PeerJ Computer Science

A new Kmeans clustering model and its generalization achieved by joint spectral embedding and rotation

Wenna Huang[1], Yong Peng[1,2], Yuan Ge[3] and Wanzeng Kong[1,4]

[1] School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou, Zhejiang, China
[2] Center for OPTical IMagery Analysis and Learning (OPTIMAL), Northwestern Polytechnical University, Xi'an, Shaanxi, China
[3] MoE Key Laboratory of Advanced Perception and Intelligent Control of High-end Equipment, Anhui Polytechnic University, Wuhu, Anhui, China
[4] Key Laboratory of Brain Machine Collaborative Intelligence of Zhejiang Province, Hangzhou, Zhejiang, China

ABSTRACT

The Kmeans clustering and spectral clustering are two popular clustering methods for grouping similar data points together according to their similarities. However, the performance of Kmeans clustering might be quite unstable due to the random initialization of the cluster centroids. Generally, spectral clustering methods employ a two-step strategy of spectral embedding and discretization postprocessing to obtain the cluster assignment, which easily lead to far deviation from true discrete solution during the postprocessing process. In this paper, based on the connection between the Kmeans clustering and spectral clustering, we propose a new Kmeans formulation by joint spectral embedding and spectral rotation which is an effective postprocessing approach to perform the discretization, termed KMSR. Further, instead of directly using the dot-product data similarity measure, we make generalization on KMSR by incorporating more advanced data similarity measures and call this generalized model as KMSR-G. An efficient optimization method is derived to solve the KMSR (KMSR-G) model objective whose complexity and convergence are provided. We conduct experiments on extensive benchmark datasets to validate the performance of our proposed models and the experimental results demonstrate that our models perform better than the related methods in most cases.

Submitted 29 January 2021
Accepted 1 March 2021
Published 30 March 2021

Corresponding author
Yong Peng, yongpeng@hdu.edu.cn

Academic editor
Qichun Zhang

Additional Information and Declarations can be found on page 20

DOI **10.7717/peerj-cs.450**

Subjects Algorithms and Analysis of Algorithms, Artificial Intelligence, Data Mining and Machine Learning
Keywords Kmeans clustering, Spectral clustering, Spectral rotation, Data similarity

INTRODUCTION

Clustering is one of the important research contents in many communities such as data mining and pattern recognition. Basically, it aims to group the data points into different clusters according to their similarities or densities (*Ubukata, 2019*; *Ren, Zhang & Zhang, 2019*). Over the past decades, a number of clustering algorithms have been proposed such as the *K*means clustering, spectral clustering (*Ng, Jordan & Weiss, 2001*), min-max cut (*Ding et al., 2001*; *Nie et al., 2010*), subspace clustering (*Nie & Huang, 2016*; *Xie et al., 2020*), and multi-view clustering (*Nie, Tian & Li, 2018*; *Cai et al., 2013*). Among the existing clustering methods, the most popular one is the *K*means clustering algorithm due to its simpleness

How to cite this article Huang W, Peng Y, Ge Y, Kong W. 2021. A new Kmeans clustering model and its generalization achieved by joint spectral embedding and rotation. *PeerJ Comput. Sci.* **7**:e450 http://doi.org/10.7717/peerj-cs.450

and efficiency, which aims to learn certain cluster centroids to minimize the within cluster data distances. However, the *K*means algorithm suffers a great impact on its clustering performance due to the random initialization of cluster centroids.

By characterizing the data connection with an appropriate graph whose vertices represent the data points and the weights represent the connection between data pairs, spectral clustering tries to partition the vertices into different clusters by minimizing the cut information. There are some popular spectral clustering algorithms such as the ratio cut (RCut) (*Hagen & Kahng, 1992*), normalized cut (NCut) (*Shi & Malik, 1997*), clustering with adaptive neighbors (CAN) and its projected version (PCAN) (*Nie, Wang & Huang, 2014*), multiclass spectral clustering (*Yu & Shi, 2003*), constrained Laplacian rank (*Nie et al., 2016*), and nonnegative matrix factorization (*Peng et al., 2018*). Given a built graph, existing spectral clustering methods usually employ a two-step strategy to complete the clustering; one is performing eigen-decomposition on the graph Laplacian matrix to obtain the scaled cluster indicator matrix based on which the other aims to make discretization to get the final cluster assignment. The former step is recognized as spectral embedding and the latter step as postprocessing. Generally, the existing two approaches to complete the postprocessing task of recovering the final discrete cluster indicators from the relaxed continuous spectral vectors are the *K*means clustering and spectral rotation. As pointed by (*Huang, Nie & Huang, 2013*; *Chen et al., 2017*), using spectral rotation as the postprocessing step can usually obtain better clustering performance than that of *K*means postprocessing. However, such two-stage process has an obvious disadvantage that the final assignments may deviate far from the true discrete solution (*Huang, Nie & Huang, 2013*).

As mentioned above, both the *K*means clustering and the spectral clustering have limitations in respective fields. To this end, in this paper, we first derive the underlying connection between the *K*means clustering and spectral clustering, and then propose a new *K*means formulation by jointly performing spectral embedding and spectral rotation. The resultant KMSR model can effectively alleviate the drawback of the randomness in the initialization of cluster centroids of *K*means. Moreover, the two sub-objectives of spectral embedding and spectral rotation are jointly optimized, which can co-evolve to the optimum and avoid the sub-optimality caused by the two-step strategy. Due to that the KMSR is originated from the *K*means clustering, it measures the data similarity by directly using the dot-product weighting scheme whose performance is limited in dealing with complicated data sets. To accommodate more advanced graphs and then improve the performance of KMSR, we make corresponding extension on it by replacing the dot-product of data matrix with predefined graphs such as CAN and PCAN (*Nie, Wang & Huang, 2014*), leading to the generalized version KMSR-G. Mathematically, the KMSR model objective involves three variables respectively corresponding to the relaxed continuous cluster indicator matrix, the discrete cluster indicator, and the orthogonal transformation matrix to bridging them; therefore, under the coordinate blocking framework, we design an efficient optimization method to alternately update them, whose complexity and convergence property are also analyzed. We conduct extensive experiments on representative benchmark data sets to evaluate the performance of our proposed models. By comparing the clustering performance of KMSR and KMSR-G with related models, they both perform better in most cases.

The remainder of this paper is organized as follows. We give brief introductions to some related works including the Kmeans, spectral clustering and spectral rotation in 'Related Work'. In 'The Proposed Model', we first derive the model formulation of KMSR based on depicting the connection between Kmeans clustering and spectral clustering, and then provide the detailed optimization process to KMSR model objective. Besides, the complexity and convergence analysis of KMSR, its generalization to KMSR-G are included. In 'Experiment', extensive experiments are conducted on representative benchmark data sets to evaluate the effectiveness of KMSR and KMSR-G in data clustering. 'Conclusion' concludes the whole paper and puts forward a future work.

Notations. In this paper, matrices are written as boldface uppercase letters. Vectors are written as boldface lowercase letters. For example, the (i,j)-th element of matrix $\mathbf{W}$ is w_{ij}. The squared ℓ_2-norm of matrix $\mathbf{W} \in \mathbb{R}^{n\times m}$ is $\|\mathbf{W}\|_2^2 = \sum_{i=1}^{n}\sum_{j=1}^{m} w_{ij}^2$.

By default, we use $\mathbf{w}_i$ to represent the ith column of $\mathbf{W}$ and $\mathbf{w}^j$ to represent its jth row. We use $\mathbb{R}$ and $\mathbb{B}$ to represent the real domain and the binary domains, respectively.

RELATED WORKS

Kmeans clustering

Given a data matrix $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n] \in \mathbb{R}^{d\times n}$, the Kmeans clustering aims to partition $\mathbf{X}$ into c $(1 \le c \le n)$ clusters $C = [C_1, C_2, \ldots, C_c]$ such that the within-cluster sum of squared distances can be minimized and the sum of squared distances between clusters can be maximized. Mathematically, the objective function of Kmeans clustering is

$$\min_{C} \sum_{i=1}^{c} \sum_{\mathbf{x}_j \in C_i} \left\| \mathbf{x}_j - \boldsymbol{\mu}_i \right\|_2^2, \tag{1}$$

where $\boldsymbol{\mu}_i$ is the centroid corresponding to the cluster C_i. To optimize objective Eq. (1), the membership of each data point and the centroid of each cluster are alternately updated.

Spectral clustering

For spectral clustering, we first need to construct a graph affinity matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ according to certain similarity measures to depict the connection between data pairs. Let $\mathbf{y}^i|_{i=1}^{n}$ be the ith row vector of matrix $\mathbf{Y} = [\mathbf{y}^1; \mathbf{y}^2; \cdots; \mathbf{y}^n] \in \mathbb{B}^{n\times c}$, which corresponds to the cluster indicator vector for $\mathbf{x}_i$. The jth element of $\mathbf{y}^i$ is 1 if $\mathbf{x}_i \in C_j$, and 0 otherwise. By defining the scaled cluster indicator matrix $\mathbf{F}$ as $\mathbf{F} = \mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-1/2}$ whose jth column is given by

$$\mathbf{f}_j = [\underbrace{0, \ldots, 0}_{\sum_{i=1}^{j-1} n_i}, \underbrace{1, \ldots, 1}_{n_j}, \underbrace{0, \ldots, 0}_{\sum_{i=j+1}^{c} n_i}]^T / \sqrt{n_j}, \tag{2}$$

where n_j is the number of data in the jth cluster. Then, the objective function of spectral clustering can be formulated as

$$\min_{\mathbf{F}} \mathrm{Tr}(\mathbf{F}^T\mathbf{L}\mathbf{F}), \quad s.t. \mathbf{F} = \mathbf{Y}\,(\mathbf{Y}^T\mathbf{Y})^{-1/2}. \tag{3}$$

Here $\mathbf{L}=\mathbf{D}-\mathbf{A}$ is the Laplacian matrix, where $\mathbf{D}$ is the diagonal degree matrix with its ith diagonal element defined as $d_{ii}=\sum_{j=1}^{n}a_{ij}$.

Since $\mathbf{F}^T\mathbf{F}=(\mathbf{Y}^T\mathbf{Y})^{-1/2}\mathbf{Y}^T\mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-1/2}=\mathbf{I}$, the embedding $\mathbf{F}$ can be obtained by stacking the eigenvectors of $\mathbf{L}$ corresponding to its c smallest eigen-values. However, $\mathbf{F}$ is a real-valued matrix and therefore a postprocessing step such as the Kmeans clustering or spectral rotation (*Huang, Nie & Huang, 2013*) is necessary to perform discretization.

