KMEANS CLUSTERING BASED ON CHEBYSHEY POLYNOMIAL GRAPH FILTERING

Liang Du[⋆], *Yunhui Liang*[⋆], *Mian Ilyas Ahmad*[⋆⋆], *Peng Zhou*[††]

[⋆] School of Computer and Information Technology, Shanxi University, China
[††] School of Computer Science and Technology, Anhui University, China
[⋆⋆] National University of Sciences and Technology, Pakistan

ABSTRACT

Clustering, a key unsupervised learning method, is widely used in various fields. While the classic k-means algorithm is popular, it often neglects valuable high-order neighborhood information in data. To address this, we propose an improved K-means algorithm that incorporates Chebyshev polynomial-approximated graph filters. This approach enhances representational capabilities and has shown superior performance in experiments with multiple datasets. It adds a novel dimension to k-means clustering, promising further advancements in the field. The benefits are twofold. First, the adaptive graph filter, approximated by Chebyshev polynomial, provides a smoother representation that aids subsequent clustering. Second, the clustering results guide the search for optimal adaptive coefficients. This synergy within a unified framework mutually enhances K-means clustering and adaptive coefficient learning. This work underscores the potential of integrating filter design with machine learning tasks.

Index Terms— Chebyshev polynomials, graph filtering, K-means, unsupervised learning

1. INTRODUCTION

Cluster algorithms are essential techniques in the fields of machine learning [1, 2] and data analysis[3, 4], used to partition objects within a dataset into groups or clusters based on their similar features. The primary objective of clustering algorithms is to automatically discover underlying structures within data without requiring explicit class labels, enabling data understanding, analysis, and processing. There are various types of clustering algorithms, including partitioning clustering[5], subspace clustering[6, 7], spectral clustering[8], among others. One of the most common partitioning clustering algorithms is K-means clustering. Its steps involve selecting initial cluster centers, assigning data points to the nearest clusters, updating cluster centers, and iterating these steps until convergence. K-means clustering is favored for its simplicity, efficiency, and stable performance. It has long been considered a standard clustering method in many machine learning studies and finds extensive applications in various data analysis and data mining tasks.

In recent years, researchers have designed various algorithms for clustering problems. Ehsan et al.[9] proposed an algorithm called sparse subspace clustering for clustering data points in low-dimensional subspace union. Nie et al.[10] perform clustering by learning a graph with exactly k connected components using the constrained Laplacian rank method. Nie et al.[11] generalize the kmeans of multi-means clustering and propose a k-multiple-means method to group data points with Multiple subcluster means into a specified k cluster. Zhou et al.[12] proposed a new optimal neighborhood multi-view spectral clustering algorithm. Wen et al.[13] proposed an incomplete multi-view clustering algorithm combined with adaptive graph learning. Han et al.[14] proposed a multi-view k-means clustering algorithm combining adaptive sparse membership and weight allocation. Huang et al.[15] proposed a super-scalable spectral clustering algorithm. Xia et al.[16] proposed a fast k-means algorithm with no bounds. Yang et al.[17] proposed a clustering algorithm for large-scale hypergraphs. Kang et al.[18] proposed a structured graph learning method for subspace clustering. Tang et al.[19] proposed a unified one-step multi-view spectral clustering algorithm that integrates spectral embedding and kmeans into a unified framework and uses a partial strategy to obtain discrete clustering labels. Nie et al.[20] reformulated the classical k-means objective function as a trace maximization problem, replacing it with a new formulation.

However, these existing clustering methods have significant limitations. They rarely consider higher-order relationships among data points, which can be crucial for capturing complex data structures. Moreover, they often struggle with noisy data, which is a common issue in real-world applications. These limitations underscore the need for more robust and comprehensive clustering algorithms that can effectively leverage higher-order graph structures and enhance noise tolerance.

