A Review of "K-means Clustering via Principal Component Analysis": Insights and Limitations

Oliver JACK, Eva ROBILLARD, Paulo SILVA
Geometric Data Analysis Final Project, MVA 2024-2025

1 INTRODUCTION

1.1 General context

Being able to extract meaningful patterns and structures from large datasets is a necessity in the context of exploratory data analysis. Data clustering, an unsupervised learning technique, is the process of partitioning a dataset into subsets, or "clusters", where data points within the same cluster share similar characteristics, while points in different clusters are quite distinct from one another. Unlike classification, a supervised learning technique, clustering does not rely on pre-labeled data. Data clustering has numerous applications across various fields. In biology, it is used in genomic data analysis, for example by building groups of genes with related expression patterns (also known as coexpressed genes). In marketing, it helps segment customers based on behavior, enabling targeted strategies. In image processing, clustering techniques support object recognition and image segmentation. Traditional clustering algorithms, such as K-means [MacQueen 1967] and hierarchical clustering [Johnson 1967], have laid the foundation for modern data analysis.

The K-means clustering method is a well-known clustering technique used to categorize data into K groups, named clusters. Those clusters are characterized by their centroids, results of the minimization of the sum of squared errors between the data matrix $X = (x_1, \dots, x_n)$ and the centroid of each cluster C_k, referred to as $m_k = \sum_{i \in C_k} x_i / n_k$, with n_k the number of data points in C_k:

$$J_K = \sum_{k=1}^{K} \sum_{i \in C_k} ||x_i - m_k||_2^2. \tag{1}$$

The K-means algorithm is iterative: starting from an initial guess of the solution, it iteratively improves the solution until a local optimal solution is reached. While it is widely used due to its simplicity and efficiency, this clustering method often faces challenges with high-dimensional data. Because its objective functions are non-convex, there can be many different local optimal solutions. A significant challenge lies in the algorithm's sensitivity to the initial selection of centroids, that could lead the solutions to stay trapped in a local minima on account of the greedy nature of their update process [Fränti and Sieranoja 2019]. Starting from different initializations, it often converges to different local optimal solutions. Over the years, several strategies have been explored to overcome its limitations, enhancing the performance and stability of K-means.

Principal Components Analysis (PCA) is a linear dimensionality reduction technique. The data is transformed linearly to a new system of coordinates such that the directions (or principal components) capturing the largest variation in the data are highlighted. Now, let us present its use in practice.

Let X be the original data matrix of dimension $p \times n$, where p is the number of features and n is the number of observations. We define the centered data matrix $Y = (y_1, \dots, y_n)$ with $y_i = x_i - \bar{x}$, where $\bar{x} = \frac{1}{n} \sum_i x_i$. The covariance matrix (ignoring the factor $\frac{1}{n}$) is given by:

$$\sum_i (x_i - \bar{x})(x_i - \bar{x})^T = YY^T. \tag{2}$$

The principal directions u_k and principal components v_k are the eigenvectors satisfying:

$$YY^T u_k = \lambda_k u_k, \quad Y^T Y v_k = \lambda_k v_k, \quad v_k = \frac{Y^T u_k}{\lambda_k^{1/2}}. \tag{3}$$

These are the defining equations for the singular value decomposition (SVD) of $Y = \sum_k \lambda_k^{1/2} u_k v_k^T$. The elements of v_k are the projected values of the data points on the principal direction u_k.

One contribution that addresses the K-means clustering limitations is *'K-means Clustering via Principal Component Analysis'* authored by C. Ding and X. He and published in the *Proceedings of the 21st International Conference on Machine Learning*, Banff, Canada, in 2004.

The objective of our work is to review the paper from Ding and He, studying its context and content by bringing a critical perspective to their developments and results.

1.2 Contribution of the paper...

In this section, we explain how the paper contributes to addressing the previously-mentioned limitations of K-means clustering. The context of the paper is thoroughly developed in Section 2.

This paper by Ding and He makes a groundbreaking theoretical contribution by establishing a formal link between PCA and K-means clustering. The authors demonstrate that principal components serve as continuous solutions to the discrete cluster membership indicators in K-means. They show that the subspace spanned by cluster centroids corresponds to the spectral expansion of the data covariance matrix truncated at $K - 1$ terms. More precisely, their work highlights the connection between those two methods by proving that PCA clusters the data following the K-means objective function. This insight goes beyond the conventional noise-reduction interpretation of PCA, highlighting its intrinsic connection to clustering.

This paper also displays empirical results, using datasets like DNA gene expression and Internet newsgroups, while confirming the effectiveness of these theoretical insights, with clustering results approaching optimal solutions.

1.3 ... and contribution of our review

In this research paper, we propose a review of the work of Ding and He with the objective of understanding more thoroughly their contribution, within the related literature but also in itself. We aim to bring a critical view to some of their decisions and developments,

more particularly by proposing corrections and extensions to their results. Then, another of our developments is held in the computation of their methods, which we make accessible directly to the reader and thus contribute to the reproducibility of their work. Finally, we implement their method, while expanding their analysis to more advanced PCA methods, characterizing the scope of the performance of their approach.

In Section 2, we provide a bibliography analysis to better understand the scientific context of Ding and He's paper. The main contributions of this part are to highlight how this work contributed to the state of the art at its time of publication, but also what places it holds in the K-means clustering bibliography today. After the insight into the positioning of this work, Section 3 provides a complete analysis of the content of the paper, emphasizing the results and techniques developed. Additionnally, Section 4 presents limitations that were found from the developments of the paper, suggesting corrections and extensions to their manuscript. Finally, Section 5 evaluates the performance of the developed method. First, we carry out the same exact experiments as in the original paper, characterizing the reproducibility of the original results for the same datasets. Then, we explore the use of a new dataset for the original method and incorporate advanced PCA techniques, like Sparse PCA and Kernel PCA, to study their impact on clustering performance within the context of the method.

