International Journal of Computer Vision (2020) 128:1982–1995
https://doi.org/10.1007/s11263-020-01320-3

Enhanced Balanced Min Cut

Xiaojun Chen[1] · Weijun Hong[1] · Feiping Nie[2] · Joshua Zhexue Huang[1] · Li Shen[3]

Received: 25 September 2019 / Accepted: 6 March 2020 / Published online: 26 March 2020

Abstract
Spectral clustering is a hot topic and many spectral clustering algorithms have been proposed. These algorithms usually solve the discrete cluster indicator matrix by relaxing the original problems, obtaining the continuous solution and finally obtaining a discrete solution that is close to the continuous solution. However, such methods often result in a non-optimal solution to the original problem since the different steps solve different problems. In this paper, we propose a novel spectral clustering method, named as Enhanced Balanced Min Cut (EBMC). In the new method, a new normalized cut model is proposed, in which a set of balance parameters are learned to capture the differences among different clusters. An iterative method with proved convergence is used to effectively solve the new model without eigendecomposition. Theoretical analysis reveals the connection between EBMC and the classical normalized cut. Extensive experimental results show the effectiveness and efficiency of our approach in comparison with the state-of-the-art methods.

Keywords Clustering · Spectral clustering · Normalized cut

1 Introduction

Clustering, which partitions a set of objects in a dataset into different groups such that the objects in the same group have high similarities to each other and the objects in different groups, is one of the most important tasks in data mining. During the past decades, many clustering algorithms have been proposed, such as k-means, hierarchical clustering (Johnson 1967), density-based clustering (Ester et al. 1998), Gaussian Mixture Models (GMMs) spectral clustering (Von Luxburg 2007), subspace clustering (Elhamifar and Vidal 2013; Chen et al. 2012), multi-view clustering (Cai et al. 2011; Chen et al. 2013), co-clustering (Chen et al. 2018), etc. Among them, k-means is one of the most popularly used clustering method due to its efficiency and simpleness. However, k-means often shows poor performance since it is difficult to distinguish non-sphere clusters. Recently, spectral clustering has received a lot of attentions because it is easy to implement and often shows good clustering performance due to its ability to uncover clusters in arbitrary shapes. They have been successfully applied to many applications, such as image clustering (Chen et al. 2017), image segmentation (Shi and Malik 2000; Yu and Shi 2003) and gene expression data clustering (de Souto et al. 2008).

Given a dataset, spectral clustering usually constructs a weighted undirected graph from the pair-wise similarity matrix known as the affinity matrix. The typical spectral clustering is formalized as a min-cut problem that aims to partition the vertices in the graph into several disjoint sets such that the total weight of the set of edges with endpoints in different sets is minimized (Von Luxburg 2007), e.g., Ratio Cut (Hagen and Kahng 1992) and Normalized Cut (Shi and Malik 2000). Since it is difficult to directly solve the discrete cluster indicator matrix, researchers usually relax the dis-

Communicated by Zhouchen Lin.

✉ Xiaojun Chen
xjchen@szu.edu.cn

Weijun Hong
280996118@qq.com

Feiping Nie
feipingnie@gmail.com

Joshua Zhexue Huang
zx.huang@szu.edu.cn

Li Shen
mathshenli@gmail.com

[1] College of Computer Science and Software, Shenzhen University, Shenzhen 518060, People's Republic of China

[2] Computer Science and Center for OPTIMAL, Northwestern Polytechnical University, Xi'an 710072, Shanxi, People's Republic of China

[3] Tencent AI Lab, Shenzhen 518060, People's Republic of China

crete cluster indicator matrix into continuous one, solve the relaxed problem with eigendecomposition and finally obtain the final discrete cluster indicator matrix with k-means or spectral rotation (Yu and Shi 2003; Huang et al. 2013; Lu et al. 2016; Chen et al. 2017). However, the convergence of such a two-stage process cannot be guaranteed since the different steps solve different problems, which often lead to a non-optimal solution to the original problem. Although some new spectral clustering methods have been proposed (Nie et al. 2014, 2016), they also rely on eigendecomposition. In this new method, a balance regularization is implicitly introduced to obtain a balanced partition. An iterative algorithm is proposed to directly solve the new model without relaxation of the cluster indicator matrix that is required in the classical Normalized Cut or Ratio Cut. The new method was proved to be able to simultaneously minimize the graph cut and balance the partition. The remaining problem of this method is that a scalar is learned to balance the partition across all clusters, but it cannot capture the differences among different clusters. Moreover, the iterative process of this method is of no convergence guarantee, and it is difficult to set the two parameters in the Augmented Lagrangian multiplier (ALM) that is used to solve the cluster indicator matrix in each iteration.

In this paper, we propose an extension to SBMC, named as Enhanced Balanced Min Cut (EBMC). The main contributions of this paper include:

1. We propose a new normalized cut model with a balance regularization to avoid a trivial solution, which uses a set of balance parameters to capture the differences among different clusters. These balance parameters are automatically learned during the clustering process.
2. We propose an iterative method with proved convergence to effectively solve the new model without the relaxation step as used in the conventional spectral clustering methods.
3. We give theoretical analysis to reveal the connection between EBMC and the classical normalized cut.
4. Our experimental results have shown the superior performance and efficiency of the new method in comparison to the state-of-the-art methods.

The rest of this paper is organized as follows. Section 2 presents the notations and definitions, and gives a brief review of related work. EBMC is proposed in Sect. 3. The experimental results on both synthetic and real-world datasets are reported in Sect. 4. Conclusions and future work are given in Sect. 5.

2 Background and Related Work

In this section, we introduce the notations and definitions and give a brief review of spectral clustering.

2.1 Notations and Definitions

We summarize the notations and the definition of norms used in this paper. Matrices are written as boldface uppercase letters. Vectors are written as boldface lowercase letters. For matrix $\mathbf{M} = (m_{ij})$, its i-th row is denoted as $\mathbf{m}^i$, and its j-th column is denoted by $\mathbf{m}_j$. The Frobenius norm of the matrix $\mathbf{M} \in \mathbb{R}^{n\times m}$ is defined as $\|\mathbf{M}\|_F = \sqrt{\sum_{i=1}^n \sum_{j=1}^m m_{ij}^2}$. We denote the set of all cluster indicator matrices as Ψ, which consists of only one element of value 1 in each row and the values of other elements are zeros.

2.2 Balance Regularization

Let $\mathbf{Y} \in \Psi^{n\times c}$ be a cluster indicator matrix, in which $y_{il} = 1$ indicates that $\mathbf{x}_i$ is assigned to the l-th cluster. The balance regularization $\|\mathbf{Y}\|_b$ is defined as

$$\|\mathbf{Y}\|_b = \sum_{j=1}^{c} \left(\sum_{i=1}^{n} y_{ij} \right)^2 = Tr(\mathbf{Y}^T \mathbf{1}_n \mathbf{1}_n^T \mathbf{Y}) \tag{1}$$

The following theorem ensures that $\|\mathbf{Y}\|_b$ can be used to obtain the most balanced partition (see Appendix A for proof)

Theorem 1 *Solving* $\min_{\mathbf{Y}} \|\mathbf{Y}\|_b$ *results in the most balanced partition.*

Therefore, we can use $\|\mathbf{Y}\|_b$ as a regularization term in order to avoid trivial solution with one isolated object as cluster.

2.3 Spectral Clustering

Given an undirected weighted graph $G = (V, E)$ in which the n vertices in the vertex set node V are n samples, and the edges E is associated with an affinity matrix $\mathbf{A}$ in which a_{ij} is the similarity between the i-th object and the j-th object. Clustering the n objects represented by V is equivalent to partitioning V into c subsets. The commonly-used objective for such a partitioning is to have high sum of within-cluster similarities and low sum of inter-cluster similarities. Moreover, the clusters are expected to be balanced in the sense that the "size" of the clusters should not differ too much in order to avoid trivial solution.

Two most popular spectral clustering models are ratio cut and normalized cut. Suppose the vertices V in G is partitioned into c subsets $\{V_1, \ldots, V_c\}$, the classical ratio cut tries to balance the cardinality of the clusters as follow (Hagen and Kahng 1992)

$$\sum_{l=1}^{c} \frac{cut(V_l, V \backslash V_l)}{|V_l|} = \sum_{l=1}^{c} \frac{\sum_{v_i \in V_l, v_j \in V \backslash V_l} a_{ij}}{|V_l|} \tag{2}$$

and the classical normalized cut tries to balance the volume of the clusters as follow (Shi and Malik 2000)

$$\sum_{l=1}^{c} \frac{cut(V_l, V\backslash V_l)}{vol(V_l)} = \sum_{l=1}^{c} \frac{\sum_{v_i\in V_l, v_j\in V\backslash V_l} a_{ij}}{\sum_{v_i, v_j\in V_l} a_{ij}} \tag{3}$$

Let $\mathbf{Y} \in \Psi^{n\times c}$ be the cluster indicator matrix, in which $y_{il} = 1$ indicates that $\mathbf{x}_i$ is assigned to the l-th cluster. Then the objective function of the classical ratio cut clustering can be written as

$$\min_{\mathbf{Y}\in\Psi^{n\times c}} \sum_{l=1}^{c} \frac{\mathbf{y}_l^T \mathbf{L}_A \mathbf{y}_l}{\mathbf{y}_l^T \mathbf{y}_l} \tag{4}$$

and the objective function of the normalized cut can be represented by

$$\min_{\mathbf{Y}\in\Psi^{n\times c}} \sum_{l=1}^{c} \frac{\mathbf{y}_l^T \mathbf{L}_A \mathbf{y}_l}{\mathbf{y}_l^T \mathbf{D}_A \mathbf{y}_l} \tag{5}$$

where $\mathbf{L}_A = \mathbf{D}_A - \mathbf{A}$ is the Laplacian matrix and $\mathbf{D}_A$ is the corresponding degree matrix that is a diagonal matrix with the i-th diagonal element as $d_{ii} = \sum_{j=1}^{n} a_{ij}$.

