École Normale Supérieure Paris-Saclay

Master 2 - Mathématiques, Vision, Apprentissage

Introduction to Probabilistic Graphical Models and Deep Generative Models

Final Project Report

K-means Clustering via Principal Component Analysis: Limitations and Use in Practice

JACK Oliver
ROBILLARD Eva
2024/2025

-December 17th 2024-

école
normale
supérieure
paris–saclay

MATHÉMATIQUES
VISION
APPRENTISSAGE

1 Introduction

The paper *"K-means Clustering via Principal Component Analysis"* authored by C. Ding and X. He and published the *Proceedings of the 21st International Conference on Machine Learning*, Banff, Canada, in 2004 [Ding and He, 2004a], 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 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.

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.

Contributions This project comes from joint analysis and discussion on the paper (Section 2 and 3). Eva then focused on the limitations on the cluster centroid subspace (Section 4.1) and Oliver on the PCA clustering algorithm (Section 4.2). Implementation-wise, Eva focused on the global structure of the code and experiments, while Oliver implemented the Kernel and Sparse PCA, as well representing the results graphically and analyzing them.

2 Link between PCA and K-means Clustering

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 $X = (x_1, \ldots, x_n)$ of dimension $p \times n$, where p is the number of features and n is the number of observations, 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:

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

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. We define the centered data matrix $Y = (y_1, \ldots, y_n)$ 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:

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

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.

Since the cutoff between two clusters is not always well-defined in reality and since the combinatorial nature of the problem poses computational challenges, the authors propose to simplify the problem by considering continuous solutions instead, allowing indicator vectors to take real values in $[-1, 1]$.

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:

$$Q_K = H_K T = (\mathbf{q}_1, \ldots, \mathbf{q}_K), \tag{5}$$

where T is a $K \times K$ orthonormal matrix with a constraint on the last column vector $\mathbf{t}_K = (\sqrt{n_1/n}, \ldots, \sqrt{n_K/n})^T$. 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 $K - 1$. 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.

The authors then show that the minimization of J_K comes down to solving:

$$\max_{Q_{K-1}} \operatorname{Tr}(Q_{K-1}^T Y^T Y Q_{K-1}). \tag{6}$$

Finally, using **Fan's Theorem** (complete theorem and proof in section 4.2), one can conclude that the continuous solutions for the cluster membership indicators are given by the first $K - 1$ principal components. Furthermore, a lower bound on the objective function J_K can be derived:

$$n\overline{\mathbf{y}^2} - \sum_{k=1}^{K-1} \lambda_k < J_K < n\overline{\mathbf{y}^2}. \tag{7}$$

3 Approximations & Cluster Retrieval

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:

$$H_K H_K^T = (H_K T)(H_K T)^T = Q_K Q_K^T, \tag{8}$$

where $\mathbf{q}_1, \ldots, \mathbf{q}_K$ are the discrete valued indicators considered in (5). By replacing the $\mathbf{q}_k$ by their continuous solutions, i.e. the principal components $\mathbf{v}_k$, the following approximation is obtained:

$$C = Q_{K-1} Q_{K-1}^T \approx V_{K-1} V_{K-1}^T = \sum_{k=1}^{K-1} \mathbf{v}_k \mathbf{v}_k^T. \tag{9}$$

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

$$p_{ij} = \frac{d_{ij}}{\sqrt{d_{ii} d_{jj}}} \approx \frac{c_{ij}}{\sqrt{c_{ii} c_{jj}}}, \tag{10}$$

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

$$c_{ij} = \begin{cases} c_{ij} & \text{if } p_{ij} \geq \alpha \\ 0 & \text{otherwise} \end{cases}, \tag{11}$$

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], detailed in section 4.2.

4 Limitations & Extensions

4.1 Limitations on the Cluster Centroid Subspace & Reproducibility

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.

On the one hand, 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.* $S_b = \sum_{k=1}^{K-1} \lambda_k \mathbf{u}_k \mathbf{u}_k^T$.

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 $\mathbf{v}_1 = (1,0)^T$, $\mathbf{v}_2 = (0,1)^T$. 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. $\mathbf{v}_2 = (0,1)^T$. The cluster centroids are given by:

$$\text{Centroid}_A = \left(\frac{2+0}{2}, \frac{-1+7}{2}\right)^T = \begin{pmatrix} 1 \\ 3 \end{pmatrix}, \quad \text{Centroid}_B = \left(\frac{-2+0}{2}, \frac{-1-5}{2}\right)^T = \begin{pmatrix} -1 \\ -3 \end{pmatrix}.$$

Hence, the cluster centroid subspace is spanned by the vector $(1,3)^T$. 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.

On the other hand, 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 Extended Proof & PCA Clustering Algorithm

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 a potential 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 (6). 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 $(\mathbf{v}_1, \ldots, \mathbf{v}_n)$. The maximization of $Tr(Q^T AQ)$ subject to constraints $Q^T Q = I_K$ has the solution $Q = (\mathbf{v}_1, \ldots, \mathbf{v}_K)R$, where R is an unknown $K \times K$ orthonormal matrix and* $\max_Q \; Tr(Q^T AQ) = \sum_{i=1}^{K} \lambda_i$.