2 CONTEXT OF THE PAPER AND ALTERNATIVE METHODS

The paper we are reviewing was published in 2004. As one would expect, the context of the paper at its time of publishing and at the time of writing of this review are quite different. We are thus splitting this part of our review into two main analysis.

2.1 In what context was this paper published?

At the time of publication of this paper, the main research direction explored to provide better solutions to the K-means algorithm was through better initialization algorithms.

An early approach explored in this direction is the random initialization, a random search technique where centroids were either randomly selected from the dataset or the data points were randomly partitioned into K groups [Anderberg 1973; Forgy 1965]. Some studies suggest that random partitioning was generally more efficient than random centroid selection [Pena et al. 1999]. Another widely adopted practice involved multiple random starts, which consists of running K-means multiple times with different random initializations and selecting the best result based on the objective function [Bradley and Fayyad 1998].

Some authors explored iterative methods to refine initial centroid selection. One method suggests selecting the first centroid as the most centrally located in the set of points, and defining the subsequent ones to be far from the previously located centroids yet close to many data points [Kaufman and Rousseeuw 2009]. Another method suggests starting with the point with the maximum norm as the first centroid and choosing the subsequent centroids based on their maximum distance from those already selected [Katsavounidis et al. 1994]. We could additionally cite iterative refinements that involved clustering random sub-samples to determine centroids [Fayyad et al. 1998] or employing hierarchical agglomerative clustering (HAC) techniques [Fraley 1998; Meilă and Heckerman 1998]. Some heuristics-based methods to refine centroid selection, such as the Hartigan-Wong algorithm, have also been proven effective [Hartigan and Wong 1979].

In addition to these methods, density-based approaches improve centroid initialization by identifying regions with high data density and placing centroids within those areas. This approach ensures better representation of the overall data distribution and reduces the likelihood of poor initial placements [Xu et al. 1996].

The previously described approaches focused on the search for better initialization in the high-dimensional space. However, some researchers also explored leveraging dimensionality-reduced representations to further optimize cluster initialization.

This idea stems from a deterministic divisive hierarchical method for the initial guess of partitions, which recursively bi-partitions a cluster into two by computing the first principal component, and then uses the obtained clusters for K-means initialization. This method is known as PCA-part [Su and Dy 2004] and is conceptually close to the Principal Direction Divisive Partitioning (PDDP) algorithm introduced by Boley [Boley 1998]. Both techniques demonstrated how PCA could guide partitioning processes and paved the way for a new research direction.

A notable contribution explored projecting data into lower dimensional subspaces using PCA before applying K-means clustering in the reduced space. It was showed that this spectral relaxation of the K-means objective led to eigenvector-based solutions, not only notable for noise reduction, but also improving computational efficiency and clustering performance [Zha et al. 2001]. This method's effectiveness was empirically validated even before its theoretical formalization, for example in an experiment-based study on gene expression profiles, in which the method demonstrated significantly improved accuracy [Alizadeh et al. 2000]. Additionally, embedding data into specialized low-dimensional spaces before applying K-means, such as the eigenspace of a graph Laplacian, demonstrated the adaptability and versatility of spectral methods across various data types and objectives [Ng et al. 2001].

Building on these theoretical foundations, Ding and He's paper pioneered a pivotal theoretical analysis about the close relationship between K-means clustering and PCA. Their work demonstrated that principal components act as continuous approximations of cluster membership indicators in K-means clustering, showing that PCA inherently performs clustering in alignment with the K-means objective function. In other words, they showed that the global solution to K-means clustering lies more precisely within the PCA subspace. By applying PCA to reduce the dataset's dimensionality, initial centroids are selected along the principal components, providing a structured starting point that captures the data's variance more effectively. [Ding and He 2004a]

2.2 In what context does this paper exist today?

This paper represents a significant milestone in the work of Ding and He. It constitutes a theoretical result building upon several previous research contributions in spectral graph partitioning. In

their previous contributions, they were not only able to demonstrate the relationship between graph cuts and spectral methods for clustering, but also applied spectral clustering methods to high-dimensional data using PCA [Ding et al. 2002, 2001].

The authors actually presented 2 different papers at the *Twenty-first international conference on Machine Learning*. In addition to the reviewed paper, their second paper focused on reducing multi-way clustering problems to one-dimensional clustering using spectral ordering. Specifically, it introduced a method that uses the spectral ordering of the data points to simplify the clustering process. The paper demonstrates the close relationship between spectral clustering and eigenvectors of the Laplacian matrix, emphasizing its effectiveness in identifying underlying cluster structures, particularly when compared to recursive two-way spectral clustering or traditional K-means [Ding and He 2004b].

Following these publications, Ding and He ventured into the new research direction of hybrid methods. Their later work linked non-negative matrix factorization (NMF) with K-means, providing both theoretical insights and practical applications. They expanded the connection between matrix factorization techniques and spectral clustering, enhancing robust high-dimensional data analysis [Ding et al. 2005, 2008].

Apart from their contributions, other methods for high dimensional space solutions have been developed. One prominent approach is K-Means++, which reduces the randomness in centroid selection by spreading centroids more evenly. This method selects the first centroid randomly and then chooses subsequent centroids based on a probability proportional to the squared distance from existing centroids [Arthur and Vassilvitskii 2007]. More generally, hybrid approaches combine multiple strategies, such as integrating hierarchical clustering or alternative distance metrics before applying K-Means. These methods aim to improve the robustness of centroid initialization, addressing specific challenges posed by different datasets [Celebi et al. 2013].

One of the key advancements posterior to this paper is the development of advanced PCA-based techniques for K-means clustering, including Kernel PCA, Robust PCA, Sparse PCA and deep learning integrations. Kernel PCA extends traditional PCA to non-linear spaces, improving clustering in complex datasets, and has proven efficient in several applications [Xu and Franti 2004; Yu et al. 2011]. Sparse PCA helps handle noisy or incomplete data, maintaining clustering accuracy while reducing dimensionality, as shown in practical applications [Yousefi et al. 2017]. Additionally, deep learning integrations supplement PCA for more efficient dimensionality reduction, particularly in high-dimensional or unstructured data, as seen in applications like brain tumor segmentation in MRI images or topic extraction in NLP [Ragupathy and Karunakaran 2021; Zhang et al. 2018]. These developments provide scalable, effective clustering methods for modern data analysis challenges.