It is easy to find that the solution to Eq. (3) is not unique. That is, for any solution $\mathbf{F}$, $\mathbf{FR}$ is another solution where $\mathbf{R}$ is an arbitrary orthogonal matrix. Therefore, spectral rotation aims at finding a proper orthogonal and normalized $\mathbf{R}$ such that the resultant $\mathbf{FR}$ are closer to the discrete indicator matrix solution set than the $\mathbf{F}$ in Kmeans. Mathematically, it aims to minimize the following objective

$$\min_{\mathbf{Y},\mathbf{R}}\|\mathbf{FR}-\mathbf{Y}\|_2^2,\quad s.t. \mathbf{Y}\in\mathbb{B}^{n\times c},\mathbf{Y}\mathbf{1}_c=\mathbf{1}_n,\mathbf{R}^T\mathbf{R}=\mathbf{I}, \tag{4}$$

where $\mathbf{1}_c$ and $\mathbf{1}_n$ are both all-one column vectors with sizes of $c\times 1$ and $n\times 1$, respectively. By using the alternative optimization method, objective Eq. (4) can be solved and therefore the final cluster assignment can be obtained.

THE PROPOSED MODEL

In this section, we formulate the model objective function of KMSR and derive its optimization method. Besides, the complexity and convergence analysis are provided.

Model formulation

By introducing two matrices $\mathbf{U}=[\boldsymbol{\mu}_1,\boldsymbol{\mu}_2,\ldots,\boldsymbol{\mu}_c]\in\mathbb{R}^{d\times c}$ and $\mathbf{Y}$ such that $(\mathbf{Y}\in\mathbb{B}^{n\times c},\mathbf{Y}\mathbf{1}_c=\mathbf{1}_n)$ to respectively represent the cluster centroids and indices, the Kmeans objective in Eq. (1) can be reformulated as

$$\min_{\mathbf{Y},\mathbf{U}}\|\mathbf{X}-\mathbf{U}\mathbf{Y}^T\|^2\Leftrightarrow\min_{\mathbf{Y},\mathbf{U}}\mathrm{Tr}(\mathbf{Y}\mathbf{U}^T\mathbf{U}\mathbf{Y}^T)-2\mathrm{Tr}(\mathbf{X}^T\mathbf{U}\mathbf{Y}^T). \tag{5}$$

Since the solution to $\mathbf{U}$ is $\mathbf{U}=\mathbf{X}\mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-1}$, we have $\mathrm{Tr}(\mathbf{Y}\mathbf{U}^T\mathbf{U}\mathbf{Y}^T)=\mathrm{Tr}(\mathbf{X}^T\mathbf{U}\mathbf{Y}^T)$ and then Eq. (5) can be written as

$$\begin{aligned}&\min_{\mathbf{Y},\mathbf{U}}-\mathrm{Tr}(\mathbf{X}^T\mathbf{U}\mathbf{Y}^T)\Leftrightarrow\min_{\mathbf{Y}}-\mathrm{Tr}((\mathbf{Y}^T\mathbf{Y})^{-\frac{1}{2}}\mathbf{Y}^T(\mathbf{X}^T\mathbf{X})\mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-\frac{1}{2}})\\&\Leftrightarrow\min_{\mathbf{F}}-\mathrm{Tr}(\mathbf{F}^T(\mathbf{X}^T\mathbf{X})\mathbf{F}),\end{aligned} \tag{6}$$

where $\mathbf{F}\in\mathbb{R}^{n\times c}$ and $\mathbf{F}\triangleq\mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-\frac{1}{2}}$.

By checking Eq. (6), an usual way to solve it is relaxing the binary matrix $\mathbf{F}$ to real domain but keeping its orthogonality intact. Then, we obtain the following formulation

$$\min_{\mathbf{F}}-\mathrm{Tr}(\mathbf{F}^T\mathbf{X}^T\mathbf{X}\mathbf{F}),\quad s.t. \mathbf{F}^T\mathbf{F}=\mathbf{I},\mathbf{F}\in\mathbb{R}^{n\times c}. \tag{7}$$

Note that $\mathbf{F}$ in Eq. (7) is the continuous relaxation which preserves the orthogonality but misses the discrete nature of $\mathbf{Y}$. If we use Kmeans to find the cluster assignment which aims to jointly find the cluster indicator matrix $\mathbf{Y}$ and centroids $\mathbf{C}$ by $\min_{\mathbf{Y}\in\mathbb{B}^{n\times c},\mathbf{Y}\mathbf{1}_c=\mathbf{1}_n,\mathbf{C}}\|\mathbf{F}-\mathbf{Y}\mathbf{C}\|_F^2$, it can only guarantee that $\mathbf{YC}$ best approximates the relaxed continuous vector matrix **F** and cannot guarantee that such yielded **Y** best approximates **Y**. As mentioned in 'Spectral Clustering', the relaxed solution to Eq. (7) is not unique. Actually, for any solution **F**, **FQ** is another solution where **Q** is an arbitrary orthonormal matrix. The goal of spectral rotation is to find a proper **Q** such that the resulting **FQ** are closer to the discrete indicator matrix solution set than the **F** in *K*means. Therefore, we can take the idea of spectral rotation into account to perform post-processing operation on the optimal $\mathbf{F}^*$ of Eq. (7) to obtain the final cluster indicator matrix.

Inspired by *Chen et al. (2017)*, we aim at finding an orthonormal matrix $\mathbf{Q} \in \mathbb{R}^{c\times c}$ to minimize the discrepancy between $\mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-\frac{1}{2}}$ and $\mathbf{F}^*\mathbf{Q}$ as the postprocessing step. Mathematically, it can be achieved by solving the following objective

$$\min_{\mathbf{Y}\in\mathbb{B}^{n\times c},\mathbf{Q}} \|\mathbf{Y}\,(\mathbf{Y}^T\mathbf{Y})^{-\frac{1}{2}} - \mathbf{F}^*\mathbf{Q}\,\|_F^2, \quad s.t.\mathbf{Q}^T\mathbf{Q} = \mathbf{I}, \mathbf{Q} \in \mathbb{R}^{c\times c}. \tag{8}$$

From the above analysis, we realize that an intuitive way to handle a spectral clustering task is to get $\mathbf{F}^*$ by solving Eq. (7) followed by an appropriate **Q** from Eq. (8) to finally obtain the final cluster indicator matrix **Y**. To avoid the sub-optimality caused by such two-step process, in this paper, we propose to jointly optimize the objectives of Eq. (7) and Eq. (8) which respectively correspond to the spectral embedding and rotation, leading to the following new *K*means formulation (termed **KMSR**) as

$$\min_{\mathbf{Y},\mathbf{F},\mathbf{Q}} -\mathrm{Tr}(\mathbf{F}^T\mathbf{X}^T\mathbf{X}\mathbf{F}) + \lambda\|\mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-\frac{1}{2}} - \mathbf{F}\mathbf{Q}\|_F^2, \quad s.t.\mathbf{Y} \in \mathbb{B}^{n\times c}, \mathbf{Y}\mathbf{1}_c = \mathbf{1}_n,$$
$$\mathbf{F}^T\mathbf{F} = \mathbf{I}, \mathbf{Q}^T\mathbf{Q} = \mathbf{I}, \tag{9}$$

where $\lambda > 0$ is a regularization parameter to control the balance between the two items.

In spectral clustering, the normalized Laplacian matrix $\mathbf{L}_n$ is defined as

$$\mathbf{L}_n = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}. \tag{10}$$

If we replace **L** in Eq. (3) with $\mathbf{L}_n$, it becomes the objective function of the normalized cut. Since $\mathrm{Tr}(\mathbf{F}^T\mathbf{F})$ is a constant, we can find an interesting point that there exists an equivalence between the *K*means clustering and the normalized cut. That is, the graph affinity matrix in *K*means clustering employs the simple dot-product weighting scheme, i.e., $\mathbf{X}^T\mathbf{X}$, while it is $\tilde{\mathbf{A}} \triangleq \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$ in normalized cut.

We know that the graph quality plays an important role in spectral clustering. Sometimes, we need to learn a more robust graph to characterize the connection among data points than simply constructing it by fixed rules. Therefore, to further enhance the model performance, we make the generalization on KMSR by introducing $\tilde{\mathbf{A}}$ as the graph matrix. That is, we can incorporate more advanced graphs instead of the simple dot-product weighting scheme. We name the generalized model as **KMSR-G** which has the following objective function

$$\min_{\mathbf{Y},\mathbf{F},\mathbf{Q}} -\mathrm{Tr}(\mathbf{F}^T\tilde{\mathbf{A}}\mathbf{F}) + \lambda\|\mathbf{Y}\,(\mathbf{Y}^T\mathbf{Y})^{-\frac{1}{2}} - \mathbf{F}\mathbf{Q}\|_2^2, \quad s.t.\mathbf{Y} \in \mathbb{B}^{n\times c}, \mathbf{Y}\mathbf{1}_c = \mathbf{1}_n, \mathbf{F}^T\mathbf{F} = \mathbf{I}, \mathbf{Q}^T\mathbf{Q} = \mathbf{I}. \tag{11}$$

Obviously, KMSR-G is a general model to accommodate any predefined (or pre-learned) graph, which is expected to achieve better performance than KMSR especially when handling complicated data sets.

