Project Report - Probabilistic Graphical Models K-means on PCA (Principal Component Analysis)

Rania Bennani
rania.bennani@ens-paris-saclay.fr

Esteban Christiann
esteban.chrisitann@ens-paris-saclay.fr

Raphael Razafindralambo
raphael.razafin@gmail.com

1 INTRODUCTION

In this report, we tackle the exploration and integration of Principal Component Analysis (PCA) within the context of the K-means clustering algorithm. The combination of these two techniques holds significant importance in enhancing the performance and interpretability of clustering results.

We will first give a theoretical insight into k-means and PCA algorithms. Then, we will explain, on a more practical side, the datasets we generated to test our algorithms and the metrics to evaluate our models. Finally, we will (EM Algo ?)

2 BACKGROUND ON K-MEANS AND PCA

2.1 K-means

K-means is a popular and widely used unsupervised learning algorithm in data mining and machine learning. Being an unsupervised algorithm, K-means does not rely on predefined labels or outcomes for training. Instead, it discovers patterns and structures directly from the input data. It is primarily used for clustering, which involves partitioning a dataset into distinct groups based on similarity. The main idea behind K-means is to identify K centroids, one for each cluster, and then assign each data point to the nearest centroid, thereby forming clusters.

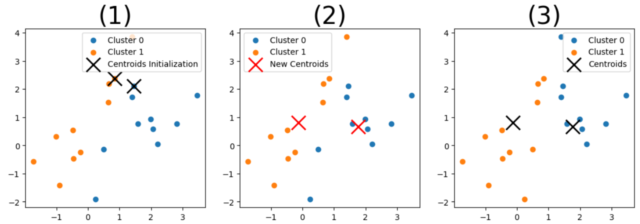

Figure 1: K-means clustering steps with centroids and clusters update. (1) shows centroids initialization and the first membership assignment, (2) shows the update of the centroids according to the new clusters, and (3) shows the second membership assignment.

Let $\mathcal{X} = \{x_1, \dots, x_n\}$ be a set of points in an euclidean space E. In the following, we denote $\|.\|$ the euclidean distance, and we assume $E = \mathbb{R}^d$ with $d \geq 1$. As an unsupervised learning algorithm, the goal of K-means clustering is to partition $\mathcal{X}$ into K clusters $C_1, \dots, C_K$ represented by the set of centroids $\mathcal{M}_K = \{m_1, \dots, m_K\} \subset \mathbb{R}^d$, without any prior knowledge of the group memberships of the data points. The clustering is formulated as the following optimization problem:

$$\min_{m_1,\dots,m_K} J_K(m_1, \dots, m_K) = \min_{m_1,\dots,m_K} \sum_{k=1}^{K} \sum_{x_i \in C_k} \|x_i - m_k\|^2$$

This optimization problem is non-convex. However, a method first developed in 1957 by S.P. Lloyd allows for a numerical finding of a local solution to this problem. *The Lloyd's algorithm in pseudocode, which is today the most well-known for solving the K-means problem is given in Appendix* A.

We note that the convergence of the K-means algorithm is due to the sequence of updated within-cluster variances $J_K^{(t)}$ that decrease while remaining positive across iterations. Additionally, this sequence becomes stationary because the cost function is limited to taking only a finite number of values.

2.2 Principal Component Analysis (PCA)

Principal Component Analysis (PCA) stands as a fundamental technique for extracting essential information from high-dimensional datasets. By reducing dimensionality while retaining crucial features, PCA facilitates a more interpretable and computationally efficient representation of the data.

Consider a dataset represented by the matrix $X = (x_1 \ \dots \ x_n)^\top$, where n is the number of observations and d is the dimensionality of each observation. The covariance matrix $\frac{X^\top X}{n}$ provides valuable insights into the relationships between variables.

Let $X = USV^\top$ be the Singular Value Decomposition (SVD) [1] of X, where U, V are orthogonal matrices, and S is rectangular with non-negative reals $\sigma_1 \leq \dots \leq \sigma_d$ on the diagonal. Therefore,

$$\frac{X^\top X}{n} = V \left(\frac{S^2}{n}\right) V^\top \tag{1}$$

and the $\lambda_i = \frac{\sigma_i^2}{n}$ are the eigenvalues of the covariance matrix and represents the variance along each principal direction. The columns $v_1, \dots, v_d$ of V are the eigenvectors. In the context of Principal Component Analysis (PCA), $v_1, \dots, v_d$ are referred to as the principal directions, and the columns of US are the principal components.