It is difficult to directly solve problems (4) and (5) since both problems are NP-complete. A commonly-used approach is to relax $\mathbf{Y}$ in both problems into continuous ones and rewrite the objective function of ratio cut as

$$\min_{\mathbf{Y}^T\mathbf{Y}=\mathbf{I}_c} Tr(\mathbf{Y}^T \mathbf{L}_A \mathbf{Y}) \tag{6}$$

and the objective function of normalized cut as

$$\min_{\mathbf{Y}^T\mathbf{D}\mathbf{Y}=\mathbf{I}_c} Tr(\mathbf{Y}^T \mathbf{L}_A \mathbf{Y}) \tag{7}$$

Problems (6) and (7) can be solved by a two-stage process: performing eigendecomposition on $\mathbf{L}_A$ first, and then obtaining the final clustering assignments from eigenvectors by k-means, spectral rotation (Huang et al. 2013) and semi-definite programming (Bie and Cristianini 2006). Chen et al. (2018) proposed a Direct Normalized Cut to directly solve the k-way normalized cut model without relaxation.

Moreover, besides the classical ratio cut and normalized cut, researchers also proposed ratio Cheeger cut and normalized Cheeger cut (Bühler and Hein 2009) for bi-clusters partition which are similar to the classical ratio cut and normalized cut. Since there exists no generally accepted multipartition version of the Cheeger cuts, they usually perform multiple bi-partitions in order to obtain a multiple-cluster partition, and the eigendecomposition is still required.

Recently, Chen et al. (2017) proposed a novel normalized cut model, named as Self-Balanced Min Cut. In their method, they propose to learn the cluster indicator matrix $\mathbf{Y}$ in order to approximate $\mathbf{A}$ with $\mathbf{Y}\mathbf{Y}^T$ by solving the following problem

$$\min_{\mathbf{Y}\in\Psi^{n\times c},} \|\mathbf{A} - s\mathbf{Y}\mathbf{Y}^T\|_F^2 \tag{8}$$

where $s > 0$ is a balance parameter.

To effectively solve the above problem, they first rewrite problem (8) as a new problem

$$\max_{\mathbf{Y}\in\Psi^{n\times c},\ s>0} 2sTr(\mathbf{Y}^T\mathbf{A}\mathbf{Y}) - s^2\,\|\mathbf{Y}\|_b \tag{9}$$

in which the balance regularization in Eq. (1) is implicitly used as a regularization to avoid trivial solution and the balance parameter s is automatically learned.

Different from the previous relaxation-based spectral clustering, they proposed an alternative optimization approach without relaxation to solve problem (9). In their method, if $s > 0$ is fixed, the optimal solution of $\mathbf{Y}$ is obtained by solving the following problem with ALM

$$\min_{\mathbf{Y}\in\Psi^{n\times c},\ \mathbf{G}=\mathbf{Y}} Tr(\mathbf{Y}^T\Theta\mathbf{G}) \tag{10}$$

where $\Theta = \frac{s}{2}\mathbf{1}_n\mathbf{1}_n^T - \mathbf{A}$.

If $\mathbf{Y}$ is fixed, the optimal solution of s is

$$s = \frac{Tr(\mathbf{Y}^T\mathbf{A}\mathbf{Y})}{\|\mathbf{Y}\|_b}. \tag{11}$$

3 Enhanced Balanced Min Cut

In this section, we propose the enhanced balanced min cut model and an efficient iterative method to solve the model.

3.1 Proposed Method

Given an undirected weighted graph $G = (V, E)$ in which the vertices V represent n samples $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ and the edges E is associated with the affinity matrix $\mathbf{A}$. $\mathbf{A}$ is normalized such that it degree matrix $\mathbf{D}_A = \mathbf{I}$. Suppose we want to cluster $\mathbf{X}$ into c clusters. Let $\mathbf{Y} \in \Psi^{n\times c}$ be the cluster indicator matrix, in which $y_{il} = 1$ indicates that $\mathbf{x}_i$ is assigned to the l-th cluster. Denoting $\mathbf{B} = \mathbf{Y}\mathbf{Y}^T$, we know that $b_{ij} = 1$ if $\mathbf{y}^i = \mathbf{y}^j$, and $b_{ij} = 0$ otherwise. To obtain a good partition $\mathbf{Y}$, we hope that $b_{ij} = 0$ if a_{ij} is small, and $b_{ij} = 1$ if a_{ij} is big. Intuitively, we can obtain the cluster assignments by minimizing the difference between $\mathbf{A}$ and $\mathbf{Y}\mathbf{Y}^T$. In SBMC (Chen et al. 2017), a scalar s is learned to balance the partition across all clusters. However, the scalar s cannot capture the differences among different clusters. To solve this problem, we propose to learn the cluster indicator matrix $\mathbf{Y}$ through optimizing the following objective function

$$\min_{\mathbf{Y}\in\Psi^{n\times c},\ \mathbf{S}} \|\mathbf{A}-\mathbf{Y}\mathbf{S}\mathbf{Y}^T\|_F^2 \tag{12}$$

where $\mathbf{S}\in\mathbb{R}^{c\times c}$ is a diagonal matrix in which s_{ll} is used to balance the l-th cluster.

Problem (12) cannot be solved directly. Fortunately, we can rewrite problem (12) as a new problem which is much easy to solve according to the following theorem (see Appendix B for proof).

Theorem 2 *Solving problem* (12) *is equivalent to solving the following problem*

$$\max_{\mathbf{Y}\in\Psi^{n\times c},\ \mathbf{S}} \sum_{l=1}^{c} \mathbf{y}_l^T(2s_{ll}\mathbf{A}-s_{ll}^2\mathbf{1}\mathbf{1}^T)\mathbf{y}_l \tag{13}$$

Problem (13) can be solved with an alternative optimization approach by fixing $\mathbf{Y}$ and solving $\mathbf{S}$, and then fixing $\mathbf{S}$ and solving $\mathbf{Y}$. In the next two subsections, we show how to update $\mathbf{Y}$ and $\mathbf{S}$ in the alternative process.

3.1.1 Update Y with S Fixed

When $\mathbf{S}$ is fixed, problem (13) can be rewritten as

$$\max_{\mathbf{Y}\in\Psi^{n\times c}} \sum_{l=1}^{c} \mathbf{y}_l^T\left(2s_{ll}\mathbf{A}-s_{ll}^2\mathbf{1}\mathbf{1}^T\right)\mathbf{y}_l \tag{14}$$

In this paper, we propose an iterative method to solve problem (14). Suppose the optimal solution of $\mathbf{Y}$ in the r-th iteration is $\mathbf{Y}^r$, $\mathbf{Y}^{r+1}$ is solved from the following problem

$$\max_{\mathbf{Y}\in\Psi^{n\times c}} \sum_{l=1}^{c} \mathbf{y}_l^T\left(2s_{ll}\mathbf{A}-s_{ll}^2\mathbf{1}\mathbf{1}^T\right)\mathbf{y}_l^r-\frac{\eta}{2}\left\|\mathbf{Y}-\mathbf{Y}^r\right\|_F^2 \tag{15}$$

where η is a constant. If $\eta>0$, the second term will make $\mathbf{Y}$ close to $\mathbf{Y}^r$ in order to ensure the convergence of the clustering process. Moreover, the second term can also be used to avoid early stopping of the optimization algorithm which will be discussed in Sect. 3.3.

It can be verified that $Tr(\mathbf{Y}^T\mathbf{Y})=Tr((\mathbf{Y}^r)^T\mathbf{Y}^r)=n$, then we can rewrite problem (15) as

$$\max_{\mathbf{Y}\in\Psi^{n\times c}} \sum_{l=1}^{c} \mathbf{y}_l^T\mathbf{M}^{(l)}\mathbf{y}_l^r \tag{16}$$

where $\mathbf{M}^{(l)}$ is defined as

$$\mathbf{M}^{(l)}=2s_{ll}\mathbf{A}-s_{ll}^2\mathbf{1}\mathbf{1}^T+\eta\mathbf{I}_n \tag{17}$$

It can be verified that problem (16) has the following optimal solution

$$y_{ij}=\begin{cases}1 & \text{if } (\mathbf{M}^{(j)})^i\mathbf{y}_{j1}^r\geq(\mathbf{M}^{(j')})^i\mathbf{y}_{j'1}^r\\ 0 & \text{otherwise}\end{cases} \tag{18}$$

The iterative process to solve problem (14) is summarized in Algorithm 1. We will discuss how to set η in Sect. 3.3.

Algorithm 1 Algorithm to solve problem (14)

```
Input: An affinity matrix $\mathbf{A}$, $\mathbf{S}$, $\mathbf{Y}^0$, $\eta$.
Set $r = 0$.
repeat
    Update $\mathbf{Y}^{r+1}$ according to Eq. (18).
    Set $r = r + 1$.
until Problem (14) converges
Output: The cluster indicator matrix $\mathbf{Y}$.
```

3.1.2 Update S with Y Fixed

If $\mathbf{Y}$ is fixed, it can be verified that problem (13) is independent between different s_{ll}, so we can solve the following problem individually for each s_{ll}

$$\max_{s_{ll}} 2s_{ll}\mathbf{y}_l^T\mathbf{A}\mathbf{y}_l-s_{ll}^2\mathbf{y}_l^T\mathbf{1}\mathbf{1}^T\mathbf{y}_l \tag{19}$$

In the end, we can obtain the optimal solution of s_{ll} to the above problem as

$$s_{ll}=\frac{\mathbf{y}_l^T\mathbf{A}\mathbf{y}_l}{\mathbf{y}_l^T\mathbf{1}\mathbf{1}^T\mathbf{y}_l} \tag{20}$$

From Eq. (20), we can see that s_{ll} is inversely proportional to $\mathbf{y}_l^T\mathbf{1}\mathbf{1}^T\mathbf{y}_l$, the squared number of samples in the l-th cluster, and proportional to the within-cluster similarities $\mathbf{y}_l^T\mathbf{A}\mathbf{y}_l$. Therefore, the more compact the l-th cluster, the bigger the s_{ll}.