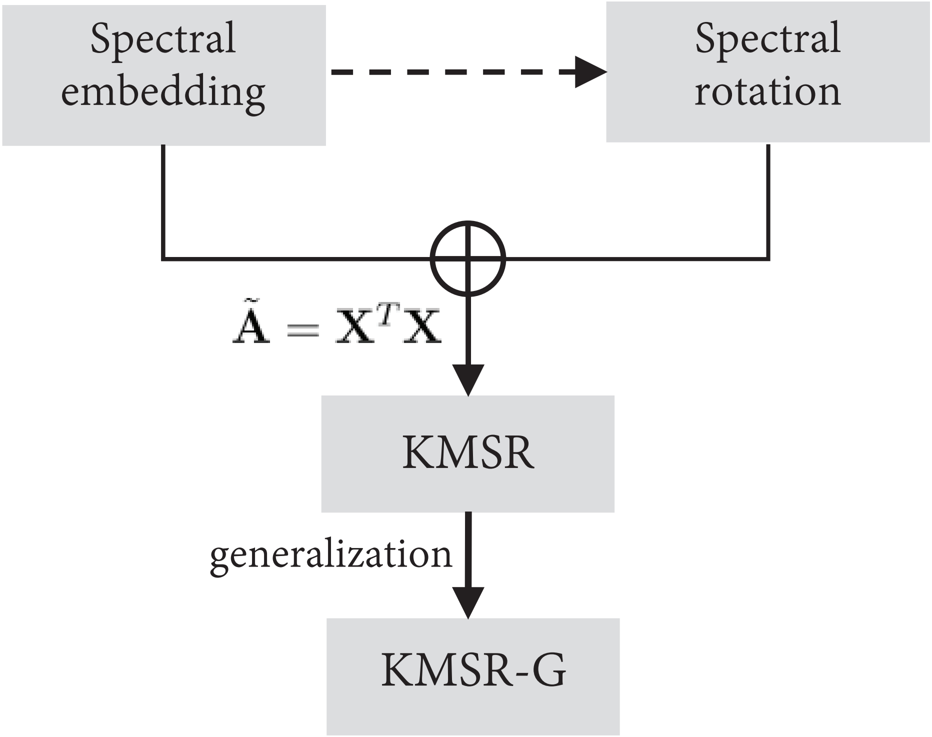

Figure 1 **The framework of the proposed models, KMSR and KMSR-G.**
Full-size DOI: 10.7717/peerjcs.450/fig-1

Figure 1 intuitively shows the framework of our proposed models from which we can observe that KMSR jointly performs spectral embedding and rotation on a specified graph. Further, KMSR-G is a generalized model to accommodate other advanced graphs, leading to better clustering performance.

Model optimization

The only difference between the model objectives of KMSR in Eq. (9) and KMSR-G Eq. (11) is the graph affinity matrix; therefore, they share the identical optimization procedure. Below taking the KMSR-G as an example, we show its detailed optimization steps based on the alternating framework (*Tang et al., 2020*). That is, we update one variable by fixing the others.

■ Update **Q** with **Y** and **F** fixed. The sub-objective associated with **Q** is

$$\min_{\mathbf{Q}} \|\mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-\frac{1}{2}} - \mathbf{F}\mathbf{Q}\|_2^2, \quad s.t. \mathbf{Q}^T\mathbf{Q} = \mathbf{I}_c, \mathbf{Q} \in \mathbb{R}^{c\times c}, \tag{12}$$

which can be further reformulated into

$$\max_{\mathbf{Q}} \mathrm{Tr}((\mathbf{Y}^T\mathbf{Y})^{-\frac{1}{2}}\mathbf{Y}^T\mathbf{F}\mathbf{Q}), \quad s.t. \mathbf{Q}^T\mathbf{Q} = \mathbf{I}_c, \mathbf{Q} \in \mathbb{R}^{c\times c}. \tag{13}$$

By denoting $\mathbf{M}_2 \triangleq (\mathbf{Y}^T\mathbf{Y})^{-\frac{1}{2}}\mathbf{Y}^T\mathbf{F}$, objective Eq. (13) can be converted to

$$\max_{\mathbf{Q}} \mathrm{Tr}(\mathbf{M}_2\mathbf{Q}), \quad s.t. \mathbf{Q}^T\mathbf{Q} = \mathbf{I}_c, \mathbf{Q} \in \mathbb{R}^{c\times c} \tag{14}$$

Suppose that the singular value decomposition of $\mathbf{M}_2$ is $\mathbf{M}_2 = \mathbf{U\Sigma V}^T$ and then we have

$$\mathrm{Tr}(\mathbf{M}_2\mathbf{Q}) = \mathrm{Tr}(\mathbf{QU\Sigma V}^T) = \mathrm{Tr}(\mathbf{\Sigma E}) = \sum_{i=1}^{c} \lambda_{ii} e_{ii}, \tag{15}$$

where $\mathbf{E} = \mathbf{V}^T\mathbf{QU}$ with λ_{ii} and e_{ii} as the (i, i)-th elements of matrix $\mathbf{\Sigma}$ and $\mathbf{E}$, respectively.

Since $\mathbf{E}^T\mathbf{E} = \mathbf{U}^T\mathbf{Q}^T\mathbf{V}\mathbf{V}^T\mathbf{QU} = \mathbf{I}_c$, i.e., $\sum_{j=1}^{c} e_{ji}^2 = 1$, we know $e_{ii} \leq 1 \ (1 \leq i \leq c)$. Meanwhile, λ_{ii} is non-negative since it is a singular value. Therefore, we have $\mathrm{Tr}(\mathbf{M}_2\mathbf{Q}) = \sum_{i=1}^{c} \lambda_{ii} e_{ii} \leq \sum_{i=1}^{c} \lambda_{ii}$, and the equality holds when $e_{ii} = 1 \ (1 \leq i \leq c)$. That is to say, $\mathrm{Tr}(\mathbf{M}_2\mathbf{Q})$ reaches its maximum when $\mathbf{E} = \mathbf{I}_c = \mathbf{V}^T\mathbf{QU}$. Then we obtain the optimal solution of $\mathbf{Q}$ as

$$\mathbf{Q} = \mathbf{V}\,\mathbf{U}^T. \tag{16}$$

■ Update $\mathbf{F}$ with $\mathbf{Y}$ and $\mathbf{Q}$ fixed. The sub-objective associated with $\mathbf{F}$ is

$$\min_{\mathbf{F}} -\mathrm{Tr}(\mathbf{F}^T\tilde{\mathbf{A}}\mathbf{F}) + \lambda \mathrm{Tr}(\mathbf{FQQ}^T\mathbf{F}^T) - 2\lambda \mathrm{Tr}(\mathbf{F}^T\mathbf{M}_1\mathbf{Q}^T), \quad s.t. \mathbf{F}^T\mathbf{F} = \mathbf{I}_c, \mathbf{F} \in \mathbb{R}^{n\times c}, \tag{17}$$

where symmetric matric $\tilde{\mathbf{A}} \in \mathbb{R}^{n\times n}$ and $\mathbf{M}_1 \triangleq \mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-\frac{1}{2}} \in \mathbb{R}^{n\times c}$. Since $\mathbf{Q} \in \mathbb{R}^{c\times c}$ is a square matrix and $\mathbf{F}^T\mathbf{F} = \mathbf{I}_c$, the second term in objective Eq. (17) is a constant and then we can get the simplified version of objective Eq. (17) as

$$\min_{\mathbf{F}} -\mathrm{Tr}(\mathbf{F}^T\tilde{\mathbf{A}}\mathbf{F}) - 2\lambda \mathrm{Tr}(\mathbf{F}^T\mathbf{M}_1\mathbf{Q}^T), \quad s.t. \mathbf{F}^T\mathbf{F} = \mathbf{I}_c, \mathbf{F} \in \mathbb{R}^{n\times c}. \tag{18}$$

Denoting $\mathbf{B} \triangleq \mathbf{M}_1\mathbf{Q}^T$, we have

$$\max_{\mathbf{F}} \mathrm{Tr}(\mathbf{F}^T\tilde{\mathbf{A}}\mathbf{F}) + 2\lambda \mathrm{Tr}(\mathbf{F}^T\mathbf{B}), \quad s.t. \mathbf{F}^T\mathbf{F} = \mathbf{I}_c, \mathbf{F} \in \mathbb{R}^{n\times c}. \tag{19}$$

The corresponding Lagrangian function of problem Eq. (19) is

$$\mathcal{L}(\mathbf{F}, \mathbf{\Lambda}) = \mathrm{Tr}(\mathbf{F}^T\tilde{\mathbf{A}}\mathbf{F}) + 2\lambda \mathrm{Tr}(\mathbf{F}^T\mathbf{B}) - \mathrm{Tr}(\mathbf{\Lambda}(\mathbf{F}^T\mathbf{F} - \mathbf{I}_c)), \tag{20}$$

where $\mathbf{\Lambda}$ is a Lagrangian multiplier in matrix form. Then we can obtain the KKT condition as

$$\frac{\partial \mathcal{L}}{\partial \mathbf{F}} = 2\tilde{\mathbf{A}}\mathbf{F} + 2\lambda\mathbf{B} - 2\mathbf{F\Lambda} = \mathbf{0} \tag{21}$$

which is difficult to solve directly.

Essentially, problem Eq. (19) is a relaxed form of quadratic optimization problem on the Stiefel manifold (QPSM). In optimization theory, the standard form of QPSM is $\min_{\mathbf{P}^T\mathbf{P}=\mathbf{I}_k} \mathrm{Tr}(\mathbf{P}^T\mathbf{HP} - 2\mathbf{P}^T\mathbf{K})$, where $\mathbf{P} \in \mathbb{R}^{m\times k}$, $\mathbf{K} \in \mathbb{R}^{m\times k}$, and the symmetric matrix $\mathbf{H} \in \mathbb{R}^{m\times m}$. This objective can be relaxed into $\max_{\mathbf{P}^T\mathbf{P}=\mathbf{I}_k} \mathrm{Tr}(\mathbf{P}^T\tilde{\mathbf{H}}\mathbf{P}) + 2\mathrm{Tr}(\mathbf{P}^T\mathbf{K})$ by introducing $\tilde{\mathbf{H}} = \alpha\mathbf{I}_m - \mathbf{H} \in \mathbb{R}^{m\times m}$, which is equivalent to Eq. (19). Inspired by the work (*Nie, Zhang & Li, 2017*), we employ the generalized power iteration (GPI) method to optimize Eq. (19) and summarize the detailed procedure in Algorithm 1.