To address these issues, we propose an improved K-means algorithm based on Chebyshev polynomial approximation of graph filters. This methodology not only leverages the first-order structural information of the graph but also captures high-order neighborhood information to better characterize the intrinsic structure of the data. Additionally, the graph filter can act on the data to obtain its smoothed representation, facilitating the extraction of meaningful data for clustering purposes.

Our contributions are summarized as follows.

- We propose to integrate graph filtering and kmeans clustering within a unified framework. On the one hand, the Kmeans clustering results can be used to guide the learning of optimal graph filter, i.e., the coefficients of Chebyshev polynomial. On the other hand, the adaptive graph filter can be used to smooth out the noises and conduct kmeans clustering on a much clean representation.
- We developed an efficient algorithm to solve the optimization problems involved, and verified the convergence of the method through experiments.
- By comparing the performance of multiple comparison methods on multiple data sets, a large number of experimental results verify the effectiveness of the proposed method. Compared with other algorithms, the performance of our algorithm is better and more stable.

2. THE PROPOSED METHOD

In this section, we briefly introduce some related knowledge. We denote the data matrix $\mathbf{X} \in \mathcal{R}^{n\times d}$, where n is the number of the samples and d is the dimension.

2.1. Chebyshev Polynomial Approximation Graph Filter

The importance of graph structure in clustering problems is that it can describe the relationship between data more comprehensively and provide richer information to help clustering algorithms better understand and separate data. In order to better capture the structural information of data, we adopt an initializing graph method to learn a non-negative normalized similarity matrix $\mathbf{S}$ [21]. Each of its elements is defined as:

$$\mathbf{S}_{i,j} = \begin{cases} \frac{\mathbf{d}_{i,k+1} - \mathbf{d}i,j}{\mathbf{d}_{i,k+1} - \sum_{h=1}^{k} \mathbf{d}_{i,k}} & j \leq k, \\ 0 & j > k, \end{cases} \tag{1}$$

where k is the number of nearest neighbor samples, $d_{i,j}$ is the i,j element of the data Euclidean distance matrix $\mathbf{D} \in \mathcal{R}^{n\times n}$.

Although first-order graph can describe the structure of data well, it does not take into account the higher-order neighborhood information of data. The complex high-level relationship mining in the data is helpful to improve the quality of the graph construction and the effect of the learning task. Therefore, we use Chebyshev polynomials to construct different order similarity matrices. The following is the recurrence relationship of different order graphs.

$$\begin{cases} \mathbf{T}_0 = \mathbf{I}, \\ \mathbf{T}_1 = \mathbf{S}, \\ \mathbf{T}_m = \mathbf{T}_m - \mathbf{T}_{m-1}. \end{cases} \tag{2}$$

After getting the different order graphs, we introduce a set of weight coefficients $\boldsymbol{\beta} = [\beta_1, \beta_2, ..., \beta_m]^T$ to aggregate them to get the graph filter.

$$\mathbf{G} = \sum_{m=0}^{\bar{m}} \beta_m \mathbf{T}_m \tag{3}$$

The Chebyshev polynomial approximation graph filter is a specific design based on Chebyshev polynomials, which can act on the data to make it smoother, while capturing higher-order neighborhood information of different orders of graphs to get a better representation of the data. In the next, we propose an improved k-means clustering algorithm based on Chebyshev polynomial approximation in order to explore better representation capabilities and more fully utilize first-order and higher-order affinity information in the data. In general, our algorithm can use the potentially rich information of different order graphs to act on the data to obtain its smooth representation and improve the performance of k-means algorithm. This idea can be intuitively implemented as follows:

$$\begin{aligned} \min_{\mathbf{Y},\mathbf{U},\boldsymbol{\beta}} \quad & \sum_{i=1}^{n}\sum_{j=1}^{c} \mathbf{Y}_{i,j} \left\|(\mathbf{G}\mathbf{X})_i - \mathbf{u}_j\right\|^2 \\ \text{s.t.} \quad & \sum_{m=1}^{\bar{m}} \beta_m = 1, 0 \leq \beta_m \leq 1, \end{aligned} \tag{4}$$

where $\mathbf{Y} \in \{0,1\}^{n\times c}$ is class indicator matrix. $\mathbf{G}$ is the Chebyshev polynomial approximation graph filter, $\mathbf{U} \in R^{c\times d}$ is the centers of the j clusters, β_m is the combination weight of the m-th Order Similarity graph. Notably, the constraint of β is added to make sure that the optimization process is stable. In the following, we design an efficient alternative optimization algorithm to solve the problem in Eq. (4).