3.1.3 Optimization Algorithm

The detailed algorithm to solve problem (13), named as Enhanced Balanced Min Cut (EBMC), is summarized in Algorithm 2. The balance matrix $\mathbf{S}$ and cluster indicator matrix $\mathbf{Y}$ are iteratively updated until convergence. If we construct a k-nn affinity matrix $\mathbf{A}$, the new algorithm needs $O(r_1(nkc^2+r_2nkc))$ time to iteratively solve $\mathbf{S}$ and $\mathbf{Y}$, where r_1 is the number of iterations to update $\mathbf{S}$ and r_2 is the average number of iterations to update $\mathbf{Y}$. Here, the discrete solution $\mathbf{Y}$ converges very fast due to its limited solution space so r_2 is usually very small. Therefore, EBMC has a time complexity of $O(nkc^2)$.

Algorithm 2 EBMC: Algorithm to solve problem (13)

```
Input: An affinity matrix $\mathbf{A}$, the number of clusters $c$, $\eta$.
Initialize $\mathbf{Y}^0$.
Set $t = 0$.
repeat
    for $l = 1$ to $c$ do
        Update $s_{ll}$ according to Eq. (20).
    end for
    Call Algorithm 1 with $\mathbf{Y}^t$ and $\eta$ to obtain the optimal solution of $\mathbf{Y}^{t+1}$.
    Set $t = t + 1$.
until problem (13) converges
Output: The cluster indicator matrix $\mathbf{Y}$.
```

The convergence of Algorithm 2 can be ensured by the following theorem (see Appendix B for proof)

Theorem 3 *If $\{\mathbf{M}^{(1)}, \ldots, \mathbf{M}^{(c)}\}$ are all positive semi-definite matrices, Algorithm 2 will converge in a finite number of iterations.*

According to Eq. (17), we know that η plays an important role in the convergence of Algorithm 2. We will discuss how to set proper η in Sect. 3.3.

3.2 Connection to the Classical Normalized Cut

In the following, we show the connection between EBMC and the classical ratio cut and normalized cut. Substituting s_{ll} in Eq. (20) into problem (13) gives

$$\max_{\mathbf{Y}\in\Psi^{n\times c}} \sum_{l=1}^{c} \frac{\left(\mathbf{y}_l^T\mathbf{A}\mathbf{y}_l\right)^2}{\mathbf{y}_l^T\mathbf{1}\mathbf{1}^T\mathbf{y}_l} \Longleftrightarrow \max_{\mathbf{Y}\in\Psi^{n\times c}} \sum_{l=1}^{c} \left(\frac{\mathbf{y}_l^T\mathbf{A}\mathbf{y}_l}{\|\mathbf{y}_l\|_2^2}\right)^2 \tag{21}$$

Note that the classical normalized cut problem is

$$\min_{\mathbf{Y}\in\Psi^{n\times c}} \sum_{l=1}^{c} \frac{y_l^T\mathbf{L}_A y_l}{\mathbf{y}_l^T\mathbf{D}_A\mathbf{y}_l} \tag{22}$$

where $\mathbf{y}_l^T\mathbf{D}_A\mathbf{y}_l$ is used to improve its robustness to isolated nodes (Shi and Malik 2000).

Problem (22) can be further rewritten as

$$\min_{\mathbf{Y}\in\Psi^{n\times c}} \sum_{l=1}^{c} 1 - \frac{\mathbf{y}_l^T\mathbf{A}\mathbf{y}_l}{\mathbf{y}_l^T\mathbf{D}_A\mathbf{y}_l} \Longleftrightarrow \max_{\mathbf{Y}\in\Psi^{n\times c}} \sum_{l=1}^{c} \frac{\mathbf{y}_l^T\mathbf{A}\mathbf{y}_l}{\mathbf{y}_l^T\mathbf{D}_A\mathbf{y}_l} \tag{23}$$

Here we can see that both problem (21) and problem (23) aim to maximize the normalized within-cluster similarities, but with different normalization terms. Compared with the classical normalized cut, the main advantage of EBMC is that it is easy to directly solve the discrete $\mathbf{Y}$ without the relation of $\mathbf{Y}$.

3.3 Adjusting η

It can be verified that the eigenvalues of $\mathbf{M}^{(l)}$ are the eigenvalues of $2s_{ll}\mathbf{A}-(s_{ll})^2\mathbf{1}\mathbf{1}^T$ plus η. As η increases, all eigenvalues of $\mathbf{M}^{(l)}$ will be larger than zero after a number of iterations and $\mathbf{M}^{(l)}$ will become positive semi-definite. Therefore, we have to set a big η in order to ensure the convergence of Algorithm 2.

However, if η is too large, we have

$$\max_{\mathbf{Y}\in\Psi^{n\times c}} \sum_{l=1}^{c} \mathbf{y}_l^T(2s_{ll}\mathbf{A} - s_{ll}^2\mathbf{1}\mathbf{1}^T)\mathbf{y}_l^r - \frac{\eta}{2}\left\|\mathbf{Y}-\mathbf{Y}^r\right\|_F^2 \tag{24}$$
$$\longrightarrow \min_{\mathbf{Y}\in\Psi^{n\times c}} \left\|\mathbf{Y}-\mathbf{Y}^r\right\|_F^2$$

and the optimal solution is $\mathbf{Y} = \mathbf{Y}^r$, indicating that $\mathbf{Y}$ cannot be updated by Algorithm 2. Therefore, we need to set a proper η in order to avoid early stopping of Algorithm 2.

In this paper, we propose a simple yet efficient method to adaptively adjust η during iterations. In the beginning of the algorithm, we set a small η. Under the circumstance, $\{\mathbf{M}^1, \ldots, \mathbf{M}^c\}$ may not be all positive semi-definite and Algorithm 2 may be instable. This will avoid early stopping of Algorithm 2 and it may jump over some regions leading to undesired local optima. After that, we increase η rapidly to a large enough value, say η^u, which is as small as possible but makes $\{\mathbf{M}^1, \ldots, \mathbf{M}^c\}$ all positive semi-definite. Then, we will keep $\eta = \eta^u$ in the following runs. According to Theorem 3, we know that Algorithm 2 will converges finally. In such case, after some iterations when η becomes large enough such that $\{\mathbf{M}^1, \ldots, \mathbf{M}^c\}$ are positive semi-definite, the algorithm will monotonically increase the problem (13) until it converges. Specifically, we define η as

$$\eta = \eta^u \times f(t) \tag{25}$$

where η^u is the upper bound of η and it is defined as (see Appendix D)

$$\eta^u = \max_l(2s_{ll} + ns_{ll}^2) \tag{26}$$

and $f(t)$ is a function to adjust η according to the number of iterations t. A simple choice of $f(t)$ is

$$f(t) = \begin{cases} \left(\frac{t}{r}\right)^\rho & \text{if } t \le r \\ 1 & \text{if } t > r \end{cases} \tag{27}$$

where r is the number of iterations that η increases to η^u and ρ controls the increase speed of η.

4 Experimental Results and Analysis

In this section, we present the experimental results conducted on both synthetic and real-life datasets to demonstrate the efficiency and effectiveness of the proposed method. The experiments were conducted on a computer with 3.4 GHZ($\times 4$) CPU and the programs were implemented in MATLAB. For the eigendecomposition based spectral clustering methods, we only computed the first c eigenvalues for efficiency.

4.1 Experiments on Synthetic Datasets

In the synthetic data experiment, we first generated a dataset $\mathcal{D}_0$, which contains four imbalanced clusters which are also diagonally arranged, with the sizes of 10×10, 30×30, 20×20 and 40×40. The data within each block are the affinities of two corresponding points in one cluster. The affinity data within each block is randomly generated with values in $[0, 1]$, while the noise data is randomly generated with values in $[0, \psi]$ where the noise level ψ is a given parameter. Moreover, to make this clustering task more challenging, we randomly pick up 25 noise data points and set their values to be 1. By setting a set of 10 parameters $\psi = \{0.5, 0.55, \ldots, 0.95\}$, we generated 10 datasets from $\mathcal{D}_0$, noted as $\mathcal{D}_1, \ldots, \mathcal{D}_{10}$, respectively.

4.1.1 Optimization Study

In this experiment, we study the effectiveness of our proposed iterative optimization method for solving problem (13). We first introduce the existing optimization methods to solve problem (13) and implement two methods, i.e., **EBMC+KM** and **EBMC+ALM**.

In **EBMC+KM**, we first relaxes the discrete $\mathbf{Y}$ in problem (13) into continuous matrix, and obtain the following relaxed problem

$$\max_{\mathbf{Y}\in\mathbb{R}^{n\times c}, \mathbf{Y}^T\mathbf{Y}=\mathbf{I}_c, \mathbf{S}} \sum_{l=1}^{c} \mathbf{y}_l^T \left(2s_{ll}\mathbf{A} - s_{ll}^2\mathbf{1}\mathbf{1}^T\right)\mathbf{y}_l \tag{28}$$

Then we propose an iterative method to alternatively update $\mathbf{Y}$ and $\mathbf{S}$. When $\mathbf{Y}$ is fixed, $\mathbf{S}$ is solved according to Eq. (20). When $\mathbf{S}$ is fixed, we rewrite problem (28) as a continuous problem with orthogonality constraint as follow

$$\max_{\mathbf{Y}\in\mathbb{R}^{n\times c}, \mathbf{Y}^T\mathbf{Y}=\mathbf{I}_c} \sum_{l=1}^{c} \mathbf{y}_l^T \left(2s_{ll}\mathbf{A} - s_{ll}^2\mathbf{1}\mathbf{1}^T\right)\mathbf{y}_l \tag{29}$$

Denote $\mathbf{N}^{(l)} = 2s_{ll}\mathbf{A} - s_{ll}^2\mathbf{1}\mathbf{1}^T$. It can be verified that the optima solution to problem (29) consists of c eigenvectors of $\{\mathbf{N}^{(1)}, \ldots, \mathbf{N}^{(c)}\}$ corresponding to their c maximum eigenvalues. Finally, k-means is used to obtain the cluster indicator matrix from the obtained continuous $\mathbf{Y}$. Although spectral rotation is another commonly-used method to obtain the cluster indicator matrix from the obtained continuous $\mathbf{Y}$, it works under the rotation property, i.e., if $\mathbf{Y}$ is a solution to the original problem, $\mathbf{YR}$ is also a solution if $\mathbf{R}$ is an arbitrary orthonormal matrix. However, we can verify that problem (13) does not have the rotation property. Therefore, we did not use spectral rotation to solve problem (13).