Algorithm 1 The GPI-based optimization to problem (19)

```
Input: symmetric matrix $\tilde{\mathbf{A}} \in \mathbb{R}^{n\times n}$, orthogonal matrix $\mathbf{Q} \in \mathbb{R}^{c\times c}$, indicator matrix $\mathbf{Y} \in \mathbb{B}^{n\times c}$, regulation parameter $\boldsymbol{\lambda}$;
Output: the scaled indicator matrix $\mathbf{F}$.
Initialize $\mathbf{F} \in \mathbb{R}^{n\times c}$ such that $\mathbf{F}^T\mathbf{F} = \mathbf{I}_c$ ;
while not converged do
    Update $\mathbf{M}_F = 2\tilde{\mathbf{A}}\mathbf{F} + 2\lambda\mathbf{B}$, where $\mathbf{B} \triangleq \mathbf{M_1}\mathbf{Q}^T$ and $\mathbf{M_1} \triangleq \mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-\frac{1}{2}}$;
    Calculate the compact SVD of $\mathbf{M}_F$ as $\mathbf{M}_F = \mathbf{USV}^T$, where $\mathbf{U} \in \mathbb{R}^{c\times c}$, $\mathbf{S} \in \mathbb{R}^{n\times c}$ and $\mathbf{V} \in \mathbb{R}^{c\times c}$;
    Update $\mathbf{F} = \mathbf{UV}^T$;
end while
```

■ Update **Y** with **F** and **Q** fixed. Similar to the optimization process of updating **Q**, the sub-objective associated with **Y** is

$$\min_{\mathbf{Y}} \|\mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-\frac{1}{2}} - \mathbf{FQ}\|_2^2, \quad s.t.\mathbf{Y}\,\mathbf{1}_c = \mathbf{1}_n, \mathbf{Y} \in \mathbb{B}^{n\times c}. \tag{22}$$

Denote $\mathbf{G} \triangleq \mathbf{FQ}$ and then optimizing Eq. (22) is equivalent to optimizing the following one

$$\max_{\mathbf{Y}} \mathrm{Tr}(\mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-\frac{1}{2}}\mathbf{G}^T), \quad s.t.\mathbf{Y}\,\mathbf{1}_c = \mathbf{1}_n, \mathbf{Y} \in \mathbb{B}^{n\times c}. \tag{23}$$

Motivated by *Chen et al. (2018)*, objective Eq. (23) can be represented as

$$\max_{\mathbf{Y}\mathbf{1}_c=\mathbf{1}_n, \mathbf{Y}\in\mathbb{B}^{n\times c}} \sum_{j=1}^{c} \frac{\mathbf{y}_j{}^T\mathbf{g}_j}{\sqrt{\mathbf{y}_j{}^T\mathbf{y}_j}}. \tag{24}$$

Since $\sqrt{\mathbf{y}_j^T\mathbf{y}_j}$ involves all rows of **Y**, we can sove **Y** row-wisely; that is, we can update one row of **Y** by fixing the others as constants. Suppose we have obtained the optimal solution $\bar{\mathbf{Y}}$ in the last iteration and the corresponding objective function values is $\mathcal{J}^{old}(\bar{\mathbf{Y}})$. The elements of each row vector $\mathbf{y}^i$ are composed of 1 or 0 where the unique 1 indicates the cluster membership of *i*th data point. To solve the *i*th row $\mathbf{y}^i$, we only need to consider the increment of the objective function value from $y_{ij} = 0$ to $y_{ij} = 1$. The increment as can be calculated as follows

$$s_{ij} = \frac{\bar{\mathbf{y}}_j{}^T\mathbf{g}_j \; + g_{ij}(1-\bar{y}_{ij})}{\sqrt{\bar{\mathbf{y}}_j{}^T\bar{\mathbf{y}}_j \; + (1-\bar{y}_{ij})}} - \frac{\bar{\mathbf{y}}_j{}^T\mathbf{g}_j \; - \bar{y}_{ij}g_{ij}}{\sqrt{\bar{\mathbf{y}}_j{}^T\bar{\mathbf{y}}_j \; - \bar{y}_{ij}}}, \tag{25}$$

whose graphical illustration is given in Fig. 2.

Then it can be verified that the optimal solution of $\mathbf{y}^i$ is

$$y_{ij} = \langle j = \arg\max_{l' \in \,[1,c]} s_{il'} \rangle \tag{26}$$

where $\langle\cdot\rangle$ is 1 if the argument is true or 0 otherwise and s_{ij} is defined by Eq. (25).

As a whole, we summarize the complete procedure of solving the objective function Eq. (11) of KMSR-G in Algorithm 2.

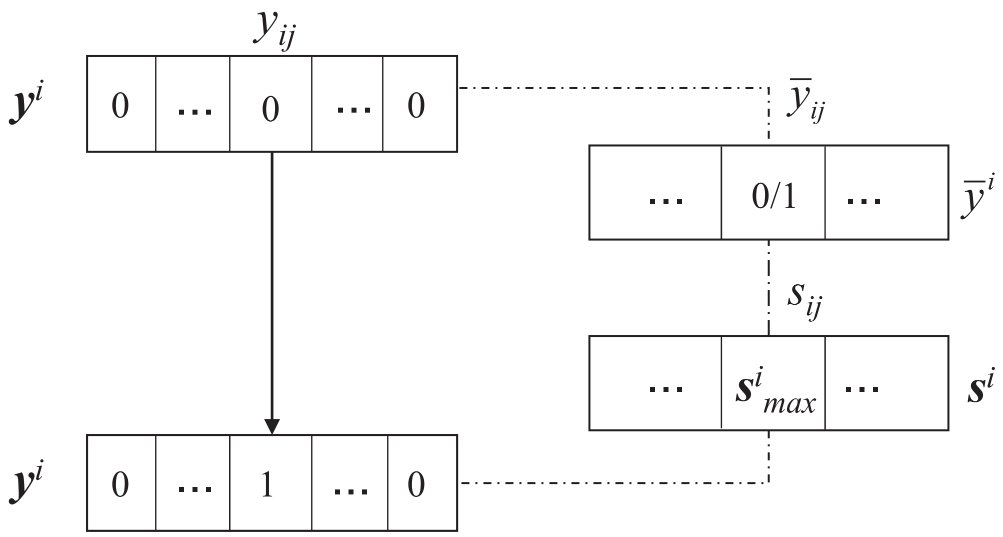

Figure 2 **The illustration procedure of updating Y (*Chen et al., 2017*).**
Full-size DOI: 10.7717/peerjcs.450/fig-2

Algorithm 2 The optimization procedure to KMSR-G objective function in (11).

```
Input: Data matrix $\mathbf{X} \in \mathbb{R}^{d \times n}$, the number of clusters $c$, and the regularization parameter $\lambda$;
Output: The binary cluster indicator matrix $\mathbf{Y} \in \mathbb{B}^{n \times c}$.
Construct graph similarity matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ and its normalized version $\tilde{\mathbf{A}} = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$;
Compute the normalized graph Laplacian matrix $\mathbf{L}_n = \mathbf{I} - \tilde{\mathbf{A}}$, where $\mathbf{D} \in \mathbb{R}^{n \times n}$ is a diagonal degree matrix with its $i$-th diagonal element $d_{ii} = \sum_{j=1}^{n} a_{ij}$;
Form the matrix $\mathbf{F}^*$ by stacking the eigenvectors of $\mathbf{L}_n$ corresponding to its $c$ smallest eigenvalues;
Initialize $\mathbf{Y}$ according to $\mathbf{Y}^* = diag(\mathbf{F}^*\mathbf{F}^{*T})^{-\frac{1}{2}}\mathbf{F}^*$ and $y_{ij} = \langle j = \text{argmax}_{j' \in [1,c]} y^*_{ij'} \rangle$;
Initialize $\mathbf{Q}$ randomly such that $\mathbf{Q}^T\mathbf{Q} = \mathbf{I}_c$;
while problem (11) does not converge do
    Update $\mathbf{F}$ according to Algorithm 1;
    Update $\mathbf{Q} = \mathbf{V}\mathbf{U}^T$, where the compact SVD of $\mathbf{M_2} = (\mathbf{Y}^T\mathbf{Y})^{-\frac{1}{2}}\mathbf{Y}^T\mathbf{F}$ is $\mathbf{U\Sigma V}^T$;
    Calculate $\mathbf{G} = \mathbf{FQ}$;
    while problem (24) does not converge do
        Update $\mathbf{Y}$ according to (26);
    end while
end while
```

Model complexity and convergence analysis

In terms of the computational complexity, if ignoring the special process in graph construction in KMSR-G, KMSR and KMSR-G share similar complexities because they involve the same optimization procedure. The below complexity analysis is based on Algorithm 2.

- *Updating the variable* **F**. We use the generalized power iteration method to update **F** in Eq. (19). According to the analysis in *Nie, Zhang & Li (2017)*, the complexity of updating **F** is $\mathcal{O}(n^2c)$.
- *Updating the variable* **Q**. The complexity of updating **Q** mainly comes from the singular value decomposition of $\mathbf{M}_2 \in \mathbb{R}^{c\times c}$, which has the complexity of $\mathcal{O}(c^3)$.
- *Updating the variable* **Y**. We need $\mathcal{O}(nc)$ time to obtain **Y** because we deal with **Y** row by row. What's more, $\overline{\mathbf{y}_j}^T\overline{\mathbf{y}_j}$ and $\overline{\mathbf{y}_j}^T\mathbf{g}_j$ can be calculated before solving **Y** and updated after solving $\mathbf{y}^i$ according to Eq. (26).

Assuming that T_1 is the maximum number of iterations for the KMSR-G, r_1 and r_2 are the average numbers of iterations to update **F** and **Y** respectively, the overall computational complexity of KMSR-G is $\mathcal{O}(T_1(n^2cr_1+c^3+ncr_2))$. In general cases, we have $c < T_1 << n$. In comparison with spectral clustering methods which usually have the time complexity of $\mathcal{O}(n^3)$ (n is the number of samples), we can easily find that KMSR and KMSR-G have a lower computational complexity in dealing with large-sized data sets.

Obviously, KMSR and KMSR-G have similar convergence properties due to the identical optimization procedure. Here we also give the analysis based on Algorithm 2. When solving the variable **F**, the GPI method is utilized whose optimization procedure is summarized in Algorithm 1. According to the appendix of *Nie, Zhang & Li (2017)*, we know that the GPI method converges to a global minimum of the quadratic problem on the Stiefel manifold, which guarantees the convergence of Algorithm 1 in updating **F**. When updating the variable **Q**, the analytical solution to **Q** can be obtained based on the singular value decomposition. For the variable **Y**, we propose to optimize it in a row-by-row manner according to Eq. (26) because determining the membership of each sample is independent. So we can convert the updating of **Y** into n independent subproblems. Further, since each row of **Y** is a binary vector in one-hot encoding, there are finite candidate solutions for each subproblem. Therefore, it can be proved that each subproblem must have an optimal solution, which guarantees the convergence of the updating of **Y**. In total, we can come to a conclusion that the optimization of KMSR-G is expected to converge in terms of iterations.