More recently, the authors published a paper with extensive experiments on four real world datasets, systematically comparing their method with previous algorithms. They demonstrated that their proposed PCA-based approach significantly improves the effectiveness of K-means clustering [Xu et al. 2015].

Overall, the topic has evolved from basic K-means enhancements to sophisticated, hybrid approaches combining initialization strategies and advanced PCA methods for high-dimensional data.

3 TOPIC OVERVIEW

3.1 Discrete vs. continuous K-means Clustering Solution

In the traditional K-means clustering problem, observations are assigned to exactly one of the K clusters. A compact way to describe a clustering solution is by defining the indicator matrix $H_K = (\mathbf{h}_1, \ldots, \mathbf{h}_K)$, where

$$\mathbf{h}_k = (0, \ldots, 0, \underbrace{1, \ldots, 1}_{n_k}, 0, \ldots, 0)^T / \sqrt{n_k}, \tag{4}$$

for $k = 1, \ldots, K$. Each of these K indicator vectors corresponds to a specific cluster and takes n binary values (1 for membership, 0 otherwise), representing the n data points. Here, without loss of generality, the data is indexed such that observations belonging to the same cluster are adjacent.

While this method of indicating the data offers a clear overview of the distribution of clusters, in reality, the cutoff between two clusters is not as well-defined as it may be when considering discrete solutions. Data points can sometimes lie closer to one cluster than another, or even show characteristics that place them in both clusters, leading to ambiguous cluster memberships. Furthermore, optimizing over such discrete solutions is often computationally challenging due to the combinatorial nature of the problem. Therefore, the authors propose to simplify the problem by considering continuous solutions instead, allowing indicator vectors to take real values in $[-1, 1]$. This continuous relaxation transforms the discrete clustering problem into a continuous optimization problem, enabling the use of methods from linear algebra and spectral theory. While continuous clustering no longer provides a clear-cut partitioning, it offers the flexibility to implement custom thresholds, allowing us to specify conditions as to when a data point is considered to belong to a particular cluster.

3.2 2-means Clustering

The case $K = 2$ is considered to be the simplest clustering problem (except for the trivial case when $K = 1$). Ding and He manage to come up with a first key result, showing the direct link between PCA and 2-means clustering. First, they define a metric measuring the distance between two clusters:

$$d(C_k, C_l) = \sum_{i \in C_k} \sum_{j \in C_l} (x_i - x_j)^2. \tag{5}$$

After rewriting the objective function

$$J_K = c - \frac{1}{2} J_D, \tag{6}$$

where $c \geq 0$ is a constant and

$$J_D = \frac{n_1 n_2}{n} \left[2 \frac{d(C_1, C_2)}{n_1 n_2} - \frac{d(C_1, C_1)}{n_1^2} - \frac{d(C_2, C_2)}{n_2^2} \right] \geq 0, \tag{7}$$

one can conclude that minimizing J_K is equivalent to maximizing J_D. Based on this, the authors show that the first principal component $\mathbf{v}_1$ is the continuous solution of the cluster indicator vector. In

other words, $\mathbf{v}_1$ maximizes the objective function J_D and the two clusters can be retrieved by considering

$$C_1 = \{i \mid \mathbf{v}_1(i) \leq 0\}, \quad C_2 = \{i \mid \mathbf{v}_1(i) > 0\}. \tag{8}$$

Furthermore, the optimal value of J_K is bounded by

$$n\overline{\mathbf{y}^2} - \lambda_1 < J_{K=2} < n\overline{\mathbf{y}^2}, \tag{9}$$

where $\overline{\mathbf{y}^2} = \sum_i y_i^T y_i / n$.

At first glance, maximizing J_D with respect to a continuous solution might seem confusing since the objective function in its original form solely relies on well-defined, fully-separated clusters. To overcome this issue, the authors suggest to rewrite J_D in matrix form:

$$-J_D = \mathbf{q}^T D \mathbf{q}, \tag{10}$$

where

$$D = (d_{ij})_{ij} = (||\mathbf{x}_i - \mathbf{x}_j||^2)_{ij}, \tag{11}$$

and

$$\mathbf{q}(i) = \begin{cases} \sqrt{\frac{n_2}{n n_1}} & \text{if } i \in C_1 \\ -\sqrt{\frac{n_1}{n n_2}} & \text{if } i \in C_2 \end{cases}. \tag{12}$$

From this, the authors propose to relax the discrete constraints, letting $\mathbf{q}$ take values in $[-1, 1]$ and using the centered distance matrix $\hat{D}$. This transforms the problem into a minimization problem in $\mathbf{q}$ of the form

$$J(\mathbf{q}) = \frac{\mathbf{q}^T \hat{D} \mathbf{q}}{\mathbf{q}^T \mathbf{q}} = -2 \frac{\mathbf{q}^T Y^T Y \mathbf{q}}{\mathbf{q}^T \mathbf{q}}. \tag{13}$$

By properties of the Rayleigh quotient, the continuous solution for the cluster indicator vector is simply the eigenvector $\mathbf{v}_1$ associated to the largest eigenvalue of $Y^T Y$.