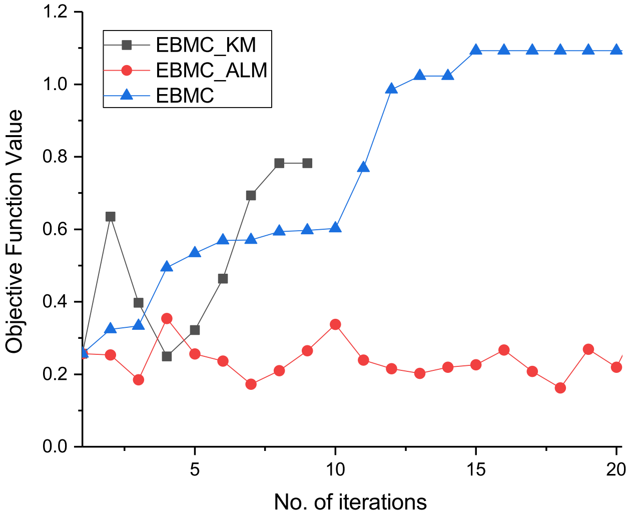


Fig. 1 Objective function values of problem (13) versus the number of iterations by **EBMC+KM**, **EBMC+ALM** and EBMC on $\mathcal{D}_8$

In **EBMC+ALM**, we solve problem (13) with an iterative method similar to Chen et al. (2017). If $\mathbf{Y}$ is fixed, $\mathbf{S}$ can be solved according to Eq. (20). If $\mathbf{S}$ is fixed, $\mathbf{Y}$ can be solved with the Augmented Lagrangian multiplier (ALM), which is same as SBMC (Chen et al. 2017). Similar to our proposed method in Algorithm 2, **EBMC+ALM** also directly solves $\mathbf{Y}$ without relaxation. However, as stated in Theorem 2, Algorithm 2 can monotonically increase the objective function value of problem (13) while **EBMC+ALM** cannot.

In the following, we select dataset D_8 to compare the effectiveness and efficiency of our proposed optimization method (in Algorithm 2) with **EBMC+KM** and **EBMC+ALM**. Figure 1 shows the objective function values by the three methods versus the number of iterations, where the same initial values for $\mathbf{Y}$ and $\mathbf{S}$ were used for all three methods. An examination of this figure reveals that our proposed method can monotonically increase the objective function value of problem (13) while the other two methods cannot. Moreover, it uncovers better solution with higher objective function values in 15 iterations. This result verifies the superior effectiveness and efficiency of our proposed optimization method.

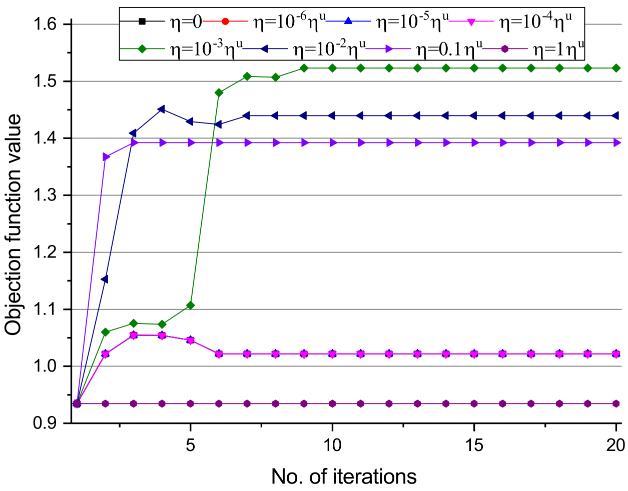


Fig. 2 Objective function values versus fixed η in EBMC on $\mathcal{D}_8$

4.1.2 Parameter Investigation

We first set η in EBMC as fixed values to investigate the effect of η on the performance of EBMC. In this experiment, we set $\eta = \{0, 10^{-6}\eta^u, 10^{-5}\eta^u, 10^{-4}\eta^u, 10^{-3}\eta^u, 10^{-2}\eta^u, 10^{-1}\eta^u, 10^{0}\eta^u\}$ where η^u is computed according to Eq. (26), and ran EBMC on $\mathcal{D}_8$. Figure 2 shows the changes of the objective function values with different fixed η. From this figure, we can see that the objective function value does not monotonically increase when $\eta \leq 10^{-4}\eta^u$ and $\eta = 10^{-2}\eta^u$. The algorithm produced the highest objective function value 1.5231 when $\eta = 10^{-3}\eta^u$. If we increase η to η^u, the algorithm won't update the initial value. This result indicates that η has a big effect on the performance of EBMC and it is difficult to set a proper η. Therefore, we propose to adaptively adjust η according to Eq. (25).

We then investigate the effect of two parameters ρ and r in Eq. (25) on the performance of EBMC. Figure 3a, c plot the objective function values versus parameter ρ, and parameter η versus parameter ρ, respectively, with fixed r set to 50. Figure 3a indicates that the highest objective function value 1.5265 was obtained when $\rho = 5$, which is higher than 1.5231 obtained by fixed η. Figure 3c indicates that the highest η^u [the biggest value in each curve is η^u according to Eq. (25)] was obtained when $\rho = 5$.

Figure 3b, d plot the objective function value versus r and parameter η versus r, respectively, in which ρ was set to 5. Figure 3b indicates that the highest objective function value 1.5265 was obtained when $r = \{10, 50\}$ and Fig. 3d indicates that the highest η^u was obtained when $r = \{10, 50\}$. We also observe that ρ affects the performance of EBMC more than r. Compared with the result in Fig. 2, we can see that EBMC with η adjusted according to Eq. (25) outperforms EBMC with fixed η. In real-world applications, we can perform hierarchy grid search to select proper ρ and r with the highest objective function value for better result.

4.1.3 Robustness

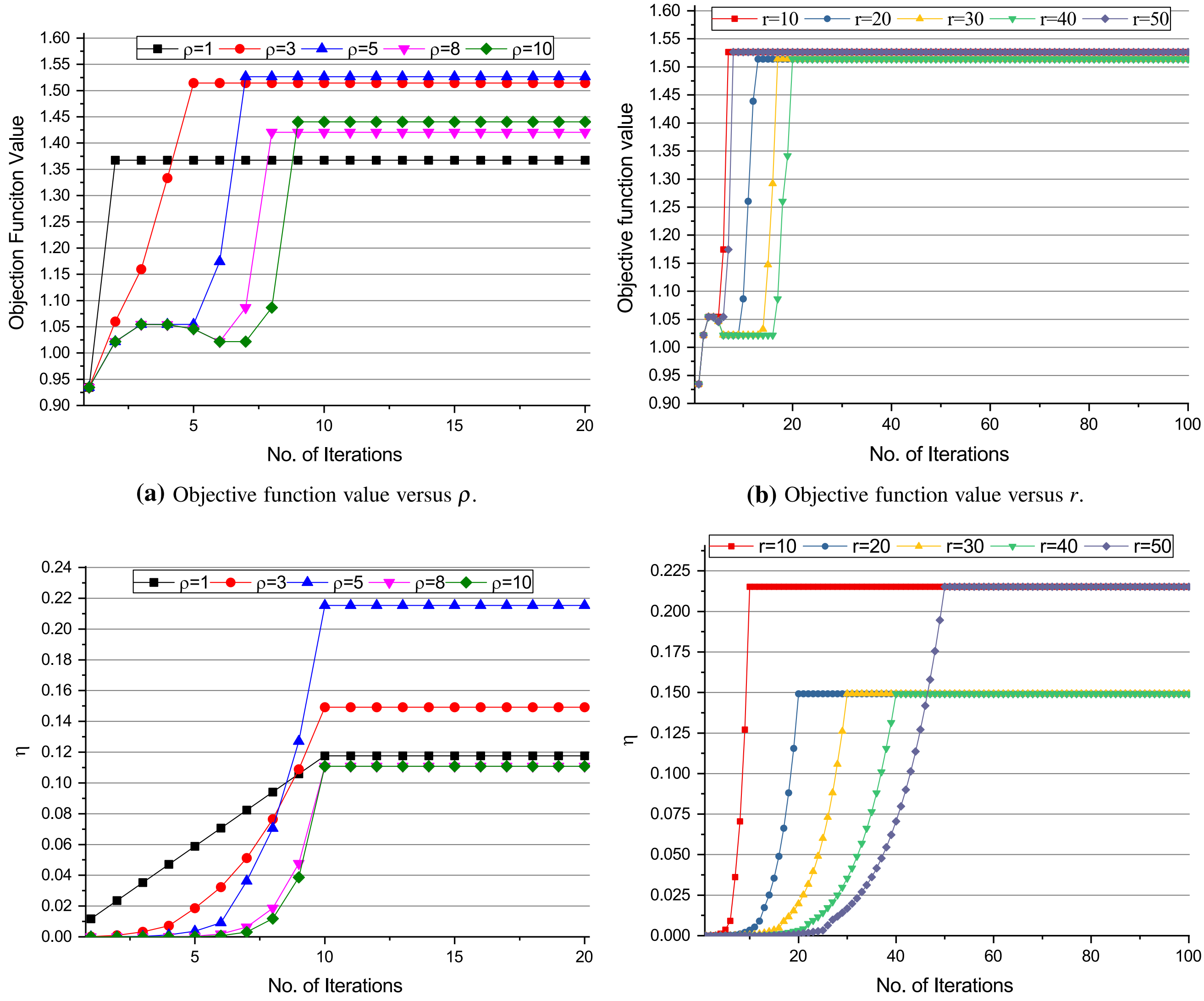
In this experiment, we compared EBMC with nine clustering methods, including Normalized Cut (NCut) (Ng et al. 2002), Ratio Cut (RCut) (Hagen and Kahng 1992), Multiclass Spectral Clustering (MSC) (Yu and Shi 2003), Spectral Embedded Clustering (SEC) (Nie et al. 2011), Constrained Laplacian Rank (CLR2-Constrained Laplacian Rank with ℓ_2 norm, CLR1-Constrained Laplacian Rank with ℓ_1 norm) (Nie et al. 2016), SBMC (Chen et al. 2017), DNC (Chen et al. 2018), **EBMC+KM** and **EBMC+ALM**. For each of ten datasets in $\{\mathcal{D}_1, \ldots, \mathcal{D}_{10}\}$, we set the neighborhood parameter $k = 10$ to construct a sparse k nearest neighbors affinity matrix with the similarity matrix construction method in(Nie et al. 2016), and used the affinity matrix to run the seven methods in order to perform fair comparison. We set $\rho = 5$ and $r = 50$ for EBMC. Since CLR1, CLR2 and SBMC are parameter free,[1] we ran each of them on each dataset 100 times and selected the best clustering result according to their objective function. For NCut and RCut, we selected the clustering results with the minimal objective function from 100 k-means clustering results on each dataset. We only computed the first c eigenvalues for efficiency. The clustering results in terms of accuracy (**ACC**), normalized mutual information (**NMI**) and rand index (**RI**) are shown in Table 1. From this table, we can see that EBMC outperforms other methods on most datasets. Especially on $\mathcal{D}_8$, EBMC achieves a nearly 10% improvement in terms of ACC, compared to the second-best method MSC. EBMC also achieves a nearly 16% improvement in terms of NMI, compared to the second-best methods RCut and MSC. We also notice that EBMC outperforms **EBMC+KM** and **EBMC+ALM** on most datasets, which again indicates the superior effectiveness of our proposed optimization method.