2.3 Relationship between K-means and PCA

A K-way clustering can be represented by a discrete matrix H_K of size $n \times K$, where each element is either 0 or $\oplus$. Here, $\oplus$ is a positive value, and its presence at $[H_K]_{i,k}$ indicates that element x_i

belongs to cluster C_k. Ding & He [2] demonstrate that the K-means objective can be expressed as:

$$J_K = \mathrm{Tr}(XX^\top) - \mathrm{Tr}(H_K^\top XX^\top H_K) \tag{2}$$

Relaxing the discrete constraint on H_K leads to the so-called *continuous K-means*, which admits an elegant closed-form solution. By applying an isometry T to the rows of H_K and denoting the resulting matrix as Q_K, and then extracting the first $K-1$ columns to form Q_{K-1}, the K-means objective can be further expressed as:

$$J_K = \mathrm{Tr}(XX^\top) - \mathrm{Tr}(Q_{K-1}^\top XX^\top Q_{K-1}) \tag{3}$$

Minimizing J_K is thus equivalent to maximizing $\mathrm{Tr}(Q_{K-1}^\top XX^\top Q_{K-1})$. Ding & He [2] provide the following theorem:

Theorem 2.1 (Ding & He [2]). *The solution of the continuous K-means objective is given by $Q_{K-1} = (u_1 \ \dots \ u_{K-1})$, the first $K-1$ columns of U. Additionally, if $J_K^\star$ is optimal, then:*

$$\overline{nx^2} - n\sum_{i=1}^{K-1}\lambda_i \leq J_K^\star < \overline{nx^2}$$

where $\overline{x^2} = \frac{1}{n}\sum_{i=1}^n x_i^\top x_i$.

The upper bound corresponds to assigning all the data points to the same cluster and is not very informative. However, the lower bound is insightful when expressed differently. Recognizing that $\overline{nx^2}$ is the squared Frobenius norm of X, $||X||_F^2 = ||USV^\top||_F^2 = ||S||_F^2 = \sum_{i=1}^d \sigma_i^2 = n\sum_{i=1}^d \lambda_i$, we get:

$$\lambda_K + \ldots + \lambda_d \leq \frac{1}{n}J_K^\star < \lambda_1 + \ldots + \lambda_d$$

This bound has a meaningful interpretation: an optimal K-way clustering cannot eliminate more variance than there is in the $K-1$ principal directions.

It's essential to note that while Ding & He [2] originally stated a strict inequality, we can construct datasets where the bound can be reached. The key is to ensure that centroids are distant enough, and the data inside each cluster should lie in a subspace orthogonal to the one spanned by the first $K-1$ directions (see fig. 5).

Moreover, another connection between K-means and PCA can be made with this result:

Theorem 2.2 (Ding & He [2]). *The cluster centroids of the continuous K-means solution lie in the subspace spanned by the first $K-1$ principal directions $v1, \ldots, v_{K-1}$.*

However, it's important to note that this theorem pertains to centroids in the continuous K-means formulation. This result is false in general for discrete K-means; the centroids of an optimal clustering may not belong to the PCA subspace (see fig. 6)

2.4 K-means initialization methods

There exist various initialization methods for K-means [3]. This subsection outlines six of the most notable ones.

Random initialization: It consists in drawing the centroids $\{m_1, \ldots, m_K\}$ uniformly according to the uniform distribution over the x_i. In practice, this is not effective. If the initialization is poorly chosen, the local minimum reached by the algorithm could be very far from the optimal solution.

K-means++: Arthur and Vassilvitskii [4] proposed a greedy method for selecting K centroids. The initialization is still characterized as random, but this method increases the probability that the chosen initial centroids are spaced out. The first centroïd m_1 is chosen randomly according to $\mathcal{U}(\{x_1, \ldots, x_n\})$. Then, for any step $k \geq 2$ if we denote $d : x \mapsto \min_{m\in\{m_1,\ldots,m_{k-1}\}} ||x - m||$ the function that associates each x with its Euclidean distance to the nearest centroid already selected, the k-th centroid m_k is chosen among $\mathcal{X}^k = \mathcal{X} \setminus \{m_1, \ldots, m_{k-1}\}$ with the probability

$$p(x) = \frac{d^2(x)}{\sum_{x_i\in\mathcal{X}^k} d^2(x_i)}$$

This is repeated until K centroids are identified.