3.3 K-means Clustering

After having established a first link between K-means clustering and PCA, the authors move to the more general setting of $K > 2$. The first step consists in rewriting the objective function J_K:

$$J_K = \mathrm{Tr}(X^T X) - \mathrm{Tr}(H_K^T X^T X H_K), \tag{14}$$

where H_K is the previously defined discrete indicator matrix. Due to the inherently redundant nature of H_K (sum of each row is equal to 1), one can perform a linear transformation T on H_K to obtain the matrix Q_K:

$$Q_K = H_K T = (\mathbf{q}_1, \ldots, \mathbf{q}_K), \tag{15}$$

where T is a $K \times K$ orthonormal matrix with a constraint on the last column vector

$$\mathbf{t}_K = (\sqrt{n_1/n}, \ldots, \sqrt{n_K/n})^T. \tag{16}$$

As a result, the last column of the matrix Q_K corresponds to $\mathbf{q}_K = (1/\sqrt{n}, \ldots, 1/\sqrt{n})^T$, while the other $K-1$ columns are the discrete cluster indicator vectors initially. After relaxation of the problem, they represent the continuous solutions that approximate the discrete cluster indicator vectors. Thanks to this transformation, the dimensionality of the problem is reduced from K to $K-1$, without losing any essential clustering information. The intuition behind this reasoning is that by assigning the data points to $K-1$ clusters, the final cluster can be formed by grouping all the remaining unassigned data points. Moreover, the continuous relaxation converts the clustering problem to a spectral optimization problem, making it computationally easier to solve.

By noticing that J_K does not vary depending on whether we consider the original data X or the centered data Y, the authors show that it can reformulated as

$$J_K = \mathrm{Tr}(Y^T Y) - \mathrm{Tr}(Q_{K-1}^T Y^T Y Q_{K-1}), \tag{17}$$

where $Q_K = (\mathbf{q}_1, \ldots, \mathbf{q}_{K-1})$. Thus, the optimization of J_K comes down to:

$$\max_{Q_{K-1}} \mathrm{Tr}(Q_{K-1}^T Y^T Y Q_{K-1}). \tag{18}$$

Before proving a key result related to the continuous solutions and PCA, Ding and He apply **Fan's Theorem** to (18), which reveals that a maximum is reached when considering $Q_{K-1} = (\mathbf{v}_1, \ldots, \mathbf{v}_{K-1})$, where $\mathbf{v}_1, \ldots, \mathbf{v}_{K-1}$ are the $K-1$ first principal components of Y. Since the authors mentioned this theorem without giving a proof, we have provided the full theorem and complete proof in section 4.2.

Finally, from this result one can conclude that the continuous solutions for the cluster membership indicators are given by the first $K-1$ principal components. Furthermore, an updated generalized lower bound can be derived for the objective function J_K:

$$n\overline{\mathbf{y}^2} - \sum_{k=1}^{K-1} \lambda_k < J_K < n\overline{\mathbf{y}^2}. \tag{19}$$

In a next step, the authors use the previous result to come up with a series of propositions. One of these claims that the cluster centroid subspace and the subspace spanned by the first $K-1$ principal components are equivalent, a questionable result which we analyze in further depth in section 4.1. Nevertheless, this result suggests that the optimal cluster centroids can be searched for in a lower-dimensional subspace of the original space, improving the overall efficiency of the optimization problem.

3.4 Practical Challenges & Approximations

While the continuous relaxation of the clustering problem makes it computationally efficient to retrieve solutions that minimize the objective function J_K, an important task lies in retrieving the actual clusters, i.e. the discrete indicator vectors $\mathbf{h}_k$ from the continuous solutions $\mathbf{q}_k$. While the authors manage to show that this optimization problem can be reduced to $K(K-1)/2 - 1$ degrees of freedom, they suggest working with approximations in practice, since finding the exact optimal linear transformation T and hence H_K is computationally hard. However, instead of approximating the indicator matrix H_K, the authors recommend approximating $H_K H_K^T$, thanks to the following property:

$$H_K H_K^T = (H_K T)(H_K T)^T = Q_K Q_K^T, \tag{20}$$

where $\mathbf{q}_1, \ldots, \mathbf{q}_K$ are the discrete valued indicators considered in (15). By replacing the $\mathbf{q}_k$ by their continuous solutions, i.e. the principal components $\mathbf{v}_k$, the following approximation is obtained:

$$C = Q_{K-1} Q_{K-1}^T \approx V_{K-1} V_{K-1}^T = \sum_{k=1}^{K-1} \mathbf{v}_k \mathbf{v}_k^T. \tag{21}$$

First of all, one can observe that the matrix $D = H_K H_K^T$ has a natural diagonal block structure due to the indexation of the data points,

with $d_{ij} > 0$ if i and j belong to the same cluster, 0 otherwise. The authors then define a connectivity probability

$$p_{ij} = \frac{d_{ij}}{\sqrt{d_{ii}d_{jj}}} \approx \frac{c_{ij}}{\sqrt{c_{ii}c_{jj}}}, \tag{22}$$

which can be used to reduce noise and threshold the approximated matrix $C \approx V_{K-1}V_{K-1}^T$ by setting

$$c_{ij} = \begin{cases} c_{ij} & \text{if } p_{ij} \geq \alpha \\ 0 & \text{otherwise} \end{cases}, \tag{23}$$

where $\alpha \in (0, 1)$. Finally, the K clusters can be derived from the updated matrix C by applying a linearized cluster assignment method proposed by the authors themselves [Ding and He 2004b].

4 LIMITATIONS & EXTENSIONS

4.1 Limitations

Despite being able to establish key links between two popular unsupervised learning techniques, there seem to be minor limitations to the work of Ding and He which we wish to highlight.

First of all, the paper suggests that the last column of the $K \times K$ orthonormal matrix T, which is used to linearly transform redundancies in H_K to Q_K, should be of the form

$$\mathbf{t}_n = \left(\sqrt{n_1/n}, \ldots, \sqrt{n_K/n}\right). \tag{24}$$

However, they then note this matrix T incorrectly in the case when $K = 2$, which should actually be of the form

$$T = \begin{pmatrix} \sqrt{n_2/n} & \sqrt{n_1/n} \\ -\sqrt{n_1/n} & \sqrt{n_2/n} \end{pmatrix}. \tag{25}$$

Furthermore, a key result in the paper, **Theorem 3.3**, can cause for confusion amongst readers. The theorem states:

Theorem. *Cluster centroid subspace is spanned by the first $K-1$ principal directions, i.e. $S_b = \sum_{k=1}^{K-1} \lambda_k \boldsymbol{u}_k \boldsymbol{u}_k^T$.*

Let us note that the $\mathbf{u}_k$ are the first $K-1$ principal directions of Y, the eigenvectors corresponding to the $K-1$ largest eigenvalues of YY^T. While this theorem is shown to be true for the continuous solutions of K-means, it is not valid in the general discrete setting. This is largely due to the use of **Theorem 3.1** in the proof of **Theorem 3.3**, which relies on the fact that the continuous membership solutions for the transformed discrete K-means clustering problem are the first $K-1$ principal components $\mathbf{v}_k$. Thus, when considering the discrete clustering solution, without the continuous relaxation on the indicator vectors, it is fairly straightforward to come up with a simple counterexample where the cluster centroid subspace is not spanned by the $K-1$ first principal directions.