2.2. Optimizing Y

Given $\mathbf{U}$ and $\boldsymbol{\beta}$, the optimization problem in Eq. (4) w.r.t. $\mathbf{Y}$ easily yields an analytical solution. Since the different i dependent terms are independent, we can optimize each i separately and set $Y_{i,j}$ equal to 1 as long as the value of j minimizes $\|(\mathbf{G}\mathbf{X})_i - \mathbf{u}_j\|^2$. Formalized can be expressed as:

$$\mathbf{Y}_{i,j} = \begin{cases} 1 & \text{if } j = \arg\min_{j'} \|(\mathbf{G}\mathbf{X})_i - \mathbf{u}_{j'}\|^2 \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

2.3. Optimizing U

Given $\mathbf{Y}$ and $\boldsymbol{\beta}$ the optimization problem in Eq. (4) w.r.t. $\mathbf{U}$ reduces to:

$$\min_{\mathbf{U}} \sum_{i=1}^{n}\sum_{j=1}^{c} \mathbf{Y}_{i,j} \left\{ \text{tr}(\mathbf{u}_j\mathbf{u}_j^T) - 2tr(\mathbf{u}_j(\mathbf{G}\mathbf{X})_i^T) \right\} \tag{6}$$

Eq. (6) is a quadratic function of $\mathbf{u}_j$, and its derivative with respect to $\mathbf{u}_j$ is equal to zero to reach the minimum value, i.e.

$$\sum_{i=1}^{n} \mathbf{Y}_{i,j}((\mathbf{GX})_i - \mathbf{u}_j) = 0. \tag{7}$$

We can easily get the result for $\mathbf{U}_k$ as:

$$\mathbf{u}_j = \frac{\sum_{i=1}^{n} \sum_{j=1}^{c} \mathbf{Y}_{i,j} (\mathbf{GX})_i}{\sum_{i=1}^{n} \sum_{j=1}^{c} \mathbf{Y}_{i,j}} \tag{8}$$

The denominator of this expression is equal to the number of data points in the cluster j, so the implication of the result is that $\mathbf{u}_j$ is equal to the average of all data points in class j.

2.4. Optimizing $\boldsymbol{\beta}$

Given $\mathbf{Y}$ and $\mathbf{U}$, the optimization problem in Eq. (4) w.r.t. $\boldsymbol{\beta}$ reduces to the following formulation:

$$\begin{aligned} \min_{\boldsymbol{\beta}} & \sum_{i=1}^{n} \sum_{j=1}^{c} \mathbf{Y}_{i,j} \{ (\sum_{m=1}^{\bar{m}} \beta_m \mathbf{T}_m \mathbf{X})_i (\sum_{m=1}^{\bar{m}} \beta_m \mathbf{T}_m \mathbf{X})_i^T \\ & - 2 (\sum_{m=1}^{\bar{m}} \beta_m \mathbf{T}_m \mathbf{X})_i \mathbf{u}_j^T \} \\ \text{s.t.} & \sum_{m=1}^{\bar{m}} \beta_m = 1, 0 \leq \beta_m \leq 1. \end{aligned} \tag{9}$$