KR algorithm: Kaufman and Rousseeuw [5] devised a method where initial centroids are chosen successively. The first chosen point is the less dissimilar point to others (central point). The subsequent points are selected to ensure their dissimilarity with all points is significantly less than these points' dissimilarity to the selected centroids.

$$m_k = \begin{cases} \arg\min\limits_{x_i\in\mathcal{X}} \sum_{j=1}^n d(x_i, x_j) & \text{if } k = 1 \\ \arg\max\limits_{x_i\in\mathcal{X}^k} \left(\sum_{j=1}^n \max\left(d(x_j, \{m_i\}_{i=1}^{k-1}) - d(x_j, x_i), 0\right)\right) & \text{if } 2 \leq k \leq K \end{cases}$$

where $\mathcal{X}^k = \mathcal{X} \setminus \{m_1, m_2, ..., m_{k-1}\}$ and d is any dissimilarity measure (e.g. $\|.\|$).

KKZ (Ioannis Katsavounidis, C.-C. Jay Kuo, and Ben Zhang) Algorithm: Katsavounidis et al. [6] suggested selecting the first centroid based on the maximum norm. For each non-selected data point (candidate point), the minimal distance to the centroid set is set, and the next centroid is the candidate point with the maximal distance from the centroids. This process repeats until K centroids are chosen. In this method,we pay attention to the points that are most far apart from each other because they are more likely to belong to different classes. The centroids are set as:

$$m_k = \begin{cases} \arg\max\limits_{x_i\in\mathcal{X}} \|x_i\| & \text{if } k = 1 \\ \arg\max\limits_{x_i\in\mathcal{X}^k} \left(\min\limits_{j=1,\ldots,k-1} \|x_i - m_j\|\right) & \text{if } 2 \leq k \leq K \end{cases}$$

where $\mathcal{X}^k = \mathcal{X} \setminus \{m_1, m_2, ..., m_{k-1}\}$.

HAC Method: Some studies [7–9] utilized hierarchical agglomerative clustering (HAC) results for centroid initialization. HAC employs a "bottom-up" approach, starting with numerous small clusters and progressively merging the closest ones until only one cluster remains. The inter-cluster distance, denoted as $d(k, l)$, is a crucial factor, and the method described in [7] is adopted in our experiments.

PCA-part Method: Su and Dy [10] introduced a "top-down" approach using recursive partitioning for initialization. They suggest recursively bi-partitioning a single cluster C_1 by computing the first principal component Φ_1. and using its sign to split the cluster. Then the next cluster to partition is chosen by selecting the cluster C_k

with the largest sum of squared errors $\sum_{i=1}^{n} \|x_i - \mu(C_k)\|^2$ where $\mu(C_k)$ denotes the mean of C_k. In each iteration, the cluster with the maximum distortion is split until K clusters are obtained. These clusters are then used for K-means initialization, referred to as the PCA-part method.

PCA-guided search: It consists of performing standard K-means in the PCA subspace. Then we use the obtained cluster membership to construct initial centroids in the full space. This method is based on the following result

The PCA K-dimentional subspace is much smaller than the original d-dimensional data space. Consequently, the computation of the distances is faster than in the full space. Moreover, it gives a better solution in terms of the objective because it searches in a restrained and relevant subspace.

PCA-solved continuous K-means: With PCA, we can get a solution of the continuous K-means problem [2]. Ding & He also propose a way to recover a clustering from this solution using *linearized clustering assignment* [11]. One should examine a rough $K-1$-rank approximation of the $n \times n$ similarity matrix $XX^\top$ given by $\sum_{k=1}^{K-1} u_k u_k^\top$, to which we add a constant matrix $ee^\top/n$, where e is the $n \times 1$ vector full of ones. Hence, let $C = ee^\top/n + \sum_{k=1}^{K-1} u_k u_k^\top$. Ding & He then prove that $C = \sum_{k=1}^{K} h_k h_k^\top$.