Consider the following counterexample:

- **Cluster A:** (2,-1); (0,7),
- **Cluster B:** (-2,-1); (0,-5).

This data is already centered and has the following covariance matrix:

$$YY^T = \begin{pmatrix} 2 & 0 & -2 & 0 \\ -1 & 7 & -1 & -5 \end{pmatrix} \begin{pmatrix} 2 & 0 & -2 & 0 \\ -1 & 7 & -1 & -5 \end{pmatrix}^T = \begin{pmatrix} 8 & 0 \\ 0 & 76 \end{pmatrix}.$$

The eigenvalues of this matrix are $\lambda_1 = 8$, $\lambda_2 = 76$, with the associated eigenvectors being $\mathbf{v}_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, $\mathbf{v}_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$. Since $K = 2$, **Theorem 3.3** claims that the cluster centroid subspace will be spanned by the $K - 1 = 1$ first principal components, i.e. $\mathbf{v}_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$. The cluster centroids are given by:

$$\text{Centroid}_A = \left(\frac{2+0}{2}, \frac{-1+7}{2}\right)^T = \begin{pmatrix} 1 \\ 3 \end{pmatrix},$$

$$\text{Centroid}_B = \left(\frac{-2+0}{2}, \frac{-1-5}{2}\right)^T = \begin{pmatrix} -1 \\ -3 \end{pmatrix}.$$

Hence, the cluster centroid subspace is spanned by the vector $\begin{pmatrix} 1 \\ 3 \end{pmatrix}$. From this result, we can conclude that the two subspaces are not equal to one another, showing that the PCA subspace does not align with the cluster centroid subspace in general.

Finally, the authors do not provide any type of code related to their computational experiments, nor do they delve into all the specific details of their proposed algorithm, hindering reproducibility and making it difficult for fellow researchers to replicate and verify any numerical results in the paper. This omission could potentially raise concerns about the accuracy and validity of the reported outcomes, as there is no way to confirm implementation details or methodological choices that could impact the findings. Furthermore, the absence of code limits the adoption of the proposed methods, preventing others from using the work as a baseline for comparison or extending it to new applications.

This lack of precision and transparency show that the proposed research paper could benefit from further revision to improve its overall accuracy and shift the attention to its key findings.

4.2 Extensions

In this section, we will discuss further extensions to the paper by providing our own complete proof of a key theorem, as well as concisely outlining the algorithm used to retrieve the K clusters in practice.

By rewriting the objective function J_K, the authors manage to transform the optimization problem to (18). This problem can then be solved directly by applying **Fan's Theorem**:

Theorem (Fan). *Let A be a $n \times n$ symmetric matrix with eigenvalues $\lambda_1 \geq \cdots \geq \lambda_n$ and corresponding eigenvectors $(\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n)$. The maximization of $Tr(Q^TAQ)$ subject to constraints $Q^TQ = I_K$ has the solution $Q = (\boldsymbol{v}_1, \ldots, \boldsymbol{v}_K)R$, where R is an unknown $K \times K$ orthonormal matrix and $\max_Q Tr(Q^TAQ) = \sum_{i=1}^K \lambda_i$.*

Proof. Since the matrix A is symmetric, we know that its eigenvectors $\mathbf{v}_k$ form an orthonormal basis of $\mathbb{R}^n$. This means that any vector in $\mathbb{R}^n$, including the column vectors of Q, can be written as a linear combination of these eigenvectors. Thus, Q can be expressed as

$$Q = VR = (\mathbf{v}_1, \ldots, \mathbf{v}_K)R,$$

where R is a $K \times K$ matrix of coefficients specifying the representation of Q in the eigenvector basis.

The constraint

$$I_K = Q^TQ = (VR)^T(VR) = R^TV^TVR = R^TR,$$

implies that R must be orthonormal.

Next, let us consider the eigenvalue decomposition of $A = V\Lambda V^T$ with $\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_n)$ and substitute $Q = VR$ into $\text{Tr}(Q^T AQ)$:

$$\text{Tr}(Q^T AQ) = \text{Tr}(R^T V^T AVR) = \text{Tr}(R^T \Lambda R).$$

By using the cyclic property of the trace, we then get

$$\text{Tr}(R^T \Lambda R) = \text{Tr}(\Lambda R R^T) = \text{Tr}(\Lambda) = \sum_{i=1}^{K} \lambda_i,$$

which completes the proof. □

Next, let us list the exact steps that can be taken to retrieve the actual K clusters. It is noteworthy that the theoretical foundations of the procedure are inspired by [Ding and He 2004b], while we mainly focus on the key implementation steps.