Discussions

We can find that the only difference between KMSR and KMSR-G is the different ways to represent the similarity matrix (also called the affinity matrix) in respective objective functions. If the normalized graph affinity matrix $\tilde{\mathbf{A}}$ is represented as $\mathbf{X}^T\mathbf{X}$, KMSR-G will degenerate to KMSR. It is obvious that the clustering performance heavily depends on the quality of the input data graph in graph-based clustering. In the section of experiments, we will adopt one rule-based method ('Heatkernel' weighting scheme) and two learning-based methods (CAN and PCAN (*Nie, Wang & Huang, 2014*)) to obtain three different affinity matrices. Then we will see how much the graph quality influences the performance of the KMSR-G.

Therefore, there are two proposed models KMSR and KMSR-G in this paper and the latter can be seen as the generalization of the former. As a summary, below we summarize the main contributions of this paper.

- We propose a novel *K*means formulation (termed KMSR) by exploring the underlying equivalence between the *K*means clustering and the spectral clustering, which is finally achieved by jointly performing the spectral embedding and rotation. Mathematically, the objective of KMSR consists of two items, which respectively aim to calculate the scaled cluster indicator matrix and perform discretization to obtain the final discrete cluster indicator matrix. When compared with *K*means clustering, the randomness would be effectively alleviated in KMSR because it jointly searches for an optimal rotator in discretization process.
- By investigating the connection between the graph affinity matrices respectively employed in KMSR and spectral clustering, we make generalization on the KMSR model to make it accommodate more advanced graphs, and finally formulate the KMSR-G model. KMSR-G is also a unified model for jointly completing the spectral embedding and rotation steps, which is often expected to obtain superior performance to KMSR.
- We propose an efficient algorithm to optimize the objective function of KMSR-G (also KMSR since they share the same optimization procedure). In the iterative procedure, there are three blocks respectively corresponding to the three variables (i.e., **F**, **Q** and **Y**) involved in KMSR-G. They are co-optimized toward the optimum. Besides, we provide detailed computational complexity and convergence analysis to the optimization algorithm.
- To evaluate the performance of KMSR and KMSR-G in data clustering, we conduct extensive experiments on twelve representative benchmark data sets. The experimental results show that both KMSR abd KMSR-G perform better than the closely related counterparts.

EXPERIMENTS

In this section, we conduct experiments on twelve representative benchmark data sets to evaluate the performance of the proposed KMSR and KMSR-G models in data clustering.

Evaluation metrics

To evaluate the clustering results, we compare the obtained label of each sample with the label provided by the data set. We use three popular metrics, i.e., *Accuracy* (Acc) , *Normalized Mutual Information* (NMI) and *Purity* (*Huang, Nie & Huang, 2015*) to measure the clustering performance of different models. Below we give the definition of these three metrics in turn.

Given a data point $\mathbf{x}_i$, we use r_i to denote the obtained cluster label and s_i to denote the ground-truth label provided by the data set. Then Acc is defined as

$$\mathrm{Acc} = \frac{1}{n}\sum_{i=1}^{n}\delta(s_i, \mathrm{map}(r_i)), \tag{27}$$

where n is the sample size, $\delta(x,y)$ is the indicator function that equals to one if $x=y$ and zero otherwise. $\mathrm{map}(r_i)$ is the permutation mapping function which maps each cluster label r_i to the equivalent class label from the data set.

Let C denote the set of clusters obtained from the ground truth and C' denote the set of the clusters obtained from the given model. Then mutual information $\mathrm{MI}(C,C')$ is defined as

$$\mathrm{MI}(C,C') = \sum_{c_i \in C, c_j' \in C'} p(c_i,c_j') \cdot \log \frac{p(c_i,c_j')}{p(c_i)p(c_j')}, \tag{28}$$

where $p(c_i)$ and $p(c_j')$ denote the probabilities that a sample arbitrarily selected from the data set belongs to the clusters c_i and c_j', respectively. Besides, $p(c_i,c_j')$ is the joint probability that the selected sample belongs to the clusters c_i and c_j' simultaneously. So the NMI is given as follows

$$\mathrm{NMI}(C,C') = \frac{\mathrm{MI}(C,C')}{\max(H(C),H(C'))}, \tag{29}$$

where $H(C)$ and $H(C')$ denote the entropies of C and C' respectively.

To compute the purity, each cluster is assigned to the class which is most frequent in the cluster, and then the accuracy of this assignment is measured by counting the number of correctly assigned documents and dividing by n. Then the clustering Purity metric is estimated by

$$\mathrm{Purity} = \frac{1}{n}\sum_{i=1}^{c} \max_j(n_i^j) \tag{30}$$

where c is the number of the clusters and n is the total number of the data points, n_i^j is the number of ith input class that is assigned to the jth cluster.

It is easy to check that all Acc, NMI and Purity metrics range from zero to one and a higher value indicates a better clustering result.

Data sets and experimental settings

Twelve real-world data sets were used in the following experiments including nine image data sets (COIL20, umist, AT&T, YaleB, Yale, PIE, AR, MNIST and jaffe) and three non-image data sets (ecoli, abalone and scale). Their basic characteristics of the sample size, dimensionality, the number of clusters were summarized in Table 1.

The following experiments can be divided into two parts. In the former part, we aim to demonstrate the effectiveness of the KMSR model in comparison with the traditional one; In the latter part, we want to evaluate the effectiveness of the generalized KMSR-G model and also provide some insights to the influence of graph quality on the clustering performance.

PART 1. Experimental settings to evaluate the performance of the KMSR model. To investigate the effectiveness of the KMSR model, we perform the pairwise comparison between the KMSR and the traditional Kmeans clustering. Since the Kmeans clustering is sensitive to the initialization, we independently repeat it 50 times. For our KMSR model,

Table 1 **The basic characteristics the twelve data sets used in the experiments.**

Data Sets	# Samples	# Dimensions	# Clusters
ecoli	327	7	5
abalone	4177	8	3
scale	625	4	3
COIL20	1440	1024	20
umist	575	644	20
AT&T	400	189	40
YaleB	2414	1024	38
Yale	165	105	15
PIE	1428	1024	68
AR	1200	261	100
MNIST	1000	784	10
jaffe	212	177	7

we repeat it 20 times. The number of clusters is set as the ground-truth. Since there is a free regularization parameter in the proposed KMSR model, we tune it from candidate values $\{10^{-3}, 10^{-2}, \ldots, 10^{3}\}$ to let it achieve the best results.

PART 2. Experimental settings to evaluate the performance of the KMSR-G model. First, we compare the generalized model KMSR-G with two closely related spectral clustering algorithms, NCut and RCut, to evaluate its effectiveness. Two commonly used post-processing methods (i.e., *K* means and spectral rotation) are adopted in spectral clustering; therefore, we implement this two versions of NCut respectively named as NCut+KM and NCut+SR. Similarly, we have two corresponding versions of RCut, RCut+KM and RCut+SR. The affinity matrix referred in these models is constructed by the 'Heatkernel' function in which the number of the nearest neighbors is set as five, and the bandwidth parameter is set as one. Second, by taking the two more advanced learning-based graph affinity matrices, CAN and PCAN, as the input graphs to KMSR-G, we respectively obtain two variant models termed KMSR-GC and KMSR-GPC. The neighborhood parameters k required in CAN and PCAN are tuned from $\{5, 10, 15, 20\}$. The CAN and PCAN method are repeated only once since their clustering results are stable (*Nie, Wang & Huang, 2014*). Both KMSR-GC and KMSR-GPC are repeated 20 times. We use CAN and PCAN as graph learning methods in our experiments for two reasons. One is that both are joint models for graph construction and scaled cluster indicator matrix learning, leading to superior graph quality. The other is that they can adaptively determine the number of neighbors in graph construction.

For all the above mentioned models, the number of clusters is set as the ground-truth in each data set and the clustering performance is evaluated based on the metrics of Acc, NMI and Purity. Besides, we respectively set the maximum numbers of iterations in updating **F** and **Y** in our proposed models to be 50 and 10. For the free regularization parameters in related models, they are tuned from $\{10^{-3}, 10^{-2}, \ldots, 10^{2}, 10^{3}\}$ to let the models achieve their best results. The average clustering results and standard deviations are reported for comparison.

Table 2 **Clustering performance (%) of *K*means and KMSR on the benchmark data sets.**

	Acc		NMI		Purity	
Data Sets	***K*means**	**KMSR**	***K*means**	**KMSR**	***K*means**	**KMSR**
ecoli	65.50 ± 6.85	**84.56 ± 0.22**	58.70 ± 2.27	**61.47 ± 0.35**	78.46 ± 1.50	**84.56 ± 0.22**
abalone	51.55 ± 0.39	**53.63 ± 0.37**	13.23 ± 0.63	**15.64 ± 0.10**	53.50 ± 0.03	**54.20 ± 0.10**
scale	55.31 ± 5.66	**64.93 ± 0.51**	17.42 ± 8.74	**34.65 ± 1.34**	69.72 ± 5.33	**77.95 ± 1.05**
COIL20	56.06 ± 4.94	**72.81 ± 0.05**	70.51 ± 2.18	**80.26 ± 0.22**	59.64 ± 4.34	**74.55 ± 0.34**
umist	39.51 ± 2.07	**51.04 ± 0.12**	58.33 ± 1.80	**65.09 ± 0.20**	46.51 ± 1.54	**57.39 ± 0.49**
AT&T	52.94 ± 4.21	**63.10 ± 1.80**	73.06 ± 2.51	**78.34 ± 0.50**	61.25 ± 3.41	**67.85 ± 2.00**
YaleB	8.49 ± 0.70	**23.14 ± 0.62**	10.46 ± 1.00	**41.70 ± 0.71**	9.22 ± 0.73	**24.25 ± 0.79**
Yale	39.09 ± 4.61	**44.48 ± 2.04**	45.19 ± 3.61	**49.13 ± 1.60**	41.36 ± 3.92	**46.06 ± 1.29**
PIE	33.56 ± 1.97	**92.38 ± 0.91**	66.40 ± 0.92	**98.13 ± 0.22**	39.22 ± 1.27	**94.45 ± 0.57**
AR	15.25 ± 0.31	**33.48 ± 1.06**	48.43 ± 0.65	**62.38 ± 0.86**	16.18 ± 0.39	**37.40 ± 1.04**
MNIST	52.04 ± 3.16	**56.17 ± 0.31**	49.83 ± 1.93	**50.83 ± 0.00**	55.83 ± 2.63	**57.50 ± 0.00**
jaffe	27.88 ± 2.92	**34.91 ± 1.33**	12.51 ± 3.02	**17.25 ± 1.35**	29.62 ± 3.38	**35.94 ± 1.22**

Experimental results and analysis

Here we show the experimental results based on which we provide the corresponding analysis.