By the definition of H_K, if the data points are indexed such that all data points within the same cluster are adjacent, C will have a block-diagonal structure. Unfortunately, there is no reason that our indexing already reveals this block-diagonal structure.

Ding & He's idea is to then use a *spectral ordering* algorithm [11]. This *spectral ordering* algorithm will find an ordering o that will reveal this block-diagonal structure if we permute the rows and columns of C following o. Now, we have to find the correct decomposition of the matrix into blocks on the diagonal. This problem can be reduced to finding valley points of a 1-D quantity. Indeed, consider the sum along the i-th antidiagonal. If the coefficient $C_{o(i),o(i)}$ is close to a corner of a block, the antidiagonal will contain few non-zero coefficients. Hence the sum along it will be low. Hence by finding valleys points of this quantity, we can identify the blocks and hence the clusters. This is the main conceptual idea behind this *linearized cluster assignment* algorithm [11]: first recover the correct ordering and then identify the blocks' corners, so the clusters.

3 EXPERIMENTAL SETUP AND DATA EXPLORATION

3.1 Synthetic datasets:

To assess the performance of different initialization methods, we conducted experiments using synthetic data on three distinct datasets. Our objective is to comprehensively evaluate performance under diverse scenarios. The first dataset is designed to yield optimal results, specifically tailored for the k-means algorithm. The second dataset aims to underscore the limitations inherent in k-means clustering. Finally, the third scenario involves a substantial dataset to evaluate performance at scale. We set the seed value to 42.

- **Dataset 1 (GM - 3 Clusters)**: A 2-dimensional Gaussian mixture with three distinct, well-separated clusters. This dataset is used to analyze the algorithm's performance in scenarios with clear, distinct groupings.
  Parameters: $\mu_1 = (0,0)^T, \mu_2 = (12,12)^T, \mu_3 = (-7,-7)^T, \Sigma_1 = I_1, \Sigma_2 = 2I_2, \Sigma_3 = I_2/2, \pi_1 = 0.4, \pi_2 = \pi_3 = 0.3$
- **Dataset 2 (Moons)**: Comprising two crescent moon-shaped clusters in $\mathbb{R}^2$, this dataset evaluates the algorithm's performance on non-linearly separable data, challenging its adaptability to complex shapes.
- **Dataset 3 (MNIST)**: The well-known MNIST dataset represents high-dimensional, real-world data. It contains a training set of 60,000 examples. Each image is a grayscale representation of a handwritten digit (from 0 to 9) and is 8x8 pixels in size. We flatten the data to make vectors. MNIST is included to test the algorithm's scalability and effectiveness in a more practical context.
- **Dataset 4 (Faces)**: The Olivetti faces dataset contains images of human faces taken between April 1992 and April 1994 at AT&T Laboratories Cambridge. There are images of 40 distinct subjects, with different lighting, facial expressions (open / closed eyes, smiling / not smiling) and facial details (glasses / no glasses).

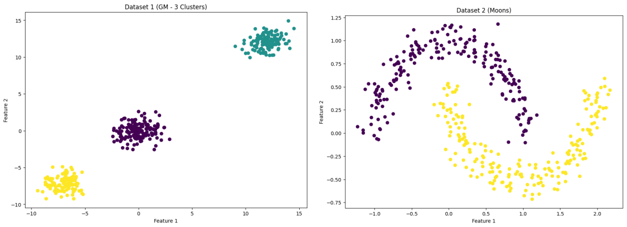

Figure 2: The first image is related to the three-clusters GMM and the second one is related to the two-moons dataset

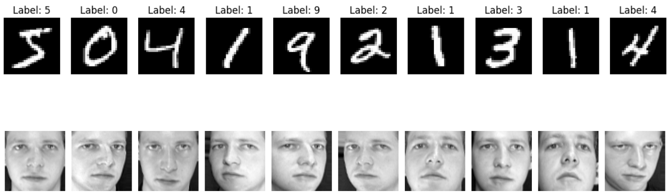

Figure 3: 10 Samples from the MNIST dataset(top figure) and from the Olivetti Faces dataset (bottom figure)

The aim is to ensure a comprehensive evaluation of the K-means algorithm across a range of different data structures and complexities. The methods that we compare are random search, K-means++, KR, KKZ, PCA-part, HAC, PCA-guided search, and continuous K-means.