Figure 4 shows the four diagonal elements in **S** learned from 10 datasets. From this figure, we can see that s_{ll} decreases with the increase of noise level ψ. According to the analysis in Sect. 3.1.2, we know that bigger s_{ll} indicates smaller and more compact cluster. This experimental result verifies the analysis in Sect. 3.1.2.

4.1.4 Scalability

In this experiment, we first selected $\mathcal{D}_8$ as the baseline datasets to generate 10 synthetic datasets $\{\mathcal{D}_{80}, \ldots, \mathcal{D}_{89}\}$ by setting the number of objects as $\{100, 200, \ldots, 100 \times 2^9\}$. Then we conducted experiment to compare the scalability of the proposed EBMC with nine clustering methods used in

[1] The parameter ρ in SBMC can be randomly initialized in $(1, 2)$.



(a) Objective function value versus ρ.

(b) Objective function value versus r.

(c) η versus ρ.

(d) η versus r.

Fig. 3 Objective function values and ηs versus ρ and r in EBMC on $\mathcal{D}_8$

Table 1 Average accuracies by 10 spectral clustering methods on 10 synthetic datasets $\{\mathcal{D}_1, \ldots, \mathcal{D}_{10}\}$

		$\mathcal{D}_1$	$\mathcal{D}_2$	$\mathcal{D}_3$	$\mathcal{D}_4$	$\mathcal{D}_5$	$\mathcal{D}_6$	$\mathcal{D}_7$	$\mathcal{D}_8$	$\mathcal{D}_9$	$\mathcal{D}_{10}$
NCut	ACC	**1.00**	**1.00**	**1.00**	**1.00**	0.98	0.81	0.77	0.54	0.44	0.30
	NMI	**1.00**	**1.00**	**1.00**	**1.00**	0.94	0.68	0.22	0.22	**0.13**	0.02
	RI	**1.00**	**1.00**	**1.00**	**1.00**	0.98	0.90	0.85	0.69	**0.65**	0.60
RCut	ACC	**1.00**	**1.00**	**1.00**	**1.00**	0.98	0.93	0.84	0.63	0.43	0.38
	NMI	**1.00**	**1.00**	**1.00**	**1.00**	0.94	0.83	0.32	0.32	0.12	0.06
	RI	**1.00**	**1.00**	**1.00**	**1.00**	0.98	0.96	0.90	0.74	0.64	0.61
MSC	ACC	**1.00**	0.99	**1.00**	0.99	0.97	0.69	0.7	0.64	0.38	0.33
	NMI	**1.00**	0.96	**1.00**	0.97	0.93	0.64	0.24	0.32	0.08	0.04
	RI	**1.00**	0.99	**1.00**	0.99	0.98	0.82	0.78	**0.75**	0.63	0.61
CLR1	ACC	**1.00**	**1.00**	**1.00**	0.77	0.99	0.92	0.66	0.48	0.41	0.42
	NMI	**1.00**	**1.00**	0.96	0.97	**1.00**	0.79	0.27	0.27	0.07	**0.07**
	RI	**1.00**	**1.00**	0.99	0.99	**1.00**	0.91	0.92	0.68	0.42	0.48

Table 1 continued

		$\mathcal{D}_1$	$\mathcal{D}_2$	$\mathcal{D}_3$	$\mathcal{D}_4$	$\mathcal{D}_5$	$\mathcal{D}_6$	$\mathcal{D}_7$	$\mathcal{D}_8$	$\mathcal{D}_9$	$\mathcal{D}_{10}$
CLR2	ACC	**1.00**	**1.00**	0.99	0.99	**1.00**	0.8	0.85	0.61	0.37	**0.44**
	NMI	**1.00**	**1.00**	**1.00**	0.70	0.97	0.83	0.12	0.12	0.04	0.06
	RI	**1.00**	**1.00**	**1.00**	0.86	0.99	0.96	0.76	0.42	0.37	0.36
SBMC	ACC	**1.00**	**1.00**	**1.00**	**1.00**	0.99	0.93	**0.88**	0.58	0.34	0.39
	NMI	**1.00**	**1.00**	**1.00**	**1.00**	0.97	0.84	0.31	0.29	0.06	**0.07**
	RI	**1.00**	**1.00**	**1.00**	**1.00**	**0.99**	0.96	**0.93**	0.72	0.60	**0.62**
DNC	ACC	**1.00**	**1.00**	**1.00**	0.99	0.98	0.79	0.75	0.68	0.41	0.36
	NMI	**1.00**	**1.00**	**1.00**	0.97	0.94	0.74	0.57	0.08	0.12	0.05
	RI	**1.00**	**1.00**	**1.00**	0.99	0.98	0.88	0.86	**0.75**	0.63	**0.62**
EBMC+KM	ACC	**1.00**	**1.00**	**1.00**	0.99	0.98	0.71	0.77	0.54	0.36	0.37
	NMI	**1.00**	**1.00**	**1.00**	0.96	0.85	0.46	0.59	0.36	0.14	0.03
	RI	**1.00**	**1.00**	**1.00**	0.99	0.95	0.69	0.81	0.74	0.50	0.40
EBMC+ALM	ACC	0.66	0.42	0.52	0.40	0.39	0.38	0.59	0.41	0.39	0.32
	NMI	0.28	0.21	0.19	0.19	0.10	0.16	0.28	0.08	0.05	0.02
	RI	0.66	0.54	0.57	0.55	0.45	0.45	0.62	0.45	0.43	0.44
EBMC	ACC	**1.00**	**1.00**	**1.00**	**1.00**	0.99	**0.96**	**0.88**	**0.7**	**0.47**	0.42
	NMI	**1.00**	**1.00**	**1.00**	**1.00**	0.97	**0.92**	**0.34**	**0.37**	**0.13**	**0.07**
	RI	**1.00**	**1.00**	**1.00**	**1.00**	**0.99**	**0.98**	**0.93**	**0.75**	0.62	0.61

The best result on each dataset is highlighted in bold

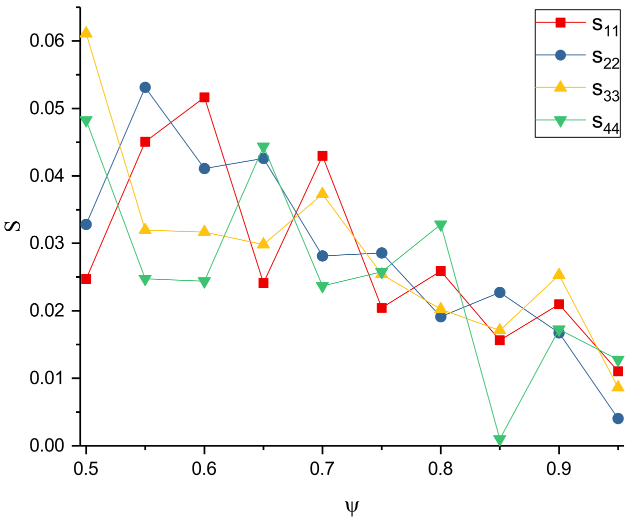


Fig. 4 The learned four learned diagonal elements in **S** by EBMC versus ψ

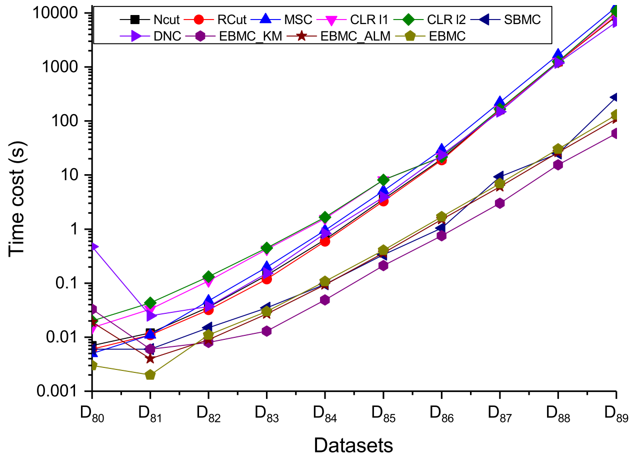


Fig. 5 Time costs by 10 spectral clustering methods on 10 synthetic datasets $\{\mathcal{D}_{80}, \ldots, \mathcal{D}_{89}\}$

the previous subsection. The time cost results are reported in Fig. 5. From this figure, we can observe that EBMC, EBMC+KM, EBMC+ALM and SBMC take comparable time and show similar tendency with the increase of the number of objects. The other six methods take much more time and their time costs increase faster than the four methods, i.e., EBMC, EBMC+KM, EBMC+ALM and SBMC. This indicates that directly solving the cluster indicator matrix takes less time than the eigendecomposition based methods. EBMC takes more time than SBMC since it has to learn more balance parameters. Although EBMC+KM takes less time than our proposed iterative method, the results in Table 1 show that it has low robustness. In summary, our proposed method is both robust and scalable.