PART 1. The experimental results of the pairwise comparison between *K*means and KMSR are shown in Table 2, which includes the average results and standard deviations over multiple runs. From this table, we can find that the best results highlighted in boldface are all from the proposed KMSR model. Therefore, we can conclude that KMSR significantly outperforms the *K*means on all the used data sets in terms of all the three clustering performance evaluation metrics, which indicates the effectiveness of jointly performing spectral embedding and rotation.

Besides the mean values, we can observe that the standard deviations on all the data sets corresponding to the KMSR is much smaller than those of *K*means, which are more explicitly shown by the statistical box digrams in Fig. 3. This means that KMSR is superior to *K*means on the model stability. We think that this improvement comes from the optimization of the orthogonal and normalized rotation matrix instead of the random initialization of cluster centroids in *K*means clustering.

PART 2. By constructing the graph affinity matrix in the 'Heatkernel' scheme, we show the experimental results of comparing KMSR-G with the four closely related models, NCut+KM, NCut+SR, RCut+KM and RCut+SR, in Table 3 where the best results are highlighted in boldface. It is obvious that KMSR-G obtained better clustering performance than the other models in most cases. For example, KMSR-G performs pretty well on the ecoli, COIL20, AT&T, PIE and MNIST data sets, which respectively obtains the improvements of 7.78%, 3.76%, 2.7%, 12.83%, and 4.05% in comparison with the second-best method in terms of the Acc metric.

Besides the theoretical analysis on the convergence of KMSR-G in 'Model complexity and convergence analysis', , we empirically show the decreasing of its objective function values on the six data sets of abalone, scale, umist, YaleB, PIE and jaffe in Fig. 4. All the

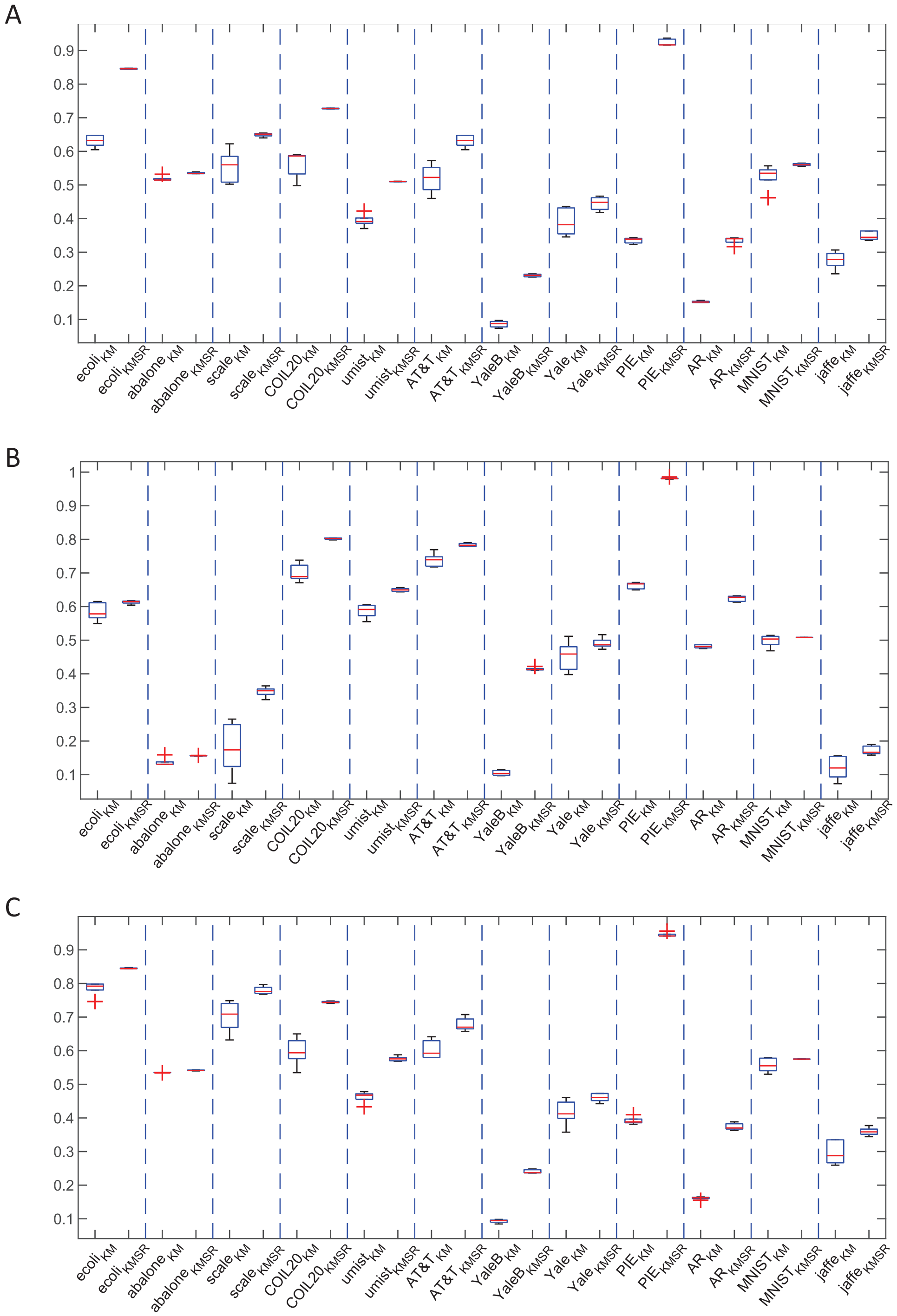

Figure 3 **Clustering results obtained by *K* means and KMSR in statistical box diagrams.** (A) Acc. (B) NMI. (C) Purity.

Full-size DOI: 10.7717/peerjcs.450/fig-3

Table 3 **Clustering performance (%) of KMSR-G and related models on twelve data sets.**

Data Sets	NCut+KM	NCut+SR	RCut+KM	RCut+SR	KMSR-G
			Acc		
ecoli	73.70 ± 0.55	75.26 ± 1.04	75.15 ± 5.25	77.84 ± 1.49	**85.63 ± 0.01**
abalone	49.42 ± 3.85	50.10 ± 2.68	48.87 ± 4.30	50.97 ± 0.00	**51.13 ± 0.22**
scale	65.61 ± 0.31	66.15 ± 0.50	63.37 ± 3.08	65.17 ± 1.61	**66.72 ± 0.45**
COIL20	72.46 ± 4.80	76.58 ± 5.16	71.45 ± 4.76	77.77 ± 3.31	**81.53 ± 0.69**
umist	50.77 ± 2.10	53.97 ± 1.98	51.32 ± 3.86	55.29 ± 1.56	**56.43 ± 1.35**
AT&T	62.70 ± 2.15	66.30 ± 2.53	62.38 ± 1.25	63.46 ± 1.01	**69.00 ± 0.71**
YaleB	32.69 ± 1.19	43.47 ± 0.33	33.62 ± 1.31	**43.57 ± 0.06**	37.76 ± 0.03
Yale	42.55 ± 0.66	42.61 ± 1.73	41.64 ± 1.56	43.42 ± 1.60	**44.24 ± 0.86**
PIE	82.56 ± 3.13	84.93 ± 3.16	81.55 ± 2.07	84.82 ± 3.96	**97.76 ± 0.01**
AR	16.65 ± 0.23	16.80 ± 0.15	16.56 ± 0.26	16.92 ± 0.11	**17.04 ± 0.06**
MNIST	49.29 ± 3.46	50.01 ± 1.45	49.70 ± 2.33	50.75 ± 1.24	**54.80 ± 0.00**
jaffe	24.43 ± 0.60	25.68 ± 0.24	24.62 ± 1.12	25.94 ± 0.48	**25.94 ± 0.00**
			NMI		
ecoli	55.94 ± 0.00	56.75 ± 0.47	57.86 ± 1.31	58.37 ± 0.19	**67.55 ± 0.24**
abalone	10.05 ± 4.54	11.42 ± 3.06	9.47 ± 4.54	12.41 ± 0.00	**12.41 ± 0.00**
scale	34.23 ± 0.35	34.64 ± 0.46	32.26 ± 3.88	34.13 ± 1.75	**36.92 ± 0.00**
COIL20	84.50 ± 1.81	87.66 ± 2.39	84.91 ± 2.41	87.78 ± 2.43	**88.89 ± 0.23**
umist	71.14 ± 1.14	72.66 ± 1.17	70.37 ± 2.01	**73.60 ± 0.81**	73.17 ± 0.30
AT&T	79.04 ± 0.80	80.39 ± 1.20	78.95 ± 0.63	79.20 ± 0.55	**81.34 ± 0.20**
YaleB	43.10 ± 0.80	**51.26 ± 0.24**	41.41 ± 0.91	50.00 ± 0.05	44.37 ± 0.09
Yale	49.94 ± 0.73	50.29 ± 1.16	49.18 ± 1.27	50.95 ± 1.00	**51.71 ± 0.04**
PIE	94.67 ± 1.08	95.12 ± 1.10	93.97 ± 0.97	95.03 ± 1.41	**99.40 ± 0.01**
AR	51.84 ± 0.22	**52.87 ± 0.18**	50.59 ± 0.38	52.11 ± 0.28	52.12 ± 0.32
MNIST	53.78 ± 2.21	55.47 ± 0.95	52.68 ± 1.12	53.69 ± 1.11	**57.03 ± 0.06**
jaffe	8.71 ± 0.82	9.40 ± 0.22	9.48 ± 0.88	9.69 ± 0.40	**11.31 ± 0.66**
			Purity		
ecoli	74.68 ± 0.14	75.72 ± 0.61	77.91 ± 0.29	78.12 ± 0.29	**85.63 ± 0.01**
abalone	50.54 ± 4.42	51.93 ± 3.31	49.89 ± 4.90	53.00 ± 0.00	**53.00 ± 0.00**
scale	78.74 ± 0.38	78.91 ± 0.55	77.41 ± 1.35	78.17 ± 0.73	**79.76 ± 0.11**
COIL20	75.23 ± 4.13	81.32 ± 5.24	76.56 ± 4.05	82.67 ± 3.33	**84.20 ± 0.34**
umist	61.96 ± 2.12	63.31 ± 1.97	60.16 ± 3.21	64.33 ± 1.66	**65.04 ± 0.00**
AT&T	66.00 ± 1.79	68.53 ± 2.86	65.75 ± 1.32	67.90 ± 1.02	**72.63 ± 0.53**
YaleB	34.72 ± 1.03	45.28 ± 0.35	35.65 ± 1.04	**45.36 ± 0.07**	39.08 ± 0.03
Yale	42.91 ± 0.51	43.00 ± 1.14	42.52 ± 1.78	43.42 ± 1.60	**44.55 ± 1.29**
PIE	85.51 ± 2.82	85.53 ± 3.24	82.46 ± 2.53	85.46 ± 3.96	**98.39 ± 0.01**
AR	17.29 ± 0.23	17.32 ± 0.13	17.50 ± 0.25	17.56 ± 0.17	**17.71 ± 0.06**
MNIST	55.74 ± 2.62	58.00 ± 2.81	55.55 ± 1.70	57.55 ± 1.59	**61.95 ± 0.07**
jaffe	24.65 ± 0.88	25.68 ± 0.24	24.79 ± 1.36	26.44 ± 0.50	**27.83 ± 0.75**

results are obtained when the regulation parameter $\boldsymbol{\lambda}$ is set as 10^{-1}. We can find that KMSR-G has desirable convergence property and usually converges in a few iterations.

