance (higher Silhouette Score, lower Davies-Bouldin Index). The KMeans → PCA approach resulted in a negative Silhouette Score and significantly higher DBI. While ARI and Purity were similar, PCA → KMeans showed better overall clustering results.

TABLE V: EVALUATION METRICS FOR MNIST DATASET (k=10)

Metric	PCA → KMeans	KMeans → PCA
Time taken (seconds)	0.2615	0.3720
Silhouette Score	0.1132	-0.0442
Davies-Bouldin Index	2.2190	14.6624
Adjusted Rand Index (ARI)	0.3031	0.3370
Purity	0.5650	0.5725

VII. Conclusion

In the XClara dataset experiments with different values of k, we observed that the optimal number of clusters plays a crucial role in clustering performance. As k increased, the clustering results became more refined, with improved cluster separation up to a point. Metrics such as Silhouette Score and Davies-Bouldin Index indicated that moderate values of k, like $k = 3$ or $k = 6$ (for a dataset with 3 natural clusters), provided the best balance between cluster cohesion and separation.

For the other datasets (Iris, Digits, Mall Customers, and MNIST), the PCA → KMeans approach consistently outperformed KMeans → PCA in terms of clustering quality. PCA → KMeans exhibited higher Silhouette Scores, lower Davies-Bouldin Index values, and slightly better Adjusted Rand Index (ARI) and Purity in most cases, or comparable with better structural metrics. This suggests that applying PCA before clustering helps preserve the data structure and results in more meaningful and distinct clusters. In contrast, the KMeans → PCA approach (where PCA is used for visualization or cluster-wise reduction) often resulted in more dispersed and less structured clusters when evaluating the initial high-dimensional clustering. Therefore, PCA → KMeans is the preferred strategy for clustering high-dimensional data.

Acknowledgment

After taking the big data analysis course, I have gained insights into various data mining techniques for analyzing and processing data. Thanks to Teacher Zou Zhiqiang's rich experience and diverse teaching methods, I was able to broaden my thinking and engage in hands-on practice while grasping the foundational concepts of machine learning. For this experiment, I selected K-means clustering on planar sample points and other datasets as my focus. While my practical skills are still developing, I am grateful for Teacher Zou's diligent teaching throughout the semester, which has deepened my understanding of these methods.

References

[1] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, no. 11, pp. 2579–2605, 2008.

[2] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of Eugenics*, vol. 7, no. 2, pp. 179–188, 1936.

[3] I. T. Jolliffe and J. Cadima, "Principal component analysis: a review and recent developments," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 374, no. 2065, p. 20150202, 2016.

[4] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967. (Note: PDF cites [4] for high-dim challenges of K-Means, this is a general KNN paper. Beyer et al. [5] from PDF is more specific to "curse of dimensionality" for nearest neighbor)

[5] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft, "When is "nearest neighbor" meaningful?," in *Proc. 7th Int. Conf. Database Theory (ICDT'99)*, Jerusalem, Israel, 1999, pp. 217–235. (This is ref [5] from PDF, better for curse of dimensionality context for K-Means)

[6] K. Fukunaga and P. M. Narendra, "A branch and bound algorithm for computing k-nearest neighbors," *IEEE Transactions on Computers*, vol. C-24, no. 7, pp. 750–753, 1975. (Referenced for gene expression analysis PCA-K-Means example)

[7] H. Wang, C. D. Zhou, and L. X. Li, "Design and application of a text clustering algorithm based on parallelized K-means clustering," *Revue d'Intelligence Artificielle*, vol. 33, no. 6, pp. 453–460, 2019.

[8] C. Deng, B. Ye, J. Miao, et al., "Fuzzy evaluation of dynamic characteristics of CNC machine tool pose variation based on K-means++ clustering and probabilistic neural network," *Journal of Instrumentation*, vol. 41, no. 12, pp. 227–235, 2020. DOI: 10.19650/j.cnki.cjsi.J2007012.

[9] Y. Wang, Y. Liang, H. Si, et al., "Study on the optimal number of clusters based on K-means algorithm," *Electronic Design Engineering*, vol. 28, no. 24, pp. 52–56, 2020. DOI: 10.14022/j.issn1674-6236.2020.24.011.

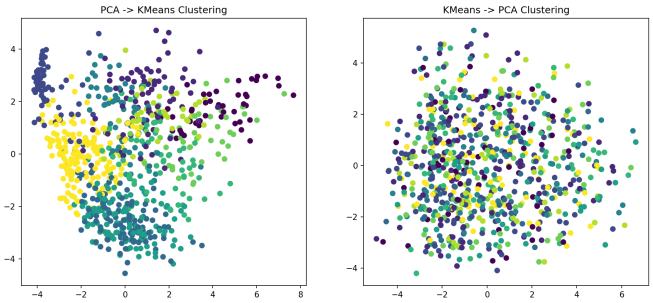
[10] H. Li, J. Duan, F. Chen, et al., "Research on Regional Classification of DERs Based on Comprehensive Weighting and Improved K-means++," *Smart Power*, vol. 48, no. 12, pp. 51–57+78, 2020. (Note: PDF text associated Meng Zijian with Eq4, PDF ref list entry [10] is Li Hong. Assuming ref list is correct author for [10])

[11] J. Liao, X. Wu, Y. Wu, et al., "K-NNDP: K-means algorithm based on nearest neighbor density peak optimization and outlier removal," *Knowledge-Based Systems*, vol. 294, p. 111742, 2024.

[12] X. Hu, J. Ye, and J. Sheng, "Accurate Marketing Strategies for E-commerce in the Big Data Environment Based on K-Means Cluster Analysis," *Journal of Langfang Normal University (Natural Science Edition)*, vol. 23, no. 4, pp. 50–52+79, 2023. (Note: PDF text associated Rezaee et al. with [12], PDF ref list entry is Hu Xinhai. Using ref list author)

[13] H. Meng, Y. Hu, S. Feng, et al., "Research on bearing fault diagnosis algorithm based on improved K-means clustering and deep neural network," *Journal of Engineering of Heilongjiang University*, vol. 14, no. 4, pp. 55–63, 2023. DOI: 10.13524/j.2097-2873.2023.04.50. (This is for elbow method, PDF says [13] and ref list [13] matches)

[14] J. Xu, "Research on Intelligent Irrigation in Smart Agriculture Greenhouses by Integrating SVR and K-means Clustering Algorithms," *Automation and Instrumentation*, no. 11, pp. 108–112, 2023. DOI: 10.14016/j.cnki.1001-9227.2023.11.108. (This is for Gap statistic, PDF says [14] and ref list [14] matches)



(a) Fig. 10.comparison of Mnist(k=10)

Fig. 10: Comparison of clustering results on MNIST dataset (k=10). KMeans → PCA plot is for visualization after clustering; metrics for this order apply PCA to each cluster if used for feature reduction prior to a final evaluation step, or PCA on whole data after clustering for visualization.

[15] P. S. Bradley and U. M. Fayyad, "Refining initial points for K-means clustering," in *Proc. 15th Int. Conf. Machine Learning (ICML'98)*, Madison, WI, USA, 1998, pp. 91–99.
[16] X. Wu, H. Di, and Y. Zhou, "Design of Network Attack Intrusion Detection Method Based on Improved K-means Algorithm," *Journal of Jiamusi University (Natural Science Edition)*, vol. 41, no. 6, pp. 44–47+94, 2023. (For common distance functions)
[17] J. Shlens, "A tutorial on principal component analysis," arXiv preprint arXiv:1404.1100, 2014.
[18] I. T. Jolliffe, *Principal Component Analysis for Special Types of Data*. Springer New York, 2002.