Obviously problem Eq. (9) is a quadratic optimization problem about $\boldsymbol{\beta}$, which can be transformed into the following form:

$$\begin{aligned} \min_{\boldsymbol{\beta}} \quad & \boldsymbol{\beta}^T \mathbf{H} \boldsymbol{\beta} + \boldsymbol{\beta}^T \mathbf{f} \\ \text{s.t.} \quad & \sum_{m=1}^{\bar{m}} \beta_m = 1, 0 \leq \beta_m \leq 1, \end{aligned} \tag{10}$$

where $\mathbf{H} \in \mathbb{R}^{\bar{m} \times \bar{m}}$ is a quadratic matrix whose p, q-th element is

$$\mathbf{H}_{p,q} = \sum_{i=1}^{n} \sum_{j=1}^{c} \mathbf{Y}_{i,j} (\mathbf{T}_p \mathbf{X})_i (\mathbf{T}_q \mathbf{X})_i^T \tag{11}$$

$\mathbf{f}$ is a vector and the m-th element is

$$\mathbf{f}_m = -2 \sum_{i=1}^{n} \sum_{j=1}^{c} \mathbf{Y}_{i,j} (\mathbf{T}_m \mathbf{X})_i \mathbf{U}_j^T \tag{12}$$

Eq. (10) is a standard quadratic optimization problem that can be easily solved using existing optimization tools.

To sum up, we summarize the complete process of our algorithm in algorithm 1.

Algorithm 1 The optimization algorithm of Eq. (4)

```
Input Data $\mathbf{X}$, the number of Order $\bar{m}$, the number of cluster $c$
Initialize Weight coefficient $\boldsymbol{\beta} = \frac{1}{\bar{m}}$
Construct Similarity matrix $\mathbf{S}$ according to Eq. (1)
Construct Graph filter $\mathbf{G}$ according to Eq. (3)
while not converged do
    Update $\mathbf{Y}$ according to Eq. (5);
    Update $\mathbf{U}$ according to Eq. (8);
    Update $\boldsymbol{\beta}$ according to Eq. (10);
end while
Output Cluster indicator matrix $\mathbf{Y}$
```

3. EXPERIMENTS

3.1. Datasets Description

We evaluated the algorithm's clustering performance on eight popular datasets from different applications. The details of these datasets are listed in Table 1.

Table 1: Descriptions of Eight Datasets

Date ID	Datasets	#Samples	#Dimensions	#Classes
D1	FACS	1013	13741	7
D2	BA	1404	320	36
D3	ISOLET	1560	617	26
D4	MSRA25	1799	256	12
D5	NEWS4A	3840	4989	4
D6	MNIST	4000	754	10
D7	CALTECH101	8671	784	101
D8	USPS	9298	256	10

Table 2: ACC of Eight Clustering Algorithms on Eight Datasets

Datasets	KMEANS	CLR	SKM	KMM	SGL	KMWNF	RBSMF	CGFKM
D1	0.5189	0.4107	0.5432	0.5770	0.6456	0.6219	0.6170	**0.7038**
D2	0.4123	0.3184	0.4038	0.2803	0.2735	0.4174	0.4330	**0.5014**
D3	0.5822	0.5436	0.5611	0.5003	0.3673	0.6500	0.5500	**0.6453**
D4	0.5514	0.5381	0.4985	0.5118	0.5814	0.5992	0.5717	**0.6570**
D5	0.2824	0.2570	0.2750	0.2569	0.2591	0.2896	0.3053	**0.3168**
D6	0.5009	0.3638	0.5118	0.6076	0.5003	0.5125	0.5404	**0.6691**
D7	0.2686	0.1696	0.2721	0.2356	0.1807	0.2787	0.2794	**0.2850**
D8	0.6491	0.7556	0.6523	0.6587	0.6853	0.6733	0.6964	**0.8002**
AVG	0.4707	0.4196	0.4647	0.4535	0.4367	0.5053	0.4991	**0.5723**

3.2. Experimental Setup

In our experiments, we compared several classic or advanced clustering algorithms in the field of clustering, including:K-means[22], Constrained Laplacian Rank(CLR)[10], Sparsified K-means(SKM)[23], K-Multiple-Means(KMM)[11], Structured Graph Learning(SGL)[18], K-Means with new formulation(KMENF)[20], robust bi-stochastic graph regularized matrix factorization(RBSMF)[24]. For fair comparison, we conducted parameter searches based on the hyperparameter settings from relevant literature and evaluated the results of different algorithms uniformly. In our proposed method, there are two parameters involved. The range of the order m is set to $[3 : 1 : 9]$, the number of k-neighbors is chosen in the range $[5, 10]$.