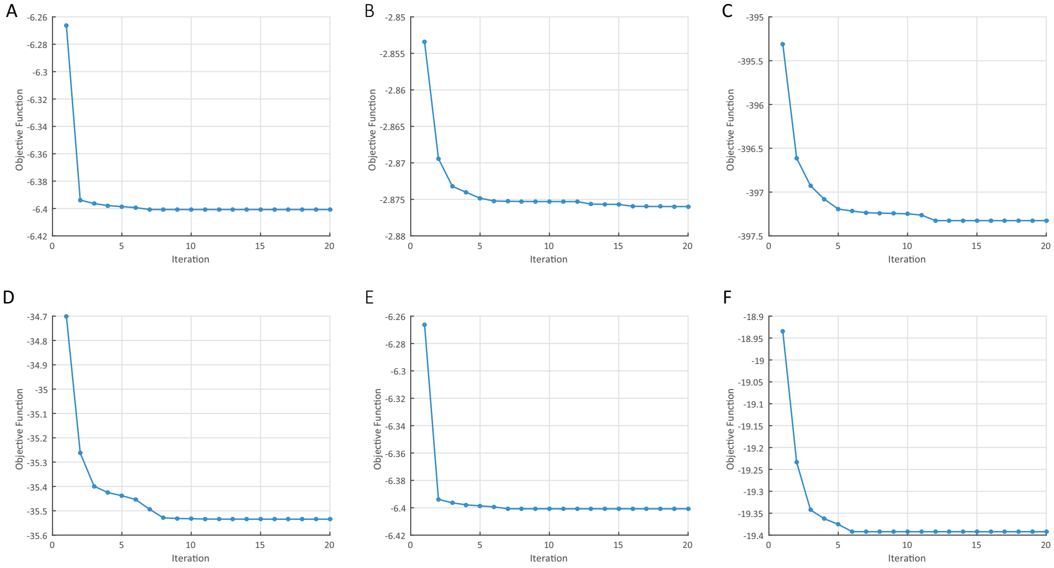

Figure 4 **The decreasing of the KMSR-G objective function values in terms of iterations ($\lambda = 0.1$).** (A) abalone. (B) scale. (C) PIE. (D) YaleB. (E) jaffe. (F) umist.

Full-size DOI: 10.7717/peerjcs.450/fig-4

In the description of experimental settings, we mentioned that the regulation parameter λ is tuned from candidate values $\{10^{-3}, 10^{-2}, \ldots, 10^{3}\}$. Here we explore the impact of such regulation parameter on the clustering performance of KMSR-G. We show the clustering accuracy of KMSR-G with the variation of parameter λ on the eight of our used data sets in Fig. 5. Generally, it depicts that KMSR-G prefers a small value of λ to achieve better clustering accuracy.

In the optimization of the binary cluster indicator matrix $\mathbf{Y}$, it is also in an iterative manner (i.e., the inner loop in Algorithm 2). Here taking the AT&T and COIL20 data sets for example, we show the number of iterations in updating $\mathbf{Y}$ in Fig. 6. Since the number of clusters is usually small in respective data sets, the updating of $\mathbf{Y}$ could be a very fast process, usually less than 10 iterations.

Besides the experiments on the rule-based graph (i.e., 'Heatkernel' function), we further try two learning-based graphs which are the CAN and PCAN (*Nie, Wang & Huang, 2014*). CAN can adaptively learn the graph affinity matrix from data by simultaneously considering the non-negativity, normalization and rank constraint (*Peng et al., 2020*) properties of a desirable graph. PCAN is its projected version, which takes the three-fold constraints into account in a subspace. We present the clustering results of CAN, KMSR-GC, PCAN and KMSR-GPC in Table 4. From the obtained results, we have the following two findings.

- By comparing the results in Tables 3 and 4, we can find that KMSR-G with CAN (or PCAN) obtained superior performance than that with 'Heatkernel' function in most cases. This indicates that the graph quality is the leading factor in graph-based clustering models. Even for one graph construction method, it functions differently on different data sets according to our experimental results.

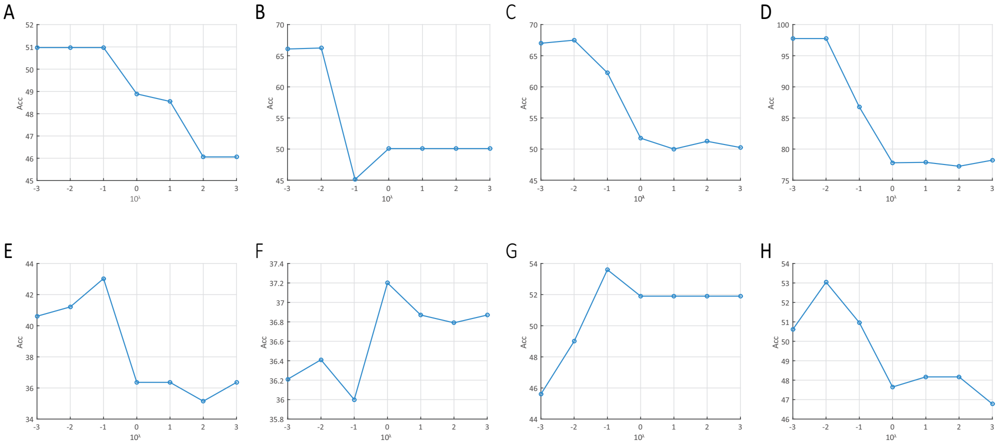

Figure 5 The accuracy of KMSR-G obtained in different settings of λ. (A) abalone. (B) scale. (C) AT&T. (D) PIE. (E) Yale. (F) YaleB. (G) MNIST. (H) umist.

Full-size DOI: 10.7717/peerjcs.450/fig-5

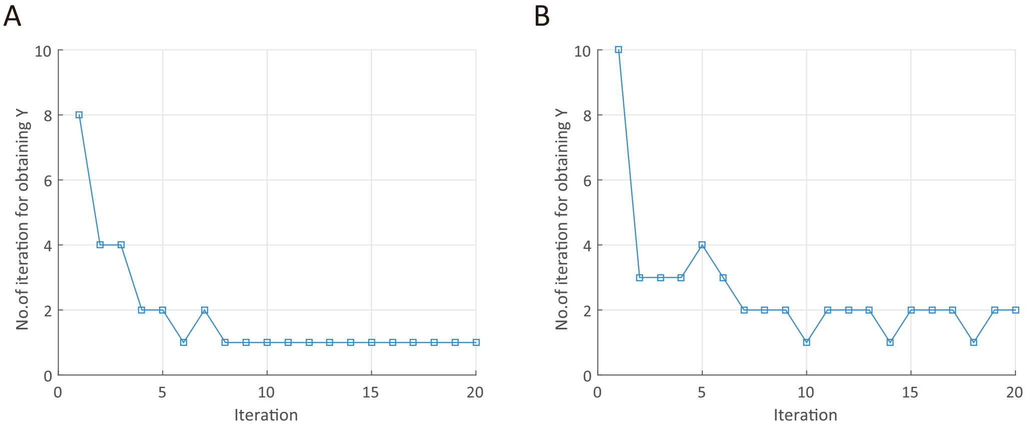

Figure 6 Number of iterations in updating Y on AT&T and COIL20 data sets. (A) AT&T. (B) COIL20.

Full-size DOI: 10.7717/peerjcs.450/fig-6

- We can generally declare that KMSR-GC is better than its baseline method CAN; and similarly, KMSR-GPC outperforms PCAN on all the data sets. Though CAN and PCAN jointly performs graph learning and clustering, the indicator matrices learned by them are still real-valued ones; therefore, the postprocessing step is necessary to make discretization. This limitation is avoided in our proposed KMSR-G model and thus improved performance is obtained.