3.2 Evaluation Metrics for PCA-enhanced K-means:

In the context of applying Principal Component Analysis (PCA) to enhance the performance of the K-means algorithm, it becomes imperative to assess the effectiveness of the combined approach. Several evaluation metrics are instrumental in gauging the success of this integration:

Remark: In our experimental setup, the datasets used for evaluation are generated with known labels (y_{true}). This availability of ground truth labels allows for a thorough evaluation of clustering performance using supervised metrics. In real-world scenarios, obtaining such labeled datasets may be challenging, but in this study, it facilitates the use of accuracy, F1 scores, and other supervised metrics for a comprehensive assessment.

- **Accuracy:** Accuracy remains a fundamental metric, indicating the overall correctness of the clustering results achieved through the combined PCA-K-means approach. It is computed as the ratio of correctly classified instances to the total number of instances:

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{Total Instances}}$$

- **Specificity:** Specificity measures the PCA-K-means model's proficiency in correctly identifying instances that do not belong to a particular class. It is calculated as the ratio of true negatives to the sum of true negatives and false positives:

$$\text{Specificity} = \frac{\text{True Negatives}}{\text{True Negatives} + \text{False Positives}}$$

- **Sensitivity (Recall):** Sensitivity, or recall, assesses the ability of the PCA-K-means model to correctly identify relevant instances. The ratio of true positives to the sum of true positives and false negatives quantifies this metric:

$$\text{Sensitivity (Recall)} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

- **Precision:** Precision measures the accuracy of positive predictions made by the PCA-K-means model. It is calculated as the ratio of true positives to the sum of true positives and false positives:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

- **F-1 Score:** The F-1 score, being the harmonic mean of precision and recall, offers a balanced assessment that considers both false positives and false negatives:

$$\text{F-1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

- **Distortion (Intra-class Variance):** Distortion evaluates the compactness of clusters generated by the PCA-K-means approach. It quantifies the average squared distance between each data point and its assigned cluster centroid, providing insight into cluster compactness:

$$\text{Distortion} = \sum_{k=1}^{K} \sum_{x_i \in C_k} \|x_i - m_k\|^2$$

These metrics collectively serve as valuable tools to comprehensively evaluate the impact of PCA on refining the clustering outcomes of the K-means algorithm.

3.3 Results

Regarding the first dataset (three-cluster GMM), we obtain the following table:

Initialization	Distortion	Accuracy	Specificity	F1 Score	Time	Runs
Random	895.79	1.0	-1.0	1.0	0.00058	100
K-Means++	895.79	1.0	-1.0	1.0	0.00040	100
PCA Guided	895.79	1.0	-1.0	1.0	0.00076	100
KKZ	895.79	1.0	-1.0	1.0	0.00697	1
KR	895.79	1.0	-1.0	1.0	0.00075	1
HAC	895.79	1.0	-1.0	1.0	0.92214	1
PCA Part	895.79	1.0	-1.0	1.0	0.00079	1
Cont. K-Means	895.79	1.0	-1.0	1.0	0.10383	1

Table 1: Performance Metrics - Scenario 1 - 3-cluster GMM

We can see that the accuracy reaches, as expected, the maximal value (i.e 1) for all initialization functions. This result is reassuring as the dataset is constructed by creating three distinct clusters. However, as this dataset is too basic and too small, the performance of each initialization method remain limited and impossible to distinguish. In fact, all the metrics, and the distortion especially, remain equal for all the initialization methods.

The second dataset gives the following table:

Initialization	Distortion	Accuracy	Specificity	F1 Score	Time	Runs
Random	165.40	0.7475	-1.0	0.75	0.00052	100
K-Means++	165.40	0.7475	-1.0	0.75	0.00058	100
PCA Guided	165.40	0.7475	-1.0	0.75	0.00086	100
KKZ	165.43	0.7475	-1.0	0.75	0.00403	1
KR	165.43	0.7475	-1.0	0.75	0.00091	1
HAC	165.40	0.7475	-1.0	0.75	0.91707	1
PCA Part	165.40	0.7475	-1.0	0.75	0.00074	1
Cont. K-Means	165.40	0.7475	-1.0	0.75	0.11258	1

Table 2: Performance Metrics - Scenario 2 - two moons dataset

Regarding the moons dataset, we observe the same trend as GMM 3-Cluster. Regardless of the chosen starting points, the distortion values remained strikingly similar. Additionally, the accuracy of the clustering results also remained unchanged. However, the latter is lower because the data is not linearly separable and thus K-means clustering does not achieve to separate the clusters.