4.2 Experiments on Real-World Datasets

In this subsection, we report the comparison results of EBMC against other spectral clustering methods on eleven real-world datasets.

Table 2 Characteristics of 11 real-world datasets

Dataset	#Samples	#Features	#Classes
Yale32	165	1024	15
ORL	400	1024	40
MSRA25	1799	256	12
USPS20	1854	256	10
uspst	2007	256	10
segment	2310	19	7
isolet	7797	617	26
letter-recognition	20,000	16	26
Cora	2708	–	7
Citeseer	3327	–	6
Pubmed	19,717	–	3

(a) Sample images in the MSRA25 dataset.

(b) Sample images in the ORL dataset.

(c) Sample images in the Yale32 dataset.

Fig. 6 Some sample images in the benchmark datasets

4.2.1 Benchmark Datasets

Eleven real-world benchmark datasets were used in these experiments. Table 2 summarizes the characteristics of these datasets and Fig. 6 shows some sample images.

1. **Yale32** dataset consists of 165 grayscale images in GIF format of 15 individuals represented by 1024 features, which was downloaded from Deng Cai's page.[2]
2. **ORL** face dataset consists of 10 different images, each with 40 distinct subjects. This dataset was downloaded from Deng Cai's page.[2]
3. **MSRA25** dataset consists of 1799 images in 10 classes represented by 256 features, which was downloaded from Feiping Nie's page.[3]
4. **USPS20** dataset consists of 1854 images in 10 classes represented by 256 features, which were 20% of the original USPS dataset (Hull 2002).
5. **uspst** dataset consists of 2007 images in 10 classes represented by 256 features which was downloaded from Feiping Nie's page.[3]
6. **segment** dataset consists of 2310 images randomly drawn from a database of 7 outdoor images represented by 19 features, which was downloaded from the UCI Machine Learning Repository.
7. **isolet** dataset consists of 7797 samples of 26 letters represented by 617 features spoken by 150 speakers. This dataset was downloaded from the UCI Machine Learning Repository.
8. **letter-recognition** dataset consists of 20,797 character images of 26 capital letters in the English alphabet represented by 16 features. This dataset was downloaded from the UCI Machine Learning Repository.
9. **Cora** is a citation network dataset (Sen et al. 2008) which consists of 2708 scientific publications classified into seven classes, and 5429 links between these publications.
10. **Citeseer** is a citation network dataset (Sen et al. 2008) which consists of 3327 scientific publications classified into six classes, and 4732 links between these publications.
11. **Pubmed** is a citation network dataset (Sen et al. 2008) which consists of 19,717 scientific publications from PubMed database pertaining to diabetes classified into three classes, and 44,338 links between these publications.

4.2.2 Results and Analysis

In this experiment, we compared EBMC with 7 clustering methods, including Normalized Cut (NCut) (Ng et al. 2002), Ratio Cut (RCut) (Hagen and Kahng 1992), Multiclass Spectral Clustering (MSC) (Yu and Shi 2003), Spectral Embedded Clustering (SEC) (Nie et al. 2011), Constrained Laplacian Rank (CLR1-Constrained Laplacian Rank with ℓ_1 norm, CLR2-Constrained Laplacian Rank with ℓ_2 norm) (Nie et al. 2016), SBMC (Chen et al. 2017) and DNC (Chen et al. 2018).

[2] http://www.cad.zju.edu.cn/home/dengcai/Data/FaceData.html.

[3] http://www.escience.cn/people/fpnie/index.html#.

Table 3 Average accuracies by 8 spectral clustering methods on 8 datasets

Method	Metric	Yale32	ORL	MSRA25	USPS20	Uspst	Segment	Isolet	Letter-recognition	Cora	Citeseer	Pubmed
NCut	ACC	0.436	0.535	0.574	0.659	0.648	0.529	0.523	**0.310**	0.297	0.203	0.386
	NMI	0.467	0.756	0.675	0.742	0.725	0.511	0.725	0.380	0.020	0.008	0.000
	RI	**0.908**	0.968	0.892	0.926	0.908	0.823	0.957	0.883	**0.737**	0.408	0.520
RCut	ACC	0.448	0.523	0.558	0.655	0.644	0.429	0.538	0.280	0.174	0.277	0.351
	NMI	0.504	0.754	0.687	0.740	0.726	0.530	0.737	0.350	0.005	0.056	**0.001**
	RI	0.903	0.970	0.862	0.925	0.921	0.815	0.959	0.849	0.185	0.709	**0.542**
MSC	ACC	0.412	0.545	0.574	0.656	0.643	0.511	0.533	0.300	0.246	0.285	0.394
	NMI	0.491	0.721	0.661	0.734	0.711	0.510	0.739	0.360	0.072	0.062	0.001
	RI	0.904	0.969	0.914	0.924	0.921	0.813	0.954	0.802	0.607	0.683	0.480
CLR1	ACC	0.303	0.338	0.558	0.439	0.477	0.420	0.227	0.170	0.302	0.208	0.399
	NMI	0.327	0.486	0.688	0.378	0.388	0.413	0.402	0.180	0.005	0.005	0.000
	RI	0.556	0.769	0.855	0.812	0.652	0.665	0.625	0.340	0.184	0.187	0.361
CLR2	ACC	0.412	0.475	0.558	0.680	0.701	0.496	0.491	0.160	0.258	0.210	0.399
	NMI	0.447	0.660	0.688	0.772	**0.787**	0.412	0.705	0.190	0.052	0.006	0.000
	RI	0.815	0.923	0.894	0.927	0.930	0.794	0.933	0.353	0.387	0.187	0.372
SBMC	ACC	0.448	0.423	0.519	0.684	0.691	0.561	0.412	0.110	0.303	0.186	0.346
	NMI	0.489	0.614	0.679	0.718	0.608	0.335	0.568	0.080	0.105	0.001	0.000
	RI	0.894	0.962	0.903	0.902	0.915	0.830	0.942	0.927	0.412	0.715	0.548
DNC	ACC	0.394	**0.588**	0.594	0.673	0.667	**0.587**	0.550	0.260	0.288	0.248	0.399
	NMI	0.471	**0.765**	0.653	0.666	0.637	**0.564**	0.747	0.380	0.078	0.044	0.000
	RI	0.895	0.971	0.908	0.920	0.912	0.829	0.960	0.922	0.525	0.342	0.357
EBMC	ACC	**0.497**	**0.588**	**0.611**	**0.727**	**0.768**	0.550	**0.565**	0.270	**0.328**	**0.473**	**0.399**
	NMI	**0.521**	**0.765**	**0.693**	**0.757**	0.777	0.546	**0.754**	**0.390**	**0.130**	**0.239**	0.000
	RI	0.907	**0.972**	**0.916**	**0.933**	**0.942**	**0.836**	**0.961**	**0.932**	0.613	**0.747**	0.357

The best result on each dataset is highlighted in bold

We used the same strategies in Sect. 4.1.3 to run these methods. The clustering results of 8 spectral clustering algorithms on 11 real-world datasets are shown in Table 3, which indicate that EBMC outperformed other methods on most datasets. Especially on the letter-recognition dataset, EBMC achieves a nearly 10% improvement in terms of ACC, compared to the second-best method MSC. EBMC also achieves a nearly 16% improvement in terms of NMI, compared to the second-best methods RCut and MSC. This result indicates the superiority of the new method.

5 Conclusions

This paper presents the novel spectral clustering method, Enhanced Balanced Min Cut (EBMC). We have introduced a set of balance parameters to capture the differences among different clusters and proposed a novel normalized cut. An iterative method with proved convergence is proposed to effectively solve the new method. Theoretical analysis reveals that EBMC can uncover compact clusters. Moreover, we have discussed the similarities and differences between the new method and conventional normalized cut, and pointed out that the new method is robust to noise.

One shortcoming is that EBMC requires an $n \times n$ affinity matrix as input so is not scalable to large-scale data. In the future work, we will improve it to handle large-scale data. Incorporating deep learning technique into EBMC is also a future direction.

Acknowledgements This research was supported by NSFC under Grant Nos. 61773268, 61502177 and Natural Science Foundation of SZU (Grant No. 000346) and the Shenzhen Research Foundation for Basic Research, China (Nos. JCYJ20180305124149387).

A Proof of Theorem 1

Proof Let $\mathbf{u} \in \mathbb{R}^{c\times 1}$ be a column vector where $u_j = \sum_{i=1}^{n} y_{ij}$ and $\sum_{j=1}^{c} u_j = n$. Let $\mathbf{v} \in \mathbb{R}^{\times 1}$ be a constant column vector where $v_j = \frac{1}{c}$. According to the Cauchy-Schwarz inequality, we have $| < \mathbf{u}, \mathbf{v} > |^2 \leq \|\mathbf{u}\|_2^2 \, \|\mathbf{v}\|_2^2$ which indicates that

$$\sum_{j=1}^{c} u_j^2 \geq \frac{n^2}{c} \tag{30}$$

and the inequality hold when $u_j = \frac{n}{c}$ for $\forall j \in [1, c]$. Therefore, $\|\mathbf{Y}\|_b$ arrives its minimum when $\sum_{i=1}^{n} f_{il} = \frac{n}{c}$ if $\frac{n}{c}$ is an integer, or $\sum_{i=1}^{n} f_{il} = \{\lfloor \frac{n}{c} \rfloor, \lceil \frac{n}{c} \rceil\}$ otherwise ($l \in [1, c]$). Therefore, solving $min_{\mathbf{Y}} \|\mathbf{Y}\|_b$ results in the most balanced partition. □

B Proof of Theorem 2

Proof Problem (12) can be rewritten as

$$\begin{aligned}\min_{\mathbf{Y}\in\Psi^{n\times c},\mathbf{S}} \|\mathbf{A} - \mathbf{YSY}^T\|_F^2 \Leftrightarrow \max_{\mathbf{Y}\in\Psi^{n\times c},\mathbf{S}} & 2Tr\left(\mathbf{SY}^T\mathbf{AY}\right) \\ & -Tr\left(\mathbf{YSY}^T\mathbf{YSY}^T\right)\end{aligned} \tag{31}$$