CONCLUSION

In this paper, based on the connection between Kmeans clustering and spectral clustering, we proposed a new Kmeans formulation by jointly performing spectral embedding and spectral rotation. The formulated KMSR model can not only improve the clustering performance but also enhance the model stability in comparison with the traditional

Table 4 **Clustering performance (%) of KMSR-G based on learning-based graphs of CAN and PCAN.**

Data Sets	CAN	KMSR-GC	PCAN	KMSR-GPC
		Acc		
ecoli	81.96	**81.96 ± 0.00**	81.96	**81.96 ± 0.00**
abalone	50.97	**53.08 ± 0.00**	50.97	**51.59 ± 0.88**
scale	53.28	**63.20 ± 0.00**	66.72	**67.84 ± 0.01**
COIL20	83.96	**86.01 ± 0.05**	81.81	**81.81 ± 0.00**
umist	69.04	**71.13 ± 0.30**	54.78	**55.04 ± 0.37**
AT&T	55.25	**63.00 ± 1.77**	60.00	**61.50 ± 0.01**
YaleB	37.12	**37.12 ± 0.00**	38.69	**38.69 ± 0.00**
Yale	41.21	**42.42 ± 0.86**	40.00	**40.91 ± 0.43**
PIE	100.0	**100.0 ± 0.00**	100.0	**100.0 ± 0.00**
AR	13.67	**13.96 ± 0.06**	13.33	**13.63 ± 0.06**
MNIST	45.80	**49.37 ± 2.14**	45.80	**45.80 ± 0.00**
jaffe	22.64	**27.59 ± 0.33**	25.00	**26.18 ± 1.00**
		NMI		
ecoli	65.74	**65.74 ± 0.00**	66.32	**66.32 ± 0.00**
abalone	12.49	**12.52 ± 0.00**	12.41	**12.42 ± 0.00**
scale	17.80	**18.42 ± 0.00**	17.04	**17.17 ± 0.01**
COIL20	91.34	**92.49 ± 0.01**	89.60	**89.60 ± 0.00**
umist	81.72	**82.15 ± 0.60**	65.02	**65.02 ± 0.00**
AT&T	73.88	**78.82 ± 0.85**	74.13	**77.32 ± 0.22**
YaleB	39.98	**39.98 ± 0.00**	39.99	**40.22 ± 0.33**
Yale	44.95	**45.77 ± 0.20**	42.64	**43.06 ± 0.58**
PIE	100.0	**100.0 ± 0.00**	100.0	**100.0 ± 0.00**
AR	42.93	**43.04 ± 0.01**	35.88	**38.04 ± 0.02**
MNIST	49.87	**49.87 ± 0.00**	46.35	**46.46 ± 0.00**
jaffe	10.80	**11.28 ± 0.56**	11.10	**11.94 ± 0.83**
		Purity		
ecoli	82.87	**82.87 ± 0.00**	82.87	**82.87 ± 0.00**
abalone	53.08	**53.00 ± 0.00**	53.00	**53.00 ± 0.00**
scale	67.20	**67.20 ± 0.00**	70.08	**70.24 ± 0.01**
COIL20	87.22	**88.26 ± 0.00**	86.11	**86.11 ± 0.00**
umist	74.78	**76.00 ± 0.74**	62.78	**62.78 ± 0.00**
AT&T	64.25	**68.38 ± 1.24**	67.50	**68.25 ± 0.01**
YaleB	39.73	**39.73 ± 0.00**	40.56	**40.56 ± 0.00**
Yale	43.03	**44.85 ± 0.00**	42.42	**42.73 ± 0.43**
PIE	100.0	**100.0 ± 0.00**	100.0	**100.0 ± 0.00**
AR	16.83	**16.96 ± 0.06**	17.17	**17.25 ± 0.01**
MNIST	50.90	**50.90 ± 0.00**	51.10	**51.10 ± 0.00**
jaffe	23.11	**28.07 ± 0.33**	25.94	**26.89 ± 0.67**

K means clustering. Further, the KMSR model was generalized as KMSR-G which can take any pre-defined graph as input and output the final discrete cluster indicator matrix. An efficient method in coordinate blocking framework was designed to optimize the proposed KMSR (KMSR-G) model objective. Extensive experiments were conducted on representative data sets to show the effectiveness of the proposed models in data clustering. As our future work, we will consider unifying the three components of graph construction, spectral embedding and spectral rotation in graph-based clustering into a complete framework. That is, we will incorporate the graph learning process into the present KMSR-G model.

ADDITIONAL INFORMATION AND DECLARATIONS

Funding

This work was supported by the Natural Science Foundation of China (61971173, U1909202, U20B2074), the National Social Science Foundation of China (19ZDA348), the Zhejiang Provincial Natural Science Foundation of China (LY21F030005), the Fundamental Research Funds for the Provincial Universities of Zhejiang (GK209907299001-008), the China Postdoctoral Science Foundation (2017M620470), the Foundation of Zhejiang Provincial Education Department of China (Y202044319), the Key Laboratory of Advanced Perception and Intelligent Control of High-end Equipment of Ministry of Education (GDSC202015) and the Jiangsu Provincial Key Laboratory for Computer Information Processing Technology, Soochow University (KJS1841). There was no additional external funding received for this study. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Grant Disclosures

The following grant information was disclosed by the authors:
Natural Science Foundation of China: 61971173, U1909202, U20B2074.
National Social Science Foundation of China: 19ZDA348.
Zhejiang Provincial Natural Science Foundation of China: LY21F030005.
Fundamental Research Funds for the Provincial Universities of Zhejiang: GK209907299001-008.
China Postdoctoral Science Foundation: 2017M620470.
Foundation of Zhejiang Provincial Education Department of China: Y202044319.
Key Laboratory of Advanced Perception and Intelligent Control of High-end Equipment of Ministry of Education: GDSC202015.
Jiangsu Provincial Key Laboratory for Computer Information Processing Technology, Soochow University: KJS1841.

Competing Interests

The authors declare there are no competing interests.

Author Contributions

- Wenna Huang performed the experiments, performed the computation work, prepared figures and/or tables, and approved the final draft.
- Yong Peng conceived and designed the experiments, authored or reviewed drafts of the paper, and approved the final draft.
- Yuan Ge performed the experiments, analyzed the data, prepared figures and/or tables, and approved the final draft.
- Wanzeng Kong analyzed the data, authored or reviewed drafts of the paper, and approved the final draft.

Data Availability

The following information was supplied regarding data availability:

Data and code are available in the Supplementary Files and GitHub: https://github.com/finepix/KMSR-G.git.

Supplemental Information

Supplemental information for this article can be found online at http://dx.doi.org/10.7717/peerj-cs.450#supplemental-information.

REFERENCES

Cai X, Nie F, Cai W, Huang H. 2013. Heterogeneous image features integration via multi-modal semi-supervised learning model. In: *IEEE international conference on computer vision, Sydney, Australia, December 1-8*, Piscataway: IEEE, 1737–1744.

Chen X, Hong W, Nie F, He D, Yang M, Huang JZ. 2018. Spectral clustering of large-scale data by directly solving normalized cut. In: *Proceedings of ACM SIGKDD international conference on knowledge discovery & data mining, London, UK, August 19–23*. 1206–1215.

Chen X, Nie F, Huang JZ, Yang M. 2017. Scalable normalized cut with improved spectral rotation. In: *Proceedings of the twenty-sixth international joint conference on artificial intelligence, Melbourne, Australia, August 19–25*. 1518–1524.

Ding C. HQ, He X, Zha H, Gu M, Simon HD. 2001. A min-max cut algorithm for graph partitioning and data clustering. In: *Proceedings of IEEE international conference on data mining, San Jose, California, USA*, Piscataway: IEEE, 107–114.

Hagen LW, Kahng AB. 1992. New spectral methods for ratio cut partitioning and clustering. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* **11(9)**:1074–1085 DOI 10.1109/43.159993.

Huang J, Nie F, Huang H. 2013. Spectral rotation versus K-Means in spectral clustering. In: *Proceedings of the twenty-seventh AAAI conference on artificial intelligence, July 14–18, Bellevue, Washington, USA*. 431–437.

Huang J, Nie F, Huang H. 2015. A new simplex sparse learning model to measure data similarity for clustering. In: *Twenty-fourth international joint conference on artificial intelligence, July 25–31, Buenos Aires, Argentina*. 3569–3575.

Ng AY, Jordan MI, Weiss Y. 2001. On spectral clustering: analysis and an algorithm. In: *Advances in neural information processing systems, December 3–8, Vancouver, British Columbia, Canada*. 849–856.

Nie F, Ding CHQ, Luo D, Huang H. 2010. Improved MinMax cut graph clustering with nonnegative relaxation. In: *European Conference on machine learning and knowledge discovery in databases, Barcelona, Spain*. 451–466.

Nie F, Huang H. 2016. Subspace clustering via new low-rank model with discrete group structure constraint. In: *Proceedings of the twenty-fifth international joint conference on artificial intelligence, New York, NY, USA, 9–15 July*. 1874–1880.

Nie F, Tian L, Li X. 2018. Multiview clustering via adaptively weighted procrustes. In: *Proceedings of ACM SIGKDD international conference on knowledge discovery & data mining, London, UK, August 19–23*, New York: ACM, 2022–2030.

Nie F, Wang X, Huang H. 2014. Clustering and projected clustering with adaptive neighbors. In: *The ACM SIGKDD international conference on knowledge discovery and data mining, New York, NY, August 24-27*, New York: ACM, 977–986.

Nie F, Wang X, Jordan MI, Huang H. 2016. The constrained Laplacian rank algorithm for graph-based clustering. In: *Proceedings of the thirtieth AAAI conference on artificial intelligence, February 12–17, Phoenix, Arizona, USA*. 1969–1976.

Nie F, Zhang R, Li X. 2017. A generalized power iteration method for solving quadratic problem on the Stiefel manifold. *Science China–Information Sciences* **60(11)**:1–10.

Peng Y, Tang R, Kong W, Qin F, Nie F. 2018. Parallel vector field regularized non-negative matrix factorization for image representation. In: *IEEE international conference on acoustics, speech and signal processing, April 15–20, Calgary, AB, Canada*, Piscataway: IEEE, 2216–2220.

Peng Y, Zhang L, Kong W, Qin F, Zhang J. 2020. Joint low-rank representation and spectral regression for robust subspace learning. *Knowledge-Based Systems* **195**:1–14.

Ren M, Zhang Q, Zhang J. 2019. An introductory survey of probability density function control. *Systems Science & Control Engineering* **7(1)**:158–170 DOI 10.1080/21642583.2019.1588804.

Shi J, Malik J. 1997. Normalized cuts and image segmentation. In: *IEEE conference on computer vision and pattern recognition, June 17–19, San Juan, Puerto Rico*, Piscataway: IEEE, 731–737.

Tang X, Zhang Q, Dai X, Zou Y. 2020. Neural membrane mutual coupling characterisation using entropy-based iterative learning identification. *IEEE Access* **8**:205231–205243 DOI 10.1109/ACCESS.2020.3037816.

Ubukata S. 2019. A unified approach for cluster-wise and general noise rejection approaches for k-means clustering. *PeerJ Computer Science* **5**:e238 DOI 10.7717/peerj-cs.238.

Xie D, Nie F, Gao Q, Xiao S. 2020. A fast algorithm for Large-scale Subspace Clustering by LPP. *IET Image Processing* **14(8)**:1475–1480 DOI 10.1049/iet-ipr.2018.6596.

Yu SX, Shi J. 2003. Multiclass spectral clustering. In: *IEEE international conference on computer vision, 14–17 October, Nice, France*, Piscataway: IEEE, 313–319.