The MINST dataset gives the following table:

Initialization	Distortion	Accuracy	Specificity	F1 Score	Time	Runs
Random	1.165168e+06	0.793545	-1.000039	0.799106	0.083380	100
K-Means++	1.165155e+06	0.794658	-1.000039	0.800150	0.083141	100
PCA Guided	1.165191e+06	0.794101	-1.000040	0.799389	0.065392	100
KKZ	1.167750e+06	0.770173	-0.999972	0.776864	0.255628	1
KR	1.187500e+06	0.713968	-0.999982	0.719605	0.114470	1
HAC	1.167771e+06	0.775181	-0.999972	0.780917	21.452304	1
PCA Part	1.171350e+06	0.705064	-1.000008	0.708457	0.092427	1
Cont. K-Means	1.168937e+06	0.795771	-1.000024	0.795928	1.406949	1

Table 3: Performance Metrics - Scenario 3

In this case, the evolution of the distortion over the runs is relevant, the following figure shows this evolution for all the initialization methods imputed:

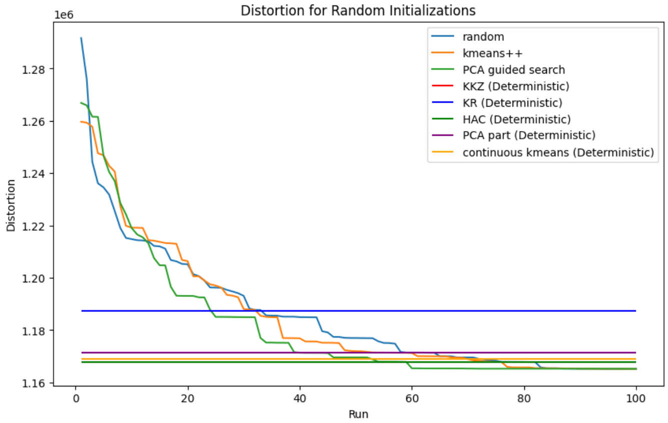

Figure 4: Experimental results on MNIST dataset

For the MNIST dataset, the lowest distorsion is achieved by the three non-deterministic methods (PCA-guided search, random and K-means++). However, the PCA-guided search method needs fewer runs as the other two to achieve its performance. On the other hand, deterministic methods such as HAC and continuous K-means can achieve competitive performance. Their distorsion is not as good as what we can expect with 100 runs of probabilistic methods, but in most runs, their performance is competitive against the probabilitic methods.

3.4 Additional Experiments: EM algorithm

In this section, we extend our exploration to the Expectation-Maximization (EM) algorithm. Applied to Gaussian Mixture Models (GMMs), EM serves as an effective clustering tool. A GMM with K components is defined through the density $p(x|\theta) = \sum_{i=1}^{K} \pi_k \mathcal{N}(x; \mu_k, \Sigma_k)$ where $\theta = (\pi, (\mu, \Sigma))$. This approach assumes that the data points are generated from a mixture of several Gaussian distributions, each representing a cluster. Thus the goal is to determine the parameters θ which maximise the likelihood of the model. In fact, K-means clustering corresponds to the EM algorithm in which we only look for the optimal parameter $\theta = \mu$, since Σ and π are fixed as $\pi_k = \frac{1}{K}$ and $\Sigma_k = I_d$ for all k. Thus, similarly as K-means clustering, initialization plays a crucial role in determining the convergence path and final solution given its sensitivity to starting conditions.