3.3. Experimental Results and Analyses

Three popular metrics are applied to quantitatively evaluate the clustering performance. They are accuracy (ACC), normalized mutual information (NMI), and purity (PURITY). After running each method under the optimal parameter settings, the clustering results of 8 different algorithms are shown in Table Figure 2, Figure 3, Figure 4. As can be seen from the experimental results, we can get the following interesting points and some detailed analysis.

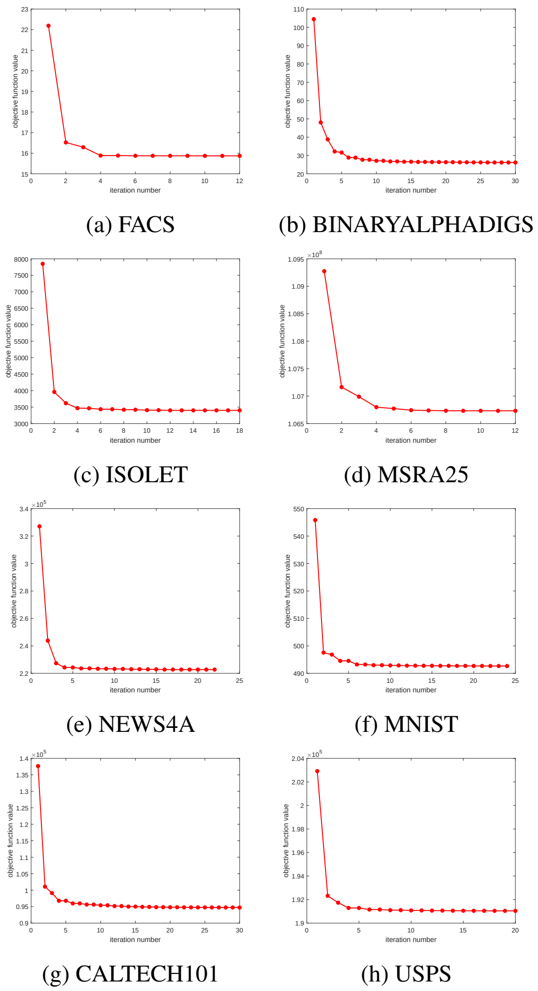
On all datasets, our method compares to other methods Good clustering performance is obtained. Compared with k-means algorithm, our method improves ACC, NMI and purity by 21.58%, 24.27% and 18.49% respectively. Compared with the second best clustering algorithm, our method improves ACC, NMI and purity by 13.26%, 17.73% and 12.79% respectively. The experimental results show that by introducing the graph filter to capture the higher-order neighborhood information of data, not only the performance of k-means algorithm is improved, but also better results are obtained compared with other advanced clustering algorithms, which fully verifies the effectiveness of using Chebyshev polynomial approximate graph filter to capture the higher-order neighborhood information.

Table 3: NMI of Eight Clustering Algorithms on Eight Datasets

Datasets	KMEANS	CLR	SKM	KMM	SGL	KMWNF	RBSMF	CGFKM
D1	0.4947	0.2691	0.5010	0.4946	0.5454	0.5375	0.5453	**0.7104**
D2	0.5699	0.4004	0.5645	0.3979	0.3714	0.5799	0.5842	**0.6401**
D3	0.7456	0.6971	0.7317	0.7045	0.5233	0.7754	0.7105	**0.7955**
D4	0.6028	0.6920	0.5819	0.5831	0.6721	0.6263	0.6462	**0.7506**
D5	0.0256	0.0042	0.0207	0.0056	0.0079	0.0297	0.0495	**0.0630**
D6	0.4739	0.4408	0.4727	0.6339	0.5002	0.4799	0.4976	**0.6742**
D7	0.5179	0.2180	0.5198	0.2791	0.3333	0.5225	0.5237	**0.5310**
D8	0.6036	0.8337	0.6077	0.7341	0.6667	0.6138	0.7012	**0.8488**
AVG	0.5043	0.4444	0.5000	0.4791	0.4525	0.5206	0.5323	**0.6267**