Algorithm:

(1) Specify the number of clusters K you wish to consider & define the bandwidth $m = \lceil n/K \rceil$ (average size of a cluster);
(2) Compute the connectivity matrix C as defined in (21) & update it to $\tilde{C}$ using the threshold approach given in (23);
(3) Compute the graph Laplacian $L = D - \tilde{C}$ where $D = \text{diag}(d_1, \ldots, d_n)$ is the degree matrix st. $d_i = \sum_j \tilde{c}_{ij}$;
(4) Compute the eigenvector $\tilde{\mathbf{w}}$ associated to the second smallest eigenvalue of L (commonly known as the Fiedler vector), capturing the optimal 1D ordering of the data;
(5) Let $\pi = \text{argsort}(\tilde{\mathbf{w}})$ be the permutation that sorts $\tilde{\mathbf{w}}$ in ascending order;
(6) For each index i, compute the Cluster Crossing

$$\rho(i) = \frac{m}{t} \sum_{j=1}^{t} c_{\pi(i-j),\pi(i+j)}$$

with $t = min(i, n-i, m)$, measuring the strength of connections across the neighborhood of a data point i in the sorted order;
(7) Smooth $\tilde{\rho}(i) = \rho(i+1/2)/4 + \rho(i)/2 + \rho(i-1/2)/4$ to reduce noise & ensure smoother cluster boundaries, where $\rho(i \pm 1/2) = \frac{m}{t} \sum_{j=1}^{m} c_{\pi(i-j),\pi(i+j\pm1)}$ with $t = min(i, n-i, m)$;
(8) Identify valley points, i.e. local minima in $\tilde{\rho}(i)$: $\tilde{\rho}(i) < \tilde{\rho}(i-1)$ & $\tilde{\rho}(i) < \tilde{\rho}(i+1)$. Each region between two valleys belongs to a composite cluster. Due to the potential noise of $\tilde{\rho}$, it is important to adopt a robust method for this step (see Figure 1 and 2 for two examples of $\tilde{\rho}$ in practice);
(9) If the number of composite clusters is equal to K, assign data points to these clusters and stop;
(10) If the number of composite clusters is less than K:
  (a) Identify the largest composite cluster;
  (b) Recursively apply the algorithm to split this cluster further;
(11) Repeat the process until exactly K clusters are formed.

We implemented the full algorithm in Python and it is worth emphasizing that our implemented code (available on GitHub[1]) is the first open-source version of this algorithm to be made public.

[1]https://github.com/olijacklu/MVA/tree/main/Geometric%20Data%20Analysis/Project

5 EXPERIMENTS & NUMERICAL EVALUATION

To evaluate the performance of the proposed algorithm, we replicated the experiments conducted in the original paper, using the same datasets and methods. Additionally, we extended these experiments by testing the algorithm on a new dataset and incorporating advanced PCA techniques, like Sparse PCA and Kernel PCA, to assess their impact on clustering performance.

5.1 Datasets

We replicated the original experiments on the Newspaper dataset[2] using balanced combinations (A2, B2, A5, B5) where, for each combination, we conducted 10 independent runs randomly sampling 100 papers from each newspaper agency. The documents underwent preprocessing to retain only words, followed by vectorization using *Scikit-learn*'s TF-IDF technique. This method generated vectors in which each element corresponds to the term frequency (TF) weighted by the inverse document frequency (IDF) of the respective term, while its label is simply the respective newspaper agency. The vectors were restricted to 1 000 features, representing the 1 000 most frequent terms across the entire corpus. Lastly, we ensured that each vector was normalized such that the sum of the squares of its elements equals 1. Furthermore, to assess the robustness of the method on different data types, we tested it on a high-dimensional gene expression cancer RNA-Seq dataset[3] with 801 observations and 20 531 features, containing expression levels for cancer samples. Overall, the dataset contains 5 labeled clusters, each of which represents a tumor subtype. The results from the PCA-based clustering were compared to the standard K-means implementation from the *Scikit-learn* library.

5.2 Incorporating Sparse PCA

Sparse PCA extends the traditional PCA technique by introducing sparsity constraints, thus being especially effective for high-dimensional datasets in which only a subset of features is relevant. With X being the data matrix with n samples and p variables, the Sparse PCA technique seeks the principal components that maximize the variance, while ensuring the restriction of the number of non-zero elements in the projection vector $\mathbf{v}$. Given an integer k with $1 \leq k \leq p$, the Sparse PCA problem can be formulated as:

$$\begin{aligned} \max \quad & \mathbf{v}^T Y \mathbf{v} \\ s.t. \quad & ||\mathbf{v}||_2 = 1 \\ & ||\mathbf{v}||_0 \leq k. \end{aligned}$$

In this implementation, the computation of the top $K-1$ principal components was modified to replace the eigenvalue decomposition of the covariance matrix with a sparse principal component extraction process.

5.3 Incorporating Kernel PCA

Kernel PCA extends traditional PCA by mapping data non-linearly into a higher-dimensional feature space using a kernel function $K(x_i, x_j) = \phi(x_i)^\top \phi(x_j)$. This transformation allows for better

[2]https://www.cs.cmu.edu/afs/cs/project/theo-11/www/naive-bayes/20_newsgroup/
[3]https://archive.ics.uci.edu/dataset/401/gene+expression+cancer+rna+seq

handling of non-linear structures in the data, and follows these steps:

(1) Compute the kernel matrix: $K_{ij} = \phi(x_i)^\top \phi(x_j)$
(2) Center the kernel matrix: $K_c = K - \mathbf{1}_n K - K\mathbf{1}_n + \mathbf{1}_n K \mathbf{1}_n$
(3) Compute the eigenvectors $\mathbf{v}_1, \ldots, \mathbf{v}_n$ of K_c
(4) Project the data onto the kernel principal components as:

$$y_k(x) = \sum_{i=1}^{n} v_{ki} K(x_i, x),$$

where v_{ki} represents the i-th component of the k-th eigenvector $\mathbf{v}_k$, and $K(x_i, x)$ is the kernel function.

The method's flexibility lies in the choice of the kernel function, with options such as the radial basis function (RBF), polynomial, or sigmoid kernels. These kernels allow customization of the transformation to align with the data's underlying structure. For our experiments, we primarily used the RBF kernel $K(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$ with a parameter $\sigma^2 = 5$. Kernel PCA replaces the eigenvalue decomposition of the covariance matrix with a decomposition of the kernel matrix, enabling the extraction of non-linear principal components.