We evaluated the same initialization methods on EM-algorithm, except random initialization is replaced by the following: $\alpha_k = \frac{1}{p}$, $\mu_k \in \mathcal{U}\{x_1, \dots, x_n\}$, and $\Sigma_k = k\sigma^2 I_d, \forall k \in \{1, \dots, p\}$ where $\sigma^2 = \frac{1}{n} \sum_{i=1}^{n} (X_i - \bar{x})^T (X_i - \bar{x})$. We call this the empirical method. For the others, we utilized the identified centroids as the initial means, their empirical covariance as the covariance matrix, and the frequency of points associated with each cluster as their respective weights. For each initialization, we measured the likelihood of θ given the outcomes $\{x_i\}_{i=1}^{n}$, and their execution time in the same way as K-means part. The dataset we employed is a Gaussian mixture with two overlapping clusters (**GM - 2 Clusters**) and a sample size of N = 2000. The parameters are $\mu_1 = (0,0)^T$, $\mu_2 = (1,1)^T$, and $\Sigma_1 = \Sigma_2 = I_2$. We tried other datasets but the algorithm struggled with convergence and achieving good numerical stability on other datasets.

Table 4: EM on GM - 2 Clusters

Initialization	Likelihood	Time (100 iterations)	Run
Empirical	-3038.592177	1.291039	10
K-means++	-3035.864649	1.361744	10
PCA-guided search	-3037.739183	1.473654	10
KKZ	-3037.665874	1.440650	1
KR	-3036.743521	1.217631	1
HAC	-3037.717420	8.021889	1
PCA part	-3037.726458	1.318762	1
Continuous K-means	-3037.702016	2.107370	1

K-means++ initialization emerged as the top performer in term of Likelihood, and KR leads to the lowest execution time.

4 CONCLUSION

As a conclusion, this exploration of the PCA-enhanced K-Means algorithm has provided valuable insights into the nuances of clustering scenarios. The examination of various initialization methods and datasets has shed light on the algorithm's robustness and sensitivity to different inputs.

The implementation of K-Means initialization functions revealed the significance of diverse strategies, such as random initialization, K-Means++, and PCA-based methods, in influencing the convergence and efficacy of the clustering algorithm. The evaluation on three distinct datasets, each posing unique challenges, allowed for a comprehensive understanding of the algorithm's performance under different scenarios. Our datasets were generatd by hand, so it was difficult to see the real influence of each initialization method. However when implementing real datasets such as MNIST, the results were more significant.

Looking forward, there is an exciting opportunity to extend this exploration by testing initialization methods on the Expectation-Maximization (EM) algorithm applied to Gaussian Mixture Models (GMM). This extension holds promise in uncovering insights into the interplay between initialization strategies and probabilistic clustering algorithms, further enriching our understanding of unsupervised learning methodologies.

[0]

REFERENCES

[1] Michael E Wall, Andreas Rechtsteiner, and Luis M Rocha. Singular value decomposition and principal component analysis. In *A practical approach to microarray data analysis*, pages 91–109. Springer, 2003.

[2] Chris Ding and Xiaofeng He. K-means clustering via principal component analysis. In *Proceedings of the twenty-first international conference on Machine learning*, page 29, 2004.

[3] M.E. Celebi, H. Kingravi, and P.A. Vela. A comparative study of efficient initialization methods for the k-means clustering algorithm. *Expert Systems with Applications*, 40(1):200–210, 2013.

[4] D. Arthur and S. Vassilvitskii. k-means++: the advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1027–1035, 2007.

[5] L. Kaufman and P.J. Rousseeuw. *Finding Groups in Data. An Introduction to Cluster Analysis.* Wiley, New York, 1990.

[6] I. Katsavounidis, C.C.J. Kuo, and Z. Zhang. A new initialization technique for generalized lloyd iteration. *IEEE Signal Processing Letters*, 1(4):144–146, 1994.

[7] C. Fraley. Algorithms for model-based gaussian hierarchical clustering. *SIAM Journal on Scientific Computing*, 20(1):270–281, 1998.

[8] M. Meila and D. Heckerman. An experimental comparison of several clustering and initialization methods. In *Proceedings of 4th Conference on Uncertainty in Artificial Intelligence*, pages 386–395, 1998.

[9] S.J. Redmond and C. Heneghan. A method for initialising the k-means clustering algorithm using kd-trees. *Pattern Recognition Letters*, 29(7):965–973, 2007.

[10] T. Su and J.G. Dy. In search of deterministic methods for initializing k-means and gaussian mixture clustering. *Intelligent Data Analysis*, 11(4):319–338, 2007.