Table 4: PURITY of Eight Clustering Algorithms on Eight Datasets

Datasets	KMEANS	CLR	SKM	KMM	SGL	KMWNF	RBSMF	CGFKM
D1	0.6270	0.4235	0.6378	0.5972	0.6713	0.6831	0.6969	**0.7868**
D2	0.4384	0.3526	0.4287	0.3068	0.3013	0.4509	0.4623	**0.5328**
D3	0.6279	0.5776	0.6087	0.5288	0.4167	**0.7038**	0.5925	0.6842
D4	0.5740	0.6092	0.5386	0.5241	0.6165	0.6003	0.6001	**0.6693**
D5	0.2834	0.2570	0.2761	0.2585	0.2612	0.2922	0.3069	**0.3214**
D6	0.5538	0.4208	0.5591	0.6470	0.5743	0.5553	0.5851	**0.7173**
D7	0.4547	0.2829	0.4546	0.3228	0.3077	0.4692	0.4641	**0.4732**
D8	0.7153	0.8160	0.7212	0.7329	0.7306	0.7356	0.7616	**0.8797**
AVG	0.5343	0.4674	0.5281	0.4898	0.4849	0.5613	0.5587	**0.6331**



(a) FACS (b) BINARYALPHADIGS

(c) ISOLET (d) MSRA25

(e) NEWS4A (f) MNIST

(g) CALTECH101 (h) USPS

Fig. 1: Convergence curves of our algorithm.

3.4. Algorithm Convergence

The examples of the objective values of our algorithm at each iteration are shown in Figure 1. As observed from these figures, the objective value is monotonically decreased and the algorithm quickly converges in less than thirty iterations.

4. CONCLUSION

This paper presents an improved k-means clustering algorithm based on Chebyshev polynomial-approximated graph filters. This method utilizes Chebyshev polynomial approximation to linearly combine different-order similarity graphs, capturing high-order neighborhood information between data points. It can simultaneously leverage both first-order and high-order connectivity information to achieve data smoothing, thereby enhancing clustering performance. Experimental results validate the effectiveness of this algorithm, and we plan to further investigate the impact of graph filters on clustering algorithm performance in future research.

5. REFERENCES

[1] Greg Hamerly and Charles Elkan, "Learning the k in k-means," *Advances in neural information processing systems*, vol. 16, 2003.

[2] Aristidis Likas, Nikos Vlassis, and Jakob J Verbeek, "The global k-means clustering algorithm," *Pattern recognition*, vol. 36, no. 2, pp. 451–461, 2003.

[3] Yi Lu, Shiyong Lu, Farshad Fotouhi, Youping Deng, and Susan J Brown, "Incremental genetic k-means algorithm and its application in gene expression data analysis," *BMC bioinformatics*, vol. 5, no. 1, pp. 1–10, 2004.

[4] Abiodun M Ikotun, Absalom E Ezugwu, Laith Abualigah, Belal Abuhaija, and Jia Heming, "K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data," *Information Sciences*, 2022.

[5] S Anitha Elavarasi, J Akilandeswari, and B Sathiyabhama, "A survey on partition clustering algorithms," *International Journal of Enterprise Computing and Business Systems*, vol. 1, no. 1, 2011.

[6] Pan Ji, Tong Zhang, Hongdong Li, Mathieu Salzmann, and Ian Reid, "Deep subspace clustering networks," *Advances in neural information processing systems*, vol. 30, 2017.