5.4 Numerical Evaluation

Our experiments highlight the strengths and limitations of the PCA-based clustering approach as proposed in the original paper. For the Newspaper dataset, the standard PCA implementation performed well, achieving accuracies comparable to or even surpassing those of the *Scikit-learn* K-means implementation. Configurations with $K = 2$ generally demonstrated stronger performance compared to $K = 5$, aligning with the findings of the original paper. Furthermore, the error between the theoretical lower bound and the objective function J_K was relatively small in general, as can be seen in Table 1. Overall, this experiment confirmed the effectiveness of the proposed method in providing a structured reduction for this dataset.

	1	2	3	4	5	6	7	8	9	10	Avg. error
Datasets: A2											
Km	160.76	157.04	159.20	162.99	161.51	161.42	159.53	159.09	164.04	163.02	–
P2	158.47	154.82	156.72	160.93	159.71	159.43	157.50	156.62	162.45	160.94	1.31%
Datasets: B2											
Km	146.14	144.40	147.18	149.00	147.46	143.36	144.30	145.17	148.84	151.08	–
P2	143.53	141.22	144.64	146.46	144.61	140.45	141.13	141.97	146.20	148.73	1.91%
Datasets: A5											
Km	391.05	391.16	393.23	394.41	392.49	387.70	390.92	399.26	396.41	387.95	–
P5	369.07	372.06	368.66	376.39	369.29	367.34	366.88	375.63	374.94	363.13	4.94%
Datasets: B5											
Km	394.32	396.26	392.89	393.04	395.94	388.55	395.83	397.03	395.05	394.10	–
P5	374.98	373.02	373.89	374.77	373.74	367.89	374.51	379.21	366.71	376.82	5.26%

Table 1: Values of objective function J_K & theoretical lower bounds.

For the gene expression dataset, the method maintained a solid performance, achieving an accuracy of 82.65%. However, this was slightly lower than the 99.50% achieved by the standard K-means algorithm. The high-dimensional nature of this dataset (over 20 000 features) may present a challenge for PCA-based methods, as the full variance structure may not be adequately captured in the reduced dimensionality subspace. Notably, the datasets used in the original paper were of much lower dimensionality: the authors reduced their gene dataset from 4 029 to 200 features using an F-statistic-based selection prior to clustering, and the Newspaper dataset consisted of only 1 000 features. This contrast emphasizes the need for careful preprocessing and dimensionality reduction strategies when applying PCA-based methods to modern, high-dimensional datasets.

The Sparse PCA implementation offered a slightly lower performance to the standard version on the Newspaper dataset. This suggests that the sparsity constraint might reduce the ability of the components to fully capture the variance structure critical for clustering. Furthermore, while Sparse PCA offers an advantage by generating interpretable components and focusing on a subset of relevant features, its computational cost made it impractical for the gene dataset due to its exceptionally high dimensionality. This highlights a limitation of Sparse PCA for datasets with tens of thousands of features, where enforcing sparsity comes with a steep trade-off in computational efficiency. However, the method remains a valuable tool for cases where interpretability and feature reduction are prioritized over absolute clustering performance.

The Kernel PCA, on the other hand, although theoretically having great potential for capturing non-linear relationships, resulted in disappointing results. For the Newspaper dataset, it performed comparably to standard PCA and did not offer a clear advantage. For the gene dataset, however, its accuracy dropped dramatically to under 30%. This decline could be attributed to multiple factors, including the lack of kernel fine-tuning or hyperparameter optimization, as well as challenges posed by the dataset itself, such as overlapping clusters or the curse of dimensionality. These results suggest that while Kernel PCA holds potential, its success is highly sensitive to kernel selection and hyperparameter calibration, which were not exhaustively explored in this study.

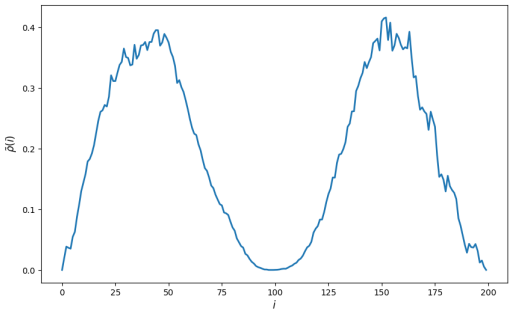

Figure 1: Smoothed Cluster Crossing $\tilde{\rho}(i)$ in the Newspaper B2 setting. The single valley can be identified relatively clearly.

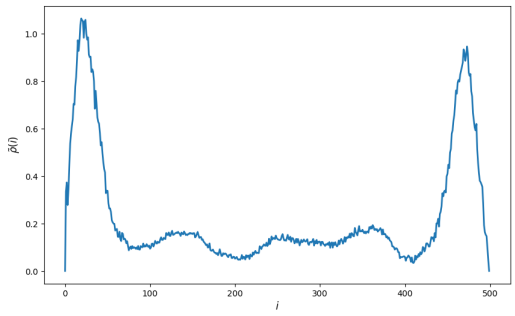

Figure 2: Smoothed Cluster Crossing $\tilde{\rho}(i)$ in the B5 Newspaper setting. The identification of the 4 valleys becomes slightly more ambiguous.

6 CONCLUSION

The paper "K-means Clustering via Principal Component Analysis" by Ding and He was a pioneering work for its time, bridging the gap between two fundamental learning techniques. They were able to show that the principal components can act as continuous approximations to the K-means cluster indicators. This laid the groundwork for further advancements in clustering methods, such as their work [Ding et al. 2005] and [Xu et al. 2014].

While the theorems they presented were both innovative and impactful, the paper also faced notable limitations, particularly in terms of precision, transparency, and reproducibility. Key aspects of their proposed algorithm were not fully detailed (e.g. valley detection), and the lack of accessible resources, such as code, may have prevented broader adoption.

To address these gaps, we extended the original work by providing a complete proof of an important theorem, outlining the full algorithm, and making the first open-source implementation in Python available on GitHub. This implementation enabled testing on both the original Newspaper dataset and a new high-dimensional RNA-Seq dataset. Furthermore, we explored advanced PCA variations, including Sparse PCA and Kernel PCA, to assess their impact on the clustering performance.