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

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

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

The constraint

$$I_K = Q^T Q = (VR)^T (VR) = R^T V^T V R = R^T R,$$

implies that R must be orthonormal.

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

$$\operatorname{Tr}(Q^T A Q) = \operatorname{Tr}(R^T V^T A V R) = \operatorname{Tr}(R^T \Lambda R).$$

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

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

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.

PCA Clustering Algorithm

```
Input: Number of clusters $K$, data size $n$.
Specify bandwidth $m = \lceil n/K \rceil$ (average size of a cluster).
Compute the connectivity matrix $C$ as defined in (9).
Update $\tilde{C}$ using the threshold approach given in (11).
Compute the graph Laplacian $L = D - \tilde{C}$, where $D = \operatorname{diag}(d_1, \dots, d_n)$ is the degree matrix such that $d_i = \sum_j \tilde{c}_{ij}$.
Compute the eigenvector $\tilde{\mathbf{w}}$ associated with the second smallest eigenvalue of $L$ (Fiedler vector), capturing the
  optimal 1D ordering of the data.
Let $\pi = \operatorname{argsort}(\tilde{\mathbf{w}})$ be the permutation that sorts $\tilde{\mathbf{w}}$ in ascending order.
for each index $i = 1, \dots, n$ do
    Compute the Cluster Crossing: $\rho(i) = \frac{m}{t} \sum_{j=1}^{t} c_{\pi(i-j),\pi(i+j)}$, with $t = \min(i, n-i, m)$.
end for
for each index $i = 1, \dots, n$ do
    Smooth $\tilde{\rho}(i) = \frac{\rho(i+1/2)}{4} + \frac{\rho(i)}{2} + \frac{\rho(i-1/2)}{4}$ to reduce noise, where $\rho(i \pm 1/2) = \frac{m}{t} \sum_{j=1}^{m} c_{\pi(i-j),\pi(i+j\pm1)}$, with
    $t = \min(i, n-i, m)$.
end for
Identify the valley points as local minima in $\tilde{\rho}(i)$: $\tilde{\rho}(i) < \tilde{\rho}(i-1)$ and $\tilde{\rho}(i) < \tilde{\rho}(i+1)$. 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. We proposed working with increasingly expanding local neighborhoods until exactly $K-1$ valleys
  have been detected. This non-parametric approach relies on the clusters being of similar size (see Figure 1 and
  2 for two examples of $\tilde{\rho}$ in practice).
Output: Final $K$ clusters, represented as sets of indices corresponding to the data points in each cluster.
```

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/Introduction%20to%20Probabilistic%20Graphical%20Models/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 then 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. Given an integer k with $1 \leq k \leq p$, the Sparse PCA problem can be formulated as:

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

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 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. 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 $\sigma^2 = 5$.

[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

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 (see appendix). Overall, this experiment confirmed the effectiveness of the proposed method in providing a structured reduction for this dataset.

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.

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, 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

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

[Ding et al., 2005] 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.

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

A Tables & Figures

Datasets: A2											
	1	2	3	4	5	6	7	8	9	10	Avg. error
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											
	1	2	3	4	5	6	7	8	9	10	Avg. error
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											
	1	2	3	4	5	6	7	8	9	10	Avg. error
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											
	1	2	3	4	5	6	7	8	9	10	Avg. error
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.

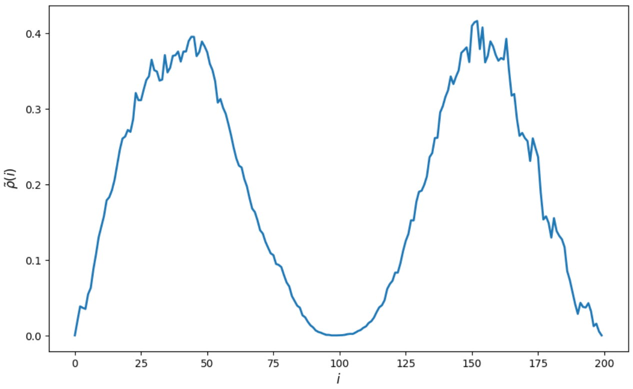

Figure 1: Smoothed Cluster Crossing $\tilde{\rho}(i)$ in the Newspaper B2 setting. The single valley can be identified relatively clearly.

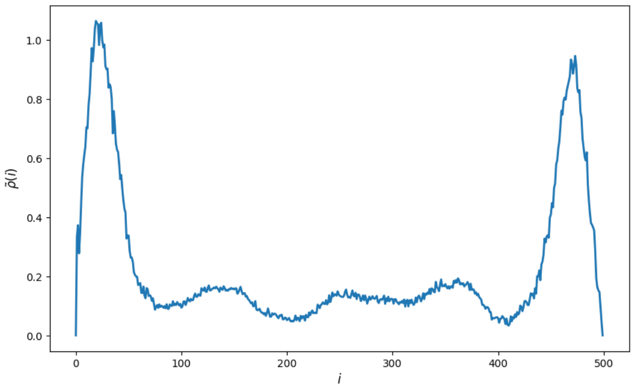

Figure 2: Smoothed Cluster Crossing $\tilde{\rho}(i)$ in the B5 Newspaper setting. The identification of the 4 valleys becomes slightly more ambiguous.