[7] Zhenglai Li, Chang Tang, Xiao Zheng, Xinwang Liu, Wei Zhang, and En Zhu, "High-order correlation preserved incomplete multi-view subspace clustering," *TIP*, vol. 31, pp. 2067–2080, 2022.

[8] Peter Macgregor and He Sun, "A tighter analysis of spectral clustering, and beyond," in *International Conference on Machine Learning*, 2022, pp. 14717–14742.

[9] Ehsan Elhamifar and René Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 11, pp. 2765–2781, 2013.

[10] Feiping Nie, Xiaoqian Wang, Michael Jordan, and Heng Huang, "The constrained laplacian rank algorithm for graph-based clustering," in *AAAI*, 2016, vol. 30.

[11] Feiping Nie, Cheng-Long Wang, and Xuelong Li, "K-multiple-means: A multiple-means clustering method with specified k clusters," in *KDD*, 2019, pp. 959–967.

[12] Sihang Zhou, Xinwang Liu, Jiyuan Liu, Xifeng Guo, Yawei Zhao, En Zhu, Yongping Zhai, Jianping Yin, and Wen Gao, "Multi-view spectral clustering with optimal neighborhood laplacian matrix," in *AAAI*, 2020, vol. 34, pp. 6965–6972.

[13] Jie Wen, Yong Xu, and Hong Liu, "Incomplete multi-view spectral clustering with adaptive graph learning," *IEEE transactions on cybernetics*, vol. 50, no. 4, pp. 1418–1429, 2018.

[14] Junwei Han, Jinglin Xu, Feiping Nie, and Xuelong Li, "Multi-view k-means clustering with adaptive sparse memberships and weight allocation," *TKDE*, vol. 34, no. 2, pp. 816–827, 2020.

[15] Dong Huang, Chang-Dong Wang, Jian-Sheng Wu, Jian-Huang Lai, and Chee-Keong Kwoh, "Ultra-scalable spectral clustering and ensemble clustering," *TKDE*, vol. 32, no. 6, pp. 1212–1226, 2019.

[16] Shuyin Xia, Daowan Peng, Deyu Meng, Changqing Zhang, Guoyin Wang, Elisabeth Giem, Wei Wei, and Zizhong Chen, "A fast adaptive k-means with no bounds," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[17] Yiyang Yang, Sucheng Deng, Juan Lu, Yuhong Li, Zhiguo Gong, Zhifeng Hao, et al., "Graphlshc: towards large scale spectral hypergraph clustering," *Information Sciences*, vol. 544, pp. 117–134, 2021.

[18] Zhao Kang, Zhiping Lin, Xiaofeng Zhu, and Wenbo Xu, "Structured graph learning for scalable subspace clustering: From single view to multiview," *IEEE Transactions on Cybernetics*, vol. 52, no. 9, pp. 8976–8986, 2021.

[19] Chang Tang, Zhenglai Li, Jun Wang, Xinwang Liu, Wei Zhang, and En Zhu, "Unified one-step multi-view spectral clustering," *TKDE*, vol. 35, no. 6, pp. 6449–6460, 2022.

[20] Feiping Nie, Ziheng Li, Rong Wang, and Xuelong Li, "An effective and efficient algorithm for k-means clustering with new formulation," *TKDE*, vol. 35, no. 4, pp. 3433–3443, 2022.

[21] Feiping Nie, Xiaoqian Wang, and Heng Huang, "Clustering and projected clustering with adaptive neighbors," in *KDD*, 2014, pp. 977–986.

[22] Christopher M Bishop and Nasser M Nasrabadi, *Pattern recognition and machine learning*, vol. 4, Springer, 2006.

[23] Farhad Pourkamali-Anaraki and Stephen Becker, "Preconditioned data sparsification for big data with applications to pca and k-means," *IEEE Transactions on Information Theory*, vol. 63, no. 5, pp. 2954–2974, 2017.

[24] Qi Wang, Xiang He, Xu Jiang, and Xuelong Li, "Robust bi-stochastic graph regularized matrix factorization for data clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 1, pp. 390–403, 2020.