$Tr(\mathbf{YSY}^T\mathbf{YSY}^T)$ can be rewritten as

$$Tr\left(\mathbf{YSY}^T\mathbf{YSY}^T\right) = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{l=1}^{c}\sum_{t=1}^{c} y_{il}s_{ll}y_{jl}y_{jt}s_{tt}y_{it} \tag{32}$$

Since $\mathbf{Y} \in \Psi^{n\times c}$ is a cluster indicator matrix, we know that $y_{il}y_{it} = 1$ if and only if $l = t$, $y_{il}^2 = y_{il}$, and $y_{jl}^2 = y_{jl}$. Thus, Eq. (32) can be rewritten as

$$\begin{aligned}Tr\left(\mathbf{YSY}^T\mathbf{YSY}^T\right) &= \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{l=1}^{c} y_{il}s_{ll}^2y_{jl} \\ &= \sum_{l=1}^{c} s_{ll}^2 \left(\sum_{j=1}^{n} y_{il}\right)^2 \\ &= Tr\left(\mathbf{S}^2\mathbf{Y}^T\mathbf{11}^T\mathbf{Y}\right)\end{aligned} \tag{33}$$

Therefore, we have

$$\begin{aligned}&\max_{\mathbf{Y}\in\Psi^{n\times c},\mathbf{S}} 2Tr(\mathbf{SY}^T\mathbf{AY}) - Tr(\mathbf{YSY}^T\mathbf{YSY}^T) \\ &\Leftrightarrow \max_{\mathbf{Y}\in\Psi^{n\times c},\mathbf{S}} 2Tr(\mathbf{SY}^T\mathbf{AY}) - Tr(\mathbf{S}^2\mathbf{Y}^T\mathbf{11}^T\mathbf{Y}) \\ &\Leftrightarrow \max_{\mathbf{Y}\in\Psi^{n\times c},\mathbf{S}} \sum_{l=1}^{c} \mathbf{y}_l^T(2s_{ll}\mathbf{A} - s_{ll}^2\mathbf{11}^T)\mathbf{y}_l\end{aligned} \tag{34}$$

which completes the proof. □

C Proof of Theorem 3

Denote the objective function of EBMC in Eq. (13) as $\mathcal{P}(\mathbf{S}, \mathbf{Y})$, and the optimal solution of $\mathbf{S}$ and $\mathbf{Y}$ in the t-th and $(t+1)$-th iteration as $\mathbf{S}_t$, $\mathbf{Y}_t$ and $\mathbf{S}_{t+1}$, $\mathbf{Y}_{t+1}$. Before giving the proof of Theorem 3, we first give the following lemmas.

Lemma 1 *If* $\{\mathbf{M}^{(1)}, \ldots, \mathbf{M}^{(c)}\}$ *are all positive semi-definite matrices in the* t*-th iteration, the following inequation holds*

$$\mathcal{P}(\mathbf{S}_t, \mathbf{Y}_{t+1}) \geq \mathcal{P}(\mathbf{S}_t, \mathbf{Y}_t) + \eta\, \|\mathbf{Y}_{t+1} - \mathbf{Y}_t\|_2^2 \tag{35}$$

Proof Denote $\mathbf{Y}$ in the t-th and $(t+1)$-th iterations as $\mathbf{Y}_t$ and $\mathbf{Y}_{t+1}$ and $\mathbf{S}$. in the t-th and $(t+1)$-th iterations as $\mathbf{S}_t$ and $\mathbf{S}_{t+1}$, respectively. Since $\mathbf{Y}_{t+1}$ is the optimal solution to problem (16), we have

$$\begin{aligned}&\sum_{l=1}^{c}(\mathbf{y}_l)_{t+1}^T(\mathbf{M}^{(l)})_t(\mathbf{y}_l)_t - \frac{\eta}{2}\|\mathbf{Y}_{t+1} - \mathbf{Y}_t\|_F^2 \\ &\quad\geq \sum_{l=1}^{c}(\mathbf{y}_l)_t^T(\mathbf{M}^{(l)})_t(\mathbf{y}_l)_t\end{aligned} \tag{36}$$

Since the matrix $(\mathbf{M}^{(l)})_t$ is positive semi-definite, we can rewrite $(\mathbf{M}^{(l)})_t = (\mathbf{Q}_B^{(l)})^T\mathbf{Q}_B^{(l)}$ via Cholesky factorization. Then Eq. (36) can be rewritten as

$$\begin{aligned}&\sum_{l=1}^{c}(\mathbf{y}_l)_{t+1}^T(\mathbf{Q}_B^{(l)})^T\mathbf{Q}_B^{(l)}(\mathbf{y}_l)_{t+1} - \frac{\eta}{2}\|\mathbf{Y}_{t+1} - \mathbf{Y}_t\|_F^2 \\ &\quad\geq \sum_{l=1}^{c}(\mathbf{y}_l)_t^T(\mathbf{Q}_B^{(l)})^T\mathbf{Q}_B^{(l)}(\mathbf{y}_l)_t^T\end{aligned} \tag{37}$$

The inequation $\left\|\mathbf{Q}_B^{(l)}(\mathbf{y}_l)_{t+1} - \mathbf{Q}_B^{(l)}(\mathbf{y}_l)_t\right\|_F^2 \geq 0$ can be rewritten as

$$\begin{aligned}&(\mathbf{y}_l)_{t+1}^T(\mathbf{Q}_B^{(l)})^T\mathbf{Q}_B^{(l)}(\mathbf{y}_l)_{t+1} - 2(\mathbf{y}_l)_{t+1}^T(\mathbf{Q}_B^{(l)})^T\mathbf{Q}_B^{(l)}(\mathbf{y}_l)_t \\ &\quad+ (\mathbf{y}_l)_t^T(\mathbf{Q}_B^{(l)})^T\mathbf{Q}_B^{(l)}(\mathbf{y}_l)_t \geq 0\end{aligned} \tag{38}$$

Summarizing Eq. (38) over all l gives

$$\begin{aligned}&\sum_{l=1}^{c}(\mathbf{y}_l)_{t+1}^T(\mathbf{Q}_B^{(l)})^T\mathbf{Q}_B^{(l)}(\mathbf{y}_l)_{t+1} \\ &\quad-2\sum_{l=1}^{c}(\mathbf{y}_l)_{t+1}^T(\mathbf{Q}_B^{(l)})^T\mathbf{Q}_B^{(l)}(\mathbf{y}_l)_t \\ &\quad+\sum_{l=1}^{c}(\mathbf{y}_l)_t^T(\mathbf{Q}_B^{(l)})^T\mathbf{Q}_B^{(l)}(\mathbf{y}_l)_t \geq 0\end{aligned} \tag{39}$$

Multiplying Eq. (37) by 2 and summing over it and Eq. (39) gives

$$\begin{aligned}&\sum_{l=1}^{c}(\mathbf{y}_l)_{t+1}^T(\mathbf{Q}_B^{(l)})^T(\mathbf{Q}_B^{(l)})(\mathbf{y}_l)_{t+1}\\&\quad\geq\sum_{l=1}^{c}(\mathbf{y}_l)_t^T(\mathbf{Q}_B^{(l)})^T(\mathbf{Q}_B^{(l)})(\mathbf{y}_l)_t+\eta\left\|\mathbf{Y}_{t+1}-\mathbf{Y}_t\right\|_F^2\end{aligned} \tag{40}$$

which equals to

$$\begin{aligned}&\sum_{l=1}^{c}(\mathbf{y}_l)_{t+1}^T(\mathbf{M}^{(l)})_t(\mathbf{y}_l)_{t+1}\\&\quad\geq\sum_{l=1}^{c}(\mathbf{y}_l)_t^T(\mathbf{M}^{(l)})_t(\mathbf{y}_t)_l+\eta\left\|\mathbf{Y}_{t+1}-\mathbf{Y}_t\right\|_F^2\end{aligned} \tag{41}$$

According to the definition of $\mathbf{M}^{(l)}$ in Eq. (17), Eq. (41) can be further rewritten as

$$\begin{aligned}&2Tr(\mathbf{S}_t(\mathbf{Y})_{t+1}^T\mathbf{A}\mathbf{Y}_{t+1})-Tr((\mathbf{S})_t^2(\mathbf{Y})_{t+1}^T\mathbf{1}\mathbf{1}^T\mathbf{Y}_{t+1})\\&\quad\geq 2Tr(\mathbf{S}_t(\mathbf{Y})_t^T\mathbf{A}\mathbf{Y}_t)\\&\qquad-Tr((\mathbf{S})_t^2(\mathbf{Y})_t^T\mathbf{1}\mathbf{1}^T\mathbf{Y}_t)+\eta\left\|\mathbf{Y}_{t+1}-\mathbf{Y}_t\right\|_F^2\end{aligned} \tag{42}$$

which equals to

$$\mathcal{P}(\mathbf{S}_t,\mathbf{Y}_{t+1})\geq\mathcal{P}(\mathbf{S}_t,\mathbf{Y}_t)+\eta\left\|\mathbf{Y}_{t+1}-\mathbf{Y}_t\right\|_F^2 \tag{43}$$

□

Since $\mathbf{S}$ is the optimal solution to problem (19) and according to Theorem 1, we can verify the following lemma:

Lemma 2 *If* $\{\mathbf{M}^{(1)},\ldots,\mathbf{M}^{(c)}\}$ *are all positive semi-definite matrices in the* t*-th iteration,* $\mathcal{P}(\mathbf{S}_{t+1},\mathbf{Y}_{t+1})\geq\mathcal{P}(\mathbf{S}_t,\mathbf{Y}_t)+\eta\left\|\mathbf{Y}_{t+1}-\mathbf{Y}_t\right\|_2^2$.

In the following analysis, we omit $\mathbf{S}_{t+1}$ in $\mathcal{P}(\mathbf{S}_{t+1},\mathbf{Y}_{t+1})$ for simplification and give the following important lemma.

Lemma 3 *Suppose that* $\{\mathbf{M}^{(1)},\ldots,\mathbf{M}^{(c)}\}$ *are all positive semi-definite matrices after the* r*-th iteration. If we take* $\mathbf{Y}$ *as random variable and* $\mathbb{E}[\|\mathbf{Y}_{t+1}-\mathbf{Y}_t\|]$ *is the expectation of* $\mathbf{Y}_{t+1}-\mathbf{Y}_t$ *where* t *is the number of iterations, it holds that* $\lim_{t\to\infty}\mathbb{E}[\|\mathbf{Y}_{t+1}-\mathbf{Y}_t\|]=0$.