Our experiments produced mixed results. The method performed well under specific conditions, such as when the number of features were low, but struggled with high-dimensional datasets. Moreover, the alternative PCA methods could be further tested, highlighting areas for potential improvement in the future.

Overall, while the original paper proposed a first link between clustering and dimensionality reduction, its impact could be increased through more detailed explanations and accessible resources. Our extensions aim to make this influential work more applicable and reproducible, setting the stage for future investigations into clustering methods for complex, high-dimensional data.

REFERENCES

Alizadeh, A. A., Eisen, M. B., Davis, R. E., Ma, C., Lossos, I. S., Rosenwald, A., Boldrick, J. C., Sabet, H., Tran, T., Yu, X., et al. (2000). Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling. *Nature*, 403(6769):503–511.

Anderberg, M. R. (1973). *Cluster analysis for applications*, volume 19. Academic press.

Arthur, D. and Vassilvitskii, S. (2007). k-means++: The advantages of careful seeding. *Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1027–1035.

Boley, D. (1998). Principal direction divisive partitioning. *Data mining and knowledge discovery*, 2:325–344.

Bradley, P. S. and Fayyad, U. M. (1998). Refining initial points for k-means clustering. In *ICML*, volume 98, pages 91–99. Citeseer.

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

Ding, C. and He, X. (2004a). K-means clustering via principal component analysis. In *Proceedings of the twenty-first international conference on Machine learning*, page 29.

Ding, C. and He, X. (2004b). Linearized cluster assignment via spectral ordering. *Proceedings of the twenty-first international conference on Machine learning*.

Ding, C., He, X., and Simon, H. D. (2005). On the equivalence of nonnegative matrix factorization and spectral clustering. In *Proceedings of the 2005 SIAM international conference on data mining*, pages 606–610. SIAM.

Ding, C., He, X., Zha, H., and Simon, H. (2002). Unsupervised learning: self-aggregation in scaled principal component space. In *European Conference on Principles of Data Mining and Knowledge Discovery*, pages 112–124. Springer.

Ding, C., Li, T., and Peng, W. (2008). On the equivalence between non-negative matrix factorization and probabilistic latent semantic indexing. *Computational Statistics & Data Analysis*, 52(8):3913–3927.

Ding, C. H., He, X., Zha, H., Gu, M., and Simon, H. D. (2001). A min-max cut algorithm for graph partitioning and data clustering. In *Proceedings 2001 IEEE international conference on data mining*, pages 107–114. IEEE.

Fayyad, U. M., Reina, C., and Bradley, P. S. (1998). Initialization of iterative refinement clustering algorithms. In *KDD*, pages 194–198.

Forgy, E. W. (1965). Cluster analysis of multivariate data: efficiency versus interpretability of classifications. *biometrics*, 21:768–769.

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

Fränti, P. and Sieranoja, S. (2019). How much can k-means be improved by using better initialization and repeats? *Pattern Recognition*, 93:95–112.

Hartigan, J. A. and Wong, M. A. (1979). Algorithm as 136: A k-means clustering algorithm. *Journal of the royal statistical society. series c (applied statistics)*, 28(1):100–108.

Johnson, S. C. (1967). Hierarchical clustering schemes. *Psychometrika*, 32(3):241–254.

Katsavounidis, I., Kuo, C.-C. J., and Zhang, Z. (1994). A new initialization technique for generalized lloyd iteration. *IEEE Signal processing letters*, 1(10):144–146.

Kaufman, L. and Rousseeuw, P. J. (2009). *Finding groups in data: an introduction to cluster analysis*. John Wiley & Sons.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability/University of California Press*.

Meilă, M. and Heckerman, D. (1998). An experimental comparison of several clustering and initialization methods. In *Proceedings of the fourteenth conference on uncertainty in artificial intelligence*, pages 386–395.

Ng, A., Jordan, M., and Weiss, Y. (2001). On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 14.

Pena, J. M., Lozano, J. A., and Larranaga, P. (1999). An empirical comparison of four initialization methods for the k-means algorithm. *Pattern recognition letters*, 20(10):1027–1040.

Ragupathy, B. and Karunakaran, M. (2021). A deep learning model integrating convolution neural network and multiple kernel k means clustering for segmenting brain tumor in magnetic resonance images. *International Journal of Imaging Systems and Technology*, 31(1):118–127.

Su, T. and Dy, J. (2004). A deterministic method for initializing k-means clustering. In *16th IEEE international conference on tools with artificial intelligence*, pages 784–786. IEEE.

Xu, M. and Franti, P. (2004). A heuristic k-means clustering algorithm by kernel pca. In *2004 International Conference on Image Processing, 2004. ICIP'04.*, volume 5, pages 3503–3506. IEEE.

Xu, Q., Ding, C., Liu, J., and Bin, L. (2014). Pca-guided search for k-means. *Pattern Recognition Letters*, 54.

Xu, Q., Ding, C., Liu, J., and Luo, B. (2015). Pca-guided search for k-means. *Pattern Recognition Letters*, 54:50–55.

Xu, X., Ester, M., Kriegel, H.-P., and Sander, J. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*, pages 226–231.

Yousefi, B., Sharifipour, H. M., Castanedo, C. I., and Maldague, X. P. (2017). Automatic irndt inspection applying sparse pca-based clustering. In *2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE)*, pages 1–4. IEEE.

Yu, S., Tranchevent, L., Liu, X., Glanzel, W., Suykens, J. A., De Moor, B., and Moreau, Y. (2011). Optimized data fusion for kernel k-means clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(5):1031–1039.

Zha, H., He, X., Ding, C., Gu, M., and Simon, H. (2001). Spectral relaxation for k-means clustering. *Advances in neural information processing systems*, 14.

Zhang, Y., Lu, J., Liu, F., Liu, Q., Porter, A., Chen, H., and Zhang, G. (2018). Does deep learning help topic extraction? a kernel k-means clustering method with word embedding. *Journal of Informetrics*, 12(4):1099–1117.