[11] Chris Ding and Xiaofeng He. Linearized cluster assignment via spectral ordering. In *Proceedings of the twenty-first international conference on Machine learning*, page 30, 2004.

[12] D. Boley. Principal direction divisive partitioning. *Data Mining and Knowledge Discovery*, 2(4):325–344, 1998.

A K-MEANS ALGORITHM PSEUDOCODE

Algorithm 1 K-means clustering

```
Require: $X$: set of $n$ data points, $K$: number of clusters
Ensure: $C$: set of $K$ clusters
Initialization of $K$ cluster centers $m_1, \dots, m_K$ (see 2.2 section).
repeat
    for $i = 1$ to $n$ do
        $j \leftarrow \arg\min_k ||x_i - y_k||^2$
        Assign $x_i$ to cluster $j$: $C_j$
    end for
    for $j = 1$ to $K$ do
        Update cluster center: $y_m = \frac{1}{|C_j|} \sum_{x_i \in C_j} x_i$ where $C_j$ data set of $j$th cluster
    end for
until Until $m_1, \dots, m_K$ no longer change
Return clusters: $C$=($C_1^*$,. . .,$C_K^*$) and cluster centers: $m^*$=($m_1^*$,. . .,$m_K^*$)
```

B PCA-RELATED FIGURES

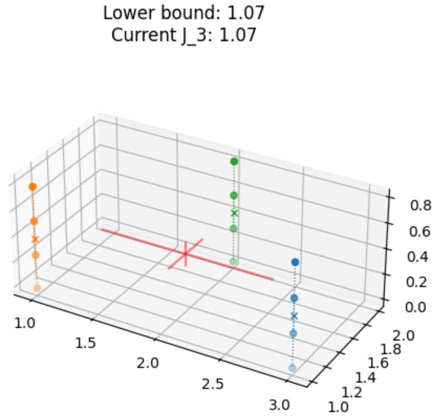

Figure 5: A dataset and a 3-way clustering achieving the lower bound of theorem 2.1. Principal directions are shown in red.

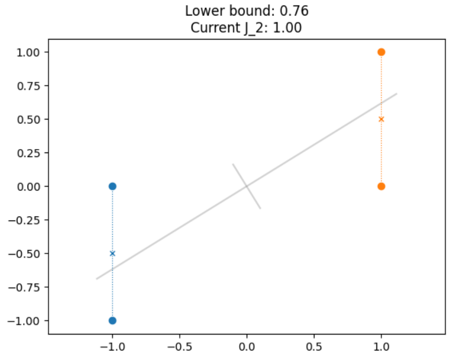

Figure 6: An example where the centroids of an optimal clustering don't belong to the PCA subspace. Here, $\mathrm{cov}X = \frac{1}{4}X^\top X = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 0.5 \end{pmatrix}$**, and the eigenvector associated with its largest eigenvalue is** $v_1 = \begin{pmatrix} 1 + \sqrt{5} \\ 2 \end{pmatrix}$**. None of the centroids of the optimal clustering shown belong to** $\mathbb{R}v_1$**.**

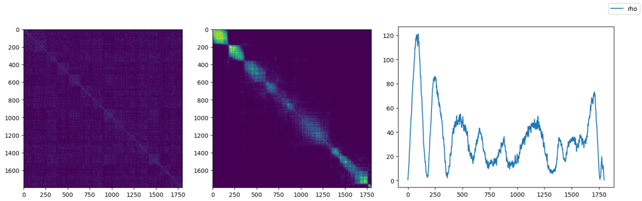

Figure 7: Recovering the clustering associated to the PCA-solved continuous K-means of the handwritten digits dataset. Left: the C matrix. Middle: the C matrix where rows and columns have been reordered according to the *spectral ordering*. Right: Plot of the quantity describing the sum along the i-th antidiagonal. Detecting valleys lets us recover the positions of the blocks and hence the clustering.

C CONTRIBUTION STATEMENTS:

Each member of this group has contributed equally to the project, both in terms of report writing and practical implementation. Given the multitude of methods employed in this project, each team member took responsibility for writing specific functions within each section.

Every group member implemented at least two initialization methods, two metrics, and two datasets. Lastly, we collectively deliberated on the application of the Expectation-Maximization (EM) algorithm to Gaussian Mixture Models (GMM).