Proof According to Lemma 1, the following inequation holds for $i\geq r$

$$\mathcal{P}(\mathbf{Y}_{i+1})-\mathcal{P}(\mathbf{Y}_i)\geq\eta\left\|\mathbf{Y}_{i+1}-\mathbf{Y}_i\right\|_2^2 \tag{44}$$

which equals to

$$\left\|\mathbf{Y}_{i+1}-\mathbf{Y}_i\right\|_F^2\leq\frac{1}{\eta}\left[\mathcal{P}(\mathbf{Y}_{i+1})-\mathcal{P}(\mathbf{Y}_i))\right] \tag{45}$$

Given $t>r$, summing the above inequality over $i=r,\ldots,t-1$ gives

$$\sum_{i=r}^{t-1}\left\|\mathbf{Y}_{i+1}-\mathbf{Y}_i\right\|_F^2\leq\frac{1}{\eta}\left[\mathcal{P}(\mathbf{Y}_t)-\mathcal{P}(\mathbf{Y}_r)\right] \tag{46}$$

It can be verified that

$$\begin{aligned}\min_{i=r,\ldots,t-1}\left\|\mathbf{Y}_{i+1}-\mathbf{Y}_i\right\|_F^2&\leq\frac{1}{t-r}\sum_{i=r}^{t}\left\|\mathbf{Y}_{i+1}-\mathbf{Y}_i\right\|_F^2\\&\leq\frac{\mathcal{P}(\mathbf{Y}_t)-\mathcal{P}(\mathbf{Y}_r)}{(t-r)\eta}\end{aligned} \tag{47}$$

indicating that

$$\lim_{t\to\infty}\min_{i=r,\ldots,t-1}\left\|\mathbf{Y}_{i+1}-\mathbf{Y}_i\right\|_F^2=0 \tag{48}$$

which indicates that $\left\|\mathbf{Y}_{i+1}-\mathbf{Y}_i\right\|_F^2\to 0$ at some iteration t. Therefore, we have

$$\lim_{t\to\infty}\left[\left\|\mathbf{Y}_t-\mathbf{Y}_{t-1}\right\|_F^2\right]=0 \tag{49}$$

□

Finally, we prove Theorem 3 as follow:

Proof We first note that problem (13) has a finite number of n^c possible solutions since $\mathbf{Y}$ is a cluster indicator matrix. According to Lemma 3, we know that Algorithm 2 monotonically increases the objective function value of problem (13) in each iteration. Hence, the result follows. □

D Determination of η^u

Denote $\eta^u=\eta_1^u+\eta_2^u$ and rewrite $\mathbf{M}^l$ as $\mathbf{M}^l=\mathbf{M}_1^l+\mathbf{M}_2^l$. where

$$\mathbf{M}_1^l=2s_{ll}\mathbf{A}+\eta_1^u\mathbf{I}_n \tag{50}$$

and

$$\mathbf{M}_2^l=\eta_2^u\mathbf{I}_n-s_{ll}^2\mathbf{1}\mathbf{1}^T \tag{51}$$

It can verified that $\mathbf{M}^l$ is positive semi-definite if $\mathbf{M}_1^l$ and $\mathbf{M}_2^l$ are positive semi-definite.

Suppose the eigendecomposition of $\mathbf{A}$ is $\mathbf{A}=\mathbf{U}_A\Sigma_A\mathbf{U}_A^{-1}$, we have

$$\mathbf{M}_1^l=\mathbf{U}_A(2s_{ll}\Sigma_A+\lambda_1\mathbf{I}_n)\mathbf{U}_A^{-1} \tag{52}$$

We know that the diagonal elements in the diagonal matrix $2s_{ll}\Sigma_A+\lambda_1\mathbf{I}_n$ are eigenvalues of $\mathbf{M}_1^l$. To make $\mathbf{M}_1^l$ positive semi-definite, we only need to make all of its eigenvalues non-negative. Therefore, we can set λ_1 such that the following inequation holds for $\forall i$

$$\eta_1^u \geq -2 s_{ll} \sigma_i(\Sigma_A) \tag{53}$$

where $\sigma_i(\Sigma_A)$ is the i-th eigenvalue of Σ_A. The proper η_1^u can be set to the maximal value of $-2 s_{ll} \sigma_i(\Sigma_A)$. Since $\mathbf{A}$ is normalized and $a_{ii} = 0$, we know that $|\sigma(\Sigma_A)| \leq 1$ according to the Gershgorin circle theorem. Therefore, we can set $\eta_1^u = 2 \max_l s_{ll}$ in order to make $\mathbf{M}_1$ positive semi-definite.

On the other hand, $\mathbf{M}_2^l$ can be rewritten as

$$\mathbf{M}_2^l = \mathbf{V} \left(\eta_2^u \mathbf{I}_n - s_{ll}^2 \begin{bmatrix} n & \cdots & \cdots \\ \cdots & \cdots & \cdots \\ \cdots & \cdots & 0 \end{bmatrix} \right) \mathbf{V}^{-1} \tag{54}$$

where $\mathbf{V}$ consists of the eigenvectors of $\mathbf{1}\mathbf{1}^T$. Since $\mathbf{M}_2^l$ is positive semi-definite if and only if all of its eigenvalues are non-negative, which leads to $\eta_2^u - n s_{ll}^2 \geq 0$ and $\eta_2^u - 0 \geq 0$. Therefore, we can set $\eta_2^u = \max_l n s_{ll}^2$ to make $\mathbf{M}_2$ positive semi-definite.

Finally, we reach the upper bound of η as follow

$$\eta^u = \max_l (2 s_{ll} + n s_{ll}^2) \tag{55}$$

η^u can be updated after updating $\mathbf{S}$ in each iteration.

References

Bie, T. D., & Cristianini, N. (2006). Fast SDP relaxations of graph cut clustering, transduction, and other combinatorial problems. *Journal of Machine Learning Research*, *7*, 1409–1436.

Bühler, T., & Hein, M. (2009). Spectral clustering based on the graph p-Laplacian. In *Proceedings of the 26th international conference on machine learning* (pp. 81–88).

Cai, X., Nie, F., Huang, H.,& Kamangar, F. (2011). Heterogeneous image feature integration via multi-modal spectral clustering. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 1977–1984). IEEE.

Chen, X., Hong, W., Nie, F., He, D., Yang, M.,& Huang, J. Z. (2018). Spectral clustering of large-scale data by directly solving normalized cut. In *Proceedings of the ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 1206–1215).

Chen, X., Huang, J. Z., Nie, F., Chen, R.,& Wu, Q. (2017). A self-balanced min-cut algorithm for image clustering. In *Proceedings of the international conference on computer vision* (pp. 2080–2088).

Chen, X., Nie, F., Huang, J. Z., & Yang, M. (2017). Scalable normalized cut with improved spectral rotation. In *Proceedings of the twenty-sixth international joint conference on artificial intelligence* (pp. 1518–1524).

Chen, X., Xu, X., Ye, Y., & Huang, J. Z. (2013). TW-k-means: Automated two-level variable weighting clustering algorithm for multiview data. *IEEE Transactions on Knowledge and Data Engineering*, *25*(4), 932–944.

Chen, X., Yang, M., Huang, J. Z., & Zhong, M. (2018). TWCC: Automated two-way subspace weighting partitional co-clustering. *Pattern Recognition*, *76*, 404–415.

Chen, X., Ye, Y., Xu, X., & Huang, J. Z. (2012). A feature group weighting method for subspace clustering of high-dimensional data. *Pattern Recognition*, *45*(1), 434–446.

de Souto, M. C., Costa, I. G., de Araujo, D. S., Ludermir, T. B., & Schliep, A. (2008). Clustering cancer gene expression data: A comparative study. *BMC Bioinformatics*, *9*(1), 497.

Elhamifar, E., & Vidal, R. (2013). Sparse subspace clustering: Algorithm, theory, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(11), 2765–2781.

Ester, M., Kriegel, H. P., & Xu, X. (1998). Density-based clustering in spatial databases: The algorithm GDBscan and its applications. *Data Mining and Knowledge Discovery*, *2*(2), 169–194.

Hagen, L., & Kahng, A. B. (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.

Huang, J., Nie, F.,& Hu, H. (2013). Spectral rotation versus k-means in spectral clustering. In *AAAI conference on artificial intelligence* (pp. 431–437).

Hull, J. J. (2002). Database for handwritten text recognition research. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, *16*(5), 550–554.

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

Lu, C., Yan, S., & Lin, Z. (2016). Convex sparse spectral clustering: Single-view to multi-view. *IEEE Transactions on Image Processing*, *25*(6), 2833–2843.

Ng, A. Y., Jordan, M. I., Weiss, Y., et al. (2002). On spectral clustering: Analysis and an algorithm. *Advances in Neural Information Processing Systems*, *2*, 849–856.

Nie, F., Wang, X., Jordan, M.,& Huang, H. (2016). The constrained Laplacian rank algorithm for graph-based clustering. In *Proceedings of the thirtieth AAAI conference on artificial intelligence* (pp. 1969–1976).

Nie, F., Wang, X.,& Huang, H. (2014). Clustering and projected clustering with adaptive neighbors. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 977–986). ACM.

Nie, F., Zeng, Z., Tsang, I. W., Xu, D., & Zhang, C. (2011). Spectral embedded clustering: A framework for in-sample and out-of-sample spectral clustering. *IEEE Transactions on Neural Networks*, *22*(11), 1796–808.

Sen, P., Namata, G., Bilgic, M., Getoor, L., Galligher, B., & Eliassi-Rad, T. (2008). Collective classification in network data. *AI Magazine*, *29*(3), 93–106.

Shi, J., & Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*(8), 888–905.

Von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and Computing*, *17*(4), 395–416.

Yu, S. X.,& Shi, J. (2003). Multiclass spectral clustering. In *Proceedings of IEEE international conference on computer vision* (vol. 1, pp. 313–319).