LABIN: Balanced Min Cut for Large-Scale Data

Xiaojun Chen, *Member, IEEE*, Renjie Chen, Qingyao Wu, Yixiang Fang, Feiping Nie, and Joshua Zhexue Huang

***Abstract*— Although many spectral clustering algorithms have been proposed during the past decades, they are not scalable to large-scale data due to their high computational complexities. In this paper, we propose a novel spectral clustering method for large-scale data, namely, large-scale balanced min cut (LABIN). A new model is proposed to extend the self-balanced min-cut (SBMC) model with the anchor-based strategy and a fast spectral rotation with linear time complexity is proposed to solve the new model. Extensive experimental results show the superior performance of our proposed method in comparison with the state-of-the-art methods including SBMC.**

***Index Terms*— Clustering, graph cut, large-scale data, spectral clustering.**

I. Introduction

CLUSTERING is a fundamental research area and many algorithms have been proposed for cluster analysis during the past decades, including spectral clustering [1], subspace clustering [2], [3], multiview clustering [4], [5], coclustering [6], [7], and so on. Among them, spectral clustering is popular because it is easy to implement and often shows good clustering performance. Popular spectral clustering algorithms include such as ratio cut (RCut) [8], normalized cut (NCut) [9], multiclass spectral clustering (MSC) [10], spectral embedded clustering [11], clustering with adaptive neighbor (CAN) [12], constrained Laplacian rank (CLR) [13], self-balanced min cut (SBMC) [14], and many other methods [15]–[18]. They have shown their power in many applications, such as image clustering [14], image segmentation [10], [19], and clustering gene expression data [20].

Manuscript received May 7, 2018; revised December 16, 2018 and March 22, 2019; accepted April 1, 2019. Date of publication May 10, 2019; date of current version February 28, 2020. This work was supported in part by NSFC under Grant 61773268, Grant 61836005, and Grant 61876208, in part by the Shenzhen Research Foundation for Basic Research, China, under Grant JCYJ20180305124149387, in part by the National Engineering Laboratory for Big Data System Computing Technology, Guangdong Provincial Scientific and Technological funds 2017B090901008 and 2018B010108002, in part by Pearl River S&T Nova Program of Guangzhou under Grant 201806010081, and in part by the CCF-Tencent Open Research Fund RAGR20190103. *(Corresponding authors: Qingyao Wu; Feiping Nie.)*

X. Chen and J. Z. Huang are with the College of Computer Science and Software, Shenzhen University, Shenzhen 518060, China (e-mail: xjchen@szu.edu.cn; zx.huang @szu.edu.cn).

R. Chen and Q. Wu are with the School of Software Engineering, South China University of Technology, Guangzhou 510630, China (e-mail: sechenrenjie@mail.scut.edu.cn; qyw@scut.edu.cn).

Y. Fang is with the Department of Computer Science, The University of Hong Kong, Hong Kong (e-mail: yxfang@cs.hku.hk).

F. Nie is with the School of Computer Science, Northwestern Polytechnical University, Xi'an 710072, China, and also with the Center for OPTIMAL, Northwestern Polytechnical University, Xi'an 710072, China (e-mail: feipingnie@gmail.com).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2019.2909425

Spectral clustering methods usually perform eigendecomposition of the similarity matrix first, and then use k-means or spectral rotation [10] to obtain the final clustering results from eigenvectors. Moreover, spectral clustering usually has a high time complexity of $O(n^3)$ where n is the number of objects due to the high time cost of eigendecomposition. To solve the scalability problem, researchers have proposed three types of methods. The first type is to reduce the time cost of the eigendecomposition step [21]–[23]. For example, Fowlkes *et al.* [21] applied the Nyström method to accelerate the eigendecomposition by randomly sampling a small submatrix from the original matrix and compute an approximation of the eigenvectors of the original matrix. Li *et al.* [22] proposed a scalable Nyström scheme by using the randomized low-rank matrix approximation algorithms. Chen *et al.* [23] developed a parallel spectral clustering package on distributed environments. The second type is to use preprocessing to reduce the data size and perform clustering on the reduced data [24], [25]. For example, Yan *et al.* [24] proposed the k-means-based approximate spectral clustering (KASP) method, which employs k-means to partition the data set into a large number of clusters and then partition these cluster centers with the traditional spectral clustering. Shinnou and Sasaki [25] proposed committees-based spectral clustering (CSC), which also partitions the data set into a large number of clusters with k-means, but then the cluster centers and the data points closed to the cluster centers are used for the subsequent clustering. However, both types of methods are based on sampling and will ignore a lot of information. The third type is to use anchor-based strategy to construct a bipartite similarity matrix and perform clustering on the bipartite similarity matrix instead of the full similarity matrix. For example, Cai and Chen [26] proposed a landmarks-based spectral clustering (LSC), which computes a bipartite similarity matrix instead of the full similarity matrix, and then performs the eigendecomposition on the low-size similarity matrix and the final clustering result is obtained by clustering the eigenvectors with k-means. Chen *et al.* [27] proposed a scalable normalized cut (SNC) where the anchor-based strategy is used to accelerate the improved spectral rotation (ISR).

Recently, Chen *et al.* [14] proposed a new spectral clustering algorithm, namely, SBMC. In the new method, an SBMC model is proposed in which a balance regularization is implicitly used in order to obtain a balanced partition. According to the analysis in [14, Sec. 5.4], the new model can simultaneously minimize the graph cut and balance the partition across all clusters. In Section II-E, we will compare SBMC with traditional NCut and point our that SBMC can be considered as a new type of robust NCut.

2162-237X © 2019 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: SHENZHEN UNIVERSITY. Downloaded on April 14,2023 at 16:50:09 UTC from IEEE Xplore. Restrictions apply.

An iterative algorithm with linear computational complexity is proposed to solve the new model. The new method was proven to be able to simultaneously minimize the graph cut and balance the partition. However, their method requires a full $n \times n$ affinity matrix as input that hinders its usage in large-scale data. SBMC employs augmented Lagrangian method (ALM) to solve the cluster indicator matrix and it is difficult to adjust the two parameters in the ALM. Moreover, their optimization method depends on the initial partition and our experimental results show that it will become unstable when dealing with large-scale data since it is difficult to generate proper initial partition for large-scale data.

In this paper, we extend SBMC to large-scale data by proposing a large-scale balanced min cut (LABIN). To reduce the size of input similarity matrix, we use the anchor-based strategy to construct a bipartite similarity matrix and perform computing on the bipartite similarity matrix instead of the full $n \times n$ similarity matrix. We use the ISR to optimize the new model and propose an efficient method to perform eigendecomposition with linear time computational complexity. It is worthwhile to highlight the following contributions.

1) Compared with the original SBMC, the new method has significant reduction on space computational complexity and is more robust due to the use of ISR.
2) Compared with existing anchor-based spectral clustering [26], [27], which are based on the traditional spectral clustering models, our method can uncover high-quality cluster structure from complex data sets that consist of noise or isolate samples.
3) Extensive experimental results show that LABIN outperforms the state-of-the-art clustering methods.

The rest of this paper is organized as follows. Section II presents the notations and definitions and gives a brief review of related work. LABIN is proposed in Section III. The experimental results on both synthetic and real-world data sets are reported in Sections IV and V, respectively. Conclusions and future work are given in Section VI.

II. Background and Related Work

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

A. Notations and Definitions

Matrices are written as boldface uppercase letters. Vectors are written as boldface lowercase letters. For matrix $\mathbf{M} = (m_{ij})$, its ith row is denoted as $\mathbf{m}^i$, and its jth column is denoted by $\mathbf{m}_j$. The Frobenius norm of the matrix $\mathbf{M} \in \mathbb{R}^{n\times m}$ is defined as $\|\mathbf{M}\|_F = \{\sum_{i=1}^{n}\sum_{j=1}^{m} m_{ij}^2\}^{1/2}$. We denote the set of all cluster indicator matrices as Ψ, in which each row contains one and only one element equal to 1 to indicate the cluster membership, whereas the rest elements are 0.

B. Spectral Clustering

Given an affinity matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ constructed from a data set $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$. Let $\mathbf{Y} \in \Psi^{n\times c}$ be the cluster indicator matrix. The objective function of the classical RCut can be written as [8]

$$\min_{\mathbf{Y}^T\mathbf{Y}=\mathbf{I}} \mathrm{Tr}(\mathbf{Y}^T\mathbf{L}_A\mathbf{Y}) \tag{1}$$

and the objective function of NCut is as follows [9]:

$$\min_{\mathbf{Y}^T\mathbf{D}\mathbf{Y}=\mathbf{I}} \mathrm{Tr}(\mathbf{Y}^T\mathbf{L}_A\mathbf{Y}) \tag{2}$$

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 ith diagonal element as $d_{ii} = \sum_{j=1}^{n} a_{ij}$. Problems (1) and (2) can be solved by a two-stage process: performing eigendecomposition on $\mathbf{L}_A$ first, and then obtaining the final clustering result by clustering eigenvectors with k-means or spectral rotation [10].

Nie *et al.* [12] proposed a new spectral clustering method CANs. CAN aims to learn a probability matrix $\mathbf{S} \in \mathbb{R}^{n\times n}$ where s_{ij} is the connected probability between $\mathbf{x}_i$ and $\mathbf{x}_j$ from the following objective function:

$$\begin{aligned} &\min_{\mathbf{S}} \sum_{i,j=1}^{n} \left(\|\mathbf{x}_i - \mathbf{x}_j\|_2^2 s_{ij} + \gamma\, s_{ji}^2\right) \\ &\text{s.t. } \mathbf{S1} = \mathbf{1}, \quad s_{ij} \in [0,1],\ \mathrm{rank}(\mathbf{L}_S) = n - c \end{aligned} \tag{3}$$

where a rank constraint $\mathrm{rank}(L_S) = n - c$ is imposed to the Laplacian matrix of $\mathbf{S}$ such that the connected components in $\mathbf{S}$ are exactly equal to the number of clusters c and the final clustering result can be obtained from these connected components. Problem (3) can be solved with an iterative method, in which the eigendecomposition is performed on $\mathbf{L}_S$ in each iteration.

Nie *et al.* [13] further improved CAN by proposing the CLR. Given an initial affinity matrix $\mathbf{A}$, CLR learns $\mathbf{S} \in \mathbb{R}^{n\times n}$ that best approximates $\mathbf{A}$. Two versions of CLR were proposed, one with the ℓ_2 norm

$$\begin{aligned} &\min_{\mathbf{S}} \|\mathbf{S} - \mathbf{A}\|_F^2 \\ &\text{s.t. } \mathbf{S1} = \mathbf{1},\ s_{ij} \in [0,1],\ \mathrm{rank}(\mathbf{L}_S) = n - c \end{aligned} \tag{4}$$

and the other one with the ℓ_1 norm

$$\begin{aligned} &\min_{\mathbf{S}} \|\mathbf{S} - \mathbf{A}\|_1 \\ &\text{s.t. } \forall\, i,\ \mathbf{s}^i\mathbf{1} = 1,\ s_{ij} \in [0,1],\ \mathrm{rank}(\mathbf{L}_S) = n - c. \end{aligned} \tag{5}$$

The above-mentioned two problems can be solved with the iterative methods, in which eigendecomposition on $\mathbf{L}_S$ is performed in each iteration. Therefore, CAN and CLR are time-consuming since they involve multiple eigendecompositions.

C. Spectral Rotation

Yu and Shi [10] proposed an MSC, in which a spectral rotation is proposed to solve the following k-way NCut problem:

$$\max_{\mathbf{Y}\in\Psi^{n\times c},\ \mathbf{Z}=\mathbf{Y}(\mathbf{Y}^T\mathbf{D}_A\mathbf{Y})^{-\frac{1}{2}}} \mathrm{Tr}(\mathbf{Z}^T\mathbf{A}\mathbf{Z}) \tag{6}$$

where $\mathbf{Z}$ is the scaled partition matrix. It is difficult to directly solve problem (6). A commonly used method is to relax $\mathbf{Z}$

Authorized licensed use limited to: SHENZHEN UNIVERSITY. Downloaded on April 14,2023 at 16:50:09 UTC from IEEE Xplore. Restrictions apply.

from the discrete values to the continuous ones and form the following new problem:

$$\max_{\mathbf{Z}^T\mathbf{D}_A\mathbf{Z}=\mathbf{I}_c,\ \mathbf{Z}=\mathbf{Y}(\mathbf{Y}^T\mathbf{D}_A\mathbf{Y})^{-\frac{1}{2}}} \mathrm{Tr}(\mathbf{Z}^T\mathbf{A}\mathbf{Z}). \tag{7}$$

According to Proposition 1 in [10], the optimal solution of $\mathbf{Z}$ is $\{\mathbf{Z}^*\mathbf{R} : \mathbf{R}^T\mathbf{R} = \mathbf{I}_c\}$ where $\mathbf{Z}^* \in \mathbb{R}^{n\times c}$ consists of c column vectors of the eigenvectors of $\mathbf{D}_A^{-1}\mathbf{A}$ which correspond to c biggest eigenvalues.

To obtain the discrete clustering assignment matrix $\mathbf{Y}$, they first compute an approximate solution $\mathbf{Y}^* = \mathrm{Diag}(\mathbf{Z}^*(\mathbf{Z}^*)^T)^{-1/2}\mathbf{Z}^*$, then $\mathbf{Y}$ can be learned by solving the following problem:

$$\min_{\mathbf{Y}\in\Psi^{n\times c},\ \mathbf{R}\in\mathbb{R}^{c\times c},\ \mathbf{Y}\mathbf{1}_{c\times 1}=\mathbf{1}_{n\times 1},\ \mathbf{R}^T\mathbf{R}=\mathbf{I}_c} \|\mathbf{Y} - \mathbf{Y}^*\mathbf{R}\|_F^2. \tag{8}$$

However, the final discrete solution $\mathbf{Y}$ may deviate from the result by directly optimizing the original objective function since $\mathbf{Y}^*$ is an approximate solution. Chen *et al.* [27] proposed an ISR to solve problem (6). They first relax problem (7) to the following continuous problem:

$$\max_{\mathbf{F}^T\mathbf{F}=\mathbf{I},\ \mathbf{F}=\mathbf{D}_A^{\frac{1}{2}}\mathbf{Y}(\mathbf{Y}^T\mathbf{D}_A\mathbf{Y})^{-\frac{1}{2}}} \mathrm{Tr}\left(\mathbf{F}^T\mathbf{D}_A^{-\frac{1}{2}}\mathbf{A}\mathbf{D}_A^{-\frac{1}{2}}\mathbf{F}\right). \tag{9}$$

Then the optimal solution of $\mathbf{F}$ becomes $\{\mathbf{F}^*\mathbf{R} : \mathbf{R}^T\mathbf{R} = \mathbf{I}_c\}$ where $\mathbf{F}^* \in \mathbb{R}^{n\times c}$ is the c column vectors of the eigenvectors of $\mathbf{D}_A^{-1/2}\mathbf{A}\mathbf{D}_A^{-1/2}$ that correspond to the c biggest eigenvalues. With $\mathbf{F}^*$, they proposed to directly obtain the discrete solution $\mathbf{Y}$ by minimizing $\|\mathbf{D}_A^{1/2}\mathbf{Y}(\mathbf{Y}^T\mathbf{D}_A\mathbf{Y})^{-1/2} - \mathbf{F}^*\mathbf{R}\|_F^2$ from the following problem:

$$\max_{\mathbf{Y}\in\Psi^{n\times c},\ \mathbf{R}\in\mathbb{R}^{c\times c},\ \mathbf{Y}\mathbf{1}_c=\mathbf{1}_{n\times 1},\ \mathbf{R}^T\mathbf{R}=\mathbf{I}_c} \mathrm{Tr}((\mathbf{Y}^T\mathbf{D}_A\mathbf{Y})^{-\frac{1}{2}}\mathbf{Y}^T\mathbf{D}_A^{\frac{1}{2}}\mathbf{F}^*\mathbf{R}). \tag{10}$$

Then they proposed an alternative method to solve problem (10). To cope with large-scale data, they further incorporated the anchor-based strategy to extend ISR to an SNC for large-scale data. Given a data set with n objects, they first use k-means to find $m \ll n$ representative data points and construct a low-size $n\times m$ representative similarity matrix, then the eigendecomposition can be performed on the representative similarity matrix. Finally, ISR is used to obtain the final clustering assignments.

D. Anchor-Based Spectral Clustering

To handle the scalability problem of spectral clustering, Liu *et al.* [28] proposed an anchor-based strategy, which is also called landmarks-based method [26]. Given a data set $\mathbf{X} \in \mathbb{R}^{d\times n}$ with n objects and d features, the anchor-based strategy first seeks m anchors to represent the whole n objects ($m \ll n$), and then construct a sample-anchor similarity matrix by calculating the distance between anchors and original objects. There are mainly two methods for anchor generation, i.e., random selection and k-means generation. k-means is preferred for anchor generation since clustering centers have a stronger representation power than random selected data [26], [28].

After we have obtained m anchors $\mathbf{W} \in \mathbb{R}^{d\times m}$, the next step is to compute a sample-anchor similarity matrix $\mathbf{B} \in \mathbb{R}^{n\times m}$ such that $\mathbf{B}\mathbf{1} = \mathbf{1}$. With $\mathbf{B}$, we can obtain an affinity matrix $\mathbf{A}$ as [28]

$$\mathbf{A} = \mathbf{B}\Delta^{-1}\mathbf{B}^T \tag{11}$$

where $\Delta \in \mathbb{R}^{m\times m}$ is a diagonal matrix and the jth diagonal entry is defined as $\Delta_{jj} = \sum_{i=1}^n b_{ij}$. $\mathbf{A}$ can be rewritten as

$$\mathbf{A} = \mathbf{P}\mathbf{P}^T \tag{12}$$

where $\mathbf{P} \in \mathbb{R}^{n\times m} = \mathbf{B}\Delta^{-1/2}$.

Liu *et al.* [28] proposed a local anchor embedding (LAE) method to learn each $\mathbf{b}_i$ for $\mathbf{x}_i \in \mathbf{X}$ by solving the following problem:

$$\min_{\mathbf{b}_i^T\mathbf{1}=1,\mathbf{b}_i\geq 0} \frac{1}{2}\|\mathbf{x}_i - \mathbf{U}_k(\mathbf{x}_i)\mathbf{b}_i\|_2^2 \tag{13}$$

where the k column vectors of $\mathbf{U}_k(\mathbf{x}_i) \in \mathbf{R}^{d\times k}$ are the k nearest anchors of $\mathbf{x}_i$. With the learned bipartite graph $\mathbf{B}$ according to (13), they have proposed a semi-supervised learning method AnchorGraphReg [28].

Cai and Chen [26] proposed to obtain sparse $\mathbf{B}$ and anchors $\mathbf{W}$ simultaneously by solving the following sparse coding problem:

$$\min_{\mathbf{W},\mathbf{B}} \frac{1}{2}\|\mathbf{X} - \mathbf{W}\mathbf{B}\|_2^2 + \gamma\|\mathbf{B}\|_{2,1} \tag{14}$$

where γ is a regularization parameter. However, it is time-consuming to solve problem (14). Therefore, they proposed to use Nadaraya–Watson kernel regression to directly compute the representation matrix $\mathbf{B}$ as follows:

$$b_{ij} = \begin{cases} \dfrac{K_h(\mathbf{x}_i - \mathbf{u}_j)}{\sum_{l=1}^m K_h(\mathbf{x}_i - \mathbf{u}_l)} & \mathbf{x}_j \in \mathcal{N}_k(\mathbf{x}_i) \\ 0 & \text{otherwise} \end{cases} \tag{15}$$

where $\mathcal{N}_k(\mathbf{x}_i)$ contains the k nearest neighbors of $\mathbf{x}_i$ and $\mathbf{u}_l$ is the lth anchor. $K_h(\mathbf{x}_i - \mathbf{u}_j)$ is the Gaussian kernel with a bandwidth h. However, b_{ij} in (15) is sensitive to the parameter in the Gaussian kernel and lacks a meaningful interpretation. With the learned bipartite graph $\mathbf{B}$, they have improved the classical NCut with the anchor-based strategy and proposed LSC for clustering of large-scale data [26].

Chen *et al.* [27] proposed to compute the sample-anchor similarity matrix $\mathbf{B}$ as follows:

$$b_{ij} = \begin{cases} \dfrac{d_{i,k+1} - \|\mathbf{x}_i - \mathbf{w}_j\|_2^2}{kd_{i,k+1} - \sum_{h=1}^k d_{i,h}} & \mathbf{x}_j \in \mathcal{N}_k(\mathbf{x}_i) \\ 0 & \text{otherwise} \end{cases} \tag{16}$$

where $d_{i,h}$ is the square of Euclidean distance between $\mathbf{x}_i$ and its hth nearest neighbor, and $\mathcal{N}_k(\mathbf{x}_i)$ consists of the k nearest neighbors of $\mathbf{x}_i$. They further proved that with $\mathbf{B}$ computed according to (16), the full affinity matrix $\mathbf{A}$ computed from (12) is symmetric and doubly stochastic. With the new method, they have improved the classical multiclass NCut and proposed SNC for clustering of large-scale data [27].

Authorized licensed use limited to: SHENZHEN UNIVERSITY. Downloaded on April 14,2023 at 16:50:09 UTC from IEEE Xplore. Restrictions apply.

E. Self-Balanced Min Cut

Chen *et al.* [14] proposed an SBMC algorithm to simultaneously minimize the graph cut and balance the partition across all clusters. In their method, they have proposed the following objective function:

$$\min_{\mathbf{Y}\in\Psi^{n\times c},s} \|\mathbf{A} - s\mathbf{Y}\mathbf{Y}^T\|_F^2 \tag{17}$$

where $s > 0$ is a balance parameter. An iterative method is proposed to solve the above problem.

To effectively solve the above problem, they first rewrite problem (17) as a new problem that is much easier to solve

$$\max_{\mathbf{Y}\in\Psi^{n\times c},s>0} 2s\mathrm{Tr}(\mathbf{Y}^T\mathbf{A}\mathbf{Y}) - s^2\|\mathbf{Y}\|_b \tag{18}$$

where $\|\mathbf{Y}\|_b$ is the balance regularization used to obtain balanced partition which is defined as [14]

$$\|\mathbf{Y}\|_b = \sum_{j=1}^{c}\left(\sum_{i=1}^{n} y_{ij}\right)^2 = \mathrm{Tr}\left(\mathbf{Y}\mathbf{1}_n\mathbf{1}_n^T\mathbf{Y}^T\right). \tag{19}$$

Then they proposed an alternative optimization approach to solve problem (18). If $s > 0$ is fixed, the optimal solution of $\mathbf{Y}$ is obtained by solving the following problem with ALM:

$$\min_{\mathbf{Y}\in\Psi^{n\times c},\mathbf{G}=\mathbf{Y}} \mathrm{Tr}(\mathbf{Y}^T\Theta\mathbf{G}) \tag{20}$$

where $\Theta = s/2\mathbf{1}_{n\times 1}\mathbf{1}_{n\times 1}^T - \mathbf{A}$.

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

$$s = \frac{\mathrm{Tr}(\mathbf{Y}^T\mathbf{A}\mathbf{Y})}{\|\mathbf{Y}\|_b}. \tag{21}$$

The detailed algorithm to solve problem (18) is summarized in Algorithm 1.

Algorithm 1 SBMC Algorithm to Solve Problem (18)

1: **Input:** An affinity matrix $\mathbf{A}$, parameter $\rho \in (1, 2)$.
2: **repeat**
3: Update $s = \frac{Tr(\mathbf{Y}^T\mathbf{A}\mathbf{Y})}{\|\mathbf{Y}\|_b}$.
4: Initialize $\mu > 0$ and $\Lambda \in \mathcal{R}^{n\times c}$.
5: Randomly initialize $\mathbf{Y}$.
6: **repeat**
7: Update $\Theta = \frac{s}{2}\mathbf{1}_{n\times 1}\mathbf{1}_{n\times 1}^T - \mathbf{A}$.
8: Update $\mathbf{G} = \mathbf{Y} - \frac{1}{\mu}(\Theta^T\mathbf{Y} - \Lambda)$.
9: Update $\mathbf{Y}$ as $y_{ij} =< j = \arg\max_{j'\in[1,c]} \mathbf{G}_{ij} - \frac{\Omega_{ij'}+\Lambda_{ij'}}{\mu} >$ where $< . >$ is 1 if the argument is true or 0 otherwise.
10: Update $\Lambda = \Lambda + \mu(\mathbf{Y} - \mathbf{G})$ and $\mu = \rho\mu$.
11: **until** problem (20) converges
12: **until** problem (18) converges
13: **Output:** The clustering result $\mathbf{Y}$.

Problem (18) can be rewritten as

$$\max_{\mathbf{Y}\in\Psi^{n\times c},s} s\left(\mathrm{Tr}(\mathbf{Y}^T\mathbf{A}\mathbf{Y}) - \frac{s}{2}\|\mathbf{Y}\|_b\right). \tag{22}$$

They have pointed that solving the above-mentioned problem will automatically adjust the balance parameter s according to the learned cluster indicator matrix $\mathbf{Y}$. On the other hand, maximizing s will maximize $\mathrm{Tr}(\mathbf{Y}^T\mathbf{A}\mathbf{Y})$, which is the objective function of min-cut. Therefore, SBMC can simultaneously minimize the graph cut and balance the partition across all clusters.

Note that the classical NCut problem is

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

where $\mathbf{y}_l^T\mathbf{D}_A\mathbf{y}_l$ is used to improve its robustness to isolated nodes [19].

Problem (23) can be further rewritten as

$$\min_{\mathbf{Y}\in\Psi^{n\times c}} \sum_{l=1}^{c} 1 - \frac{\mathbf{y}_l^T\mathbf{A}\mathbf{y}_l}{\mathbf{y}_l^T\mathbf{D}_A\mathbf{y}_l} \Longleftrightarrow \max_{\mathbf{Y}\in\Psi^{n\times c}} \sum_{l=1}^{c} \frac{\mathbf{y}_l^T\mathbf{A}\mathbf{y}_l}{\mathbf{y}_l^T\mathbf{D}_A\mathbf{y}_l}. \tag{24}$$

The normalization term in problem (24) is cluster degree $\mathbf{y}_l^T\mathbf{D}_A\mathbf{y}_l$, which will be small if the lth cluster consists of noise or isolate samples, and lead to low-quality cluster.

Substituting s in (21) into problem (18) gives

$$\max_{\mathbf{Y}\in\Psi^{n\times c}} \frac{(\mathbf{Y}^T\mathbf{A}\mathbf{Y})^2}{\|\mathbf{Y}\|_b}. \tag{25}$$

Here, we can see that problem (25) is similar to problem (24) since both of them aim to maximize the normalized within cluster similarities. However, in problem (25), maximizing $1/\|\mathbf{Y}\|_b$ will produce balanced partition that is insensitive to the noise or isolate samples. Therefore, SBMC can be considered as a new type of robust NCut. Moreover, compared with problems (25) and (24), problem (18) is easier to interpret and optimize, in which the balance parameter is automatically learned.

III. LABIN: Balanced Min Cut for Large-Scale Data

Suppose we want to cluster $\mathbf{X}$ into c clusters and let $\mathbf{Y} \in \Psi^{n\times c}$ be the cluster indicator matrix. We first construct an affinity matrix $\mathbf{A}$ according to (12) in which the sample-anchor similarity matrix $\mathbf{B}$ is constructed according to (16). Substituting the affinity matrix $\mathbf{A}$ in (12) into (18) gives the following new objective function:

$$\max_{\mathbf{Y}\in\Phi^{n\times c},s>0} 2s\mathrm{Tr}(\mathbf{Y}^T\mathbf{P}\mathbf{P}^T\mathbf{Y}) - s^2\|\mathbf{Y}\|_b \tag{26}$$

where $s > 0$ is the balance parameter, $\mathbf{P} \in \mathbb{R}^{n\times m} = \mathbf{B}\Delta^{-1/2}$ and $\Delta \in \mathbb{R}^{m\times m}$ is a diagonal matrix in which the jth diagonal entry is defined as $\Delta_{jj} = \sum_{i=1}^{n} b_{ij}$.

Problem (26) can be solved with an alternative optimization approach. In the following, we show how to update $\mathbf{Y}$ and s, respectively.

A. Solving Y With s Fixed

If s is fixed, problem (26) can be rewritten as

$$\max_{\mathbf{Y}\in\Phi^{n\times c}} \mathrm{Tr}(\mathbf{Y}^T\Theta\mathbf{Y}) \tag{27}$$

where $\Theta = \mathbf{P}\mathbf{P}^T - s/2 \times \mathbf{1}_n\mathbf{1}_n^T$.

In [14], the above problem can be solved with ALM. However, our experimental results show that this method often

Authorized licensed use limited to: SHENZHEN UNIVERSITY. Downloaded on April 14,2023 at 16:50:09 UTC from IEEE Xplore. Restrictions apply.

performs badly on large-scale data since it is difficult to adjust the two parameters in the ALM and generate proper initial partition for large-scale data. In this paper, we use the ISR to solve the above problem. Specifically, we first relax the matrix $\mathbf{Y}$ from the discrete values to the continuous ones and form the following problem:

$$\max_{\mathbf{Y}^T\mathbf{Y}=\mathbf{I}_c} \mathrm{Tr}(\mathbf{Y}^T\Theta\mathbf{Y}). \tag{28}$$

According to Proposition 1 in [10], the optimal solution of $\mathbf{Y}$ is $\{\mathbf{Y}^*\mathbf{R} : \mathbf{R}^T\mathbf{R} = \mathbf{I}_c\}$ where $\mathbf{Y}^* \in \mathbb{R}^{n\times c}$ is the c column vectors of the eigenvectors of Θ which correspond to c biggest eigenvalues. According to Theorem 1 in [27], then $\mathbf{Y}^* \in \mathbb{R}^{n\times c}$ is the c column vectors of the eigenvectors of Θ which correspond to its c biggest eigenvalues. Since Θ is a $n \times n$ matrix, directly performing eigendecomposition of Θ takes $O(n^3)$ time. In the following, we propose an efficient method to solve the eigenvectors of Θ.

We first rewrite Θ in matrix form as follows:

$$\Theta = \mathbf{Q}\begin{bmatrix}\mathbf{I}_m & \mathbf{0}_m \\ \mathbf{0}_m^T & -\frac{s}{2}\end{bmatrix}\mathbf{Q}^T \tag{29}$$

where $\mathbf{Q} \in \mathbb{R}^{n\times(m+1)}$ is defined as $\mathbf{Q} = [\mathbf{P} \quad \mathbf{1}_n]$.

Since $n \gg m$, we can perform the reduced singular value decomposition (SVD) on $\mathbf{Q}$ such that

$$\mathbf{Q} = \mathbf{U}_Q\Sigma_Q\mathbf{V}_Q^T \tag{30}$$

where $\mathbf{U}_Q \in \mathbb{R}^{n\times(m+1)}$ and $\mathbf{V}_Q \in \mathbb{R}^{(m+1)\times(m+1)}$ are two unitary matrices and $\Sigma_Q \in \mathbb{R}^{(m+1)\times(m+1)}$ is a diagonal matrix.

Then, Θ can be further rewritten as

$$\Theta = \mathbf{U}_Q\Phi\mathbf{U}_Q^T \tag{31}$$

where $\Phi \in \mathbb{R}^{(m+1)\times(m+1)}$ is defined as

$$\Phi = \Sigma_Q\mathbf{V}_Q^T\begin{bmatrix}\mathbf{I}_m & \mathbf{0}_m \\ \mathbf{0}_m^T & -\frac{s}{2}\end{bmatrix}\mathbf{V_Q}\Sigma_Q. \tag{32}$$

Suppose the eigendecomposition of Φ is $\Phi = \mathbf{V}_\Phi\Sigma_\Phi\mathbf{V}_\Phi^{-1}$, we have

$$\Theta = \mathbf{U}_Q\mathbf{V}_\Phi\Sigma_\Phi\mathbf{V}_\Phi^{-1}\mathbf{U}_Q^T \tag{33}$$

which can be rewritten as

$$\Theta\mathbf{U}_Q\mathbf{V}_\Phi = \mathbf{U}_Q\mathbf{V}_\Phi\Sigma_\Phi. \tag{34}$$

We know that $\mathbf{U}_Q\mathbf{V}_\Phi$ consists of $m+1$ eigenvectors of Θ and the diagonal elements in the diagonal matrix Σ_Φ are the corresponding eigenvalues. Therefore, the optimal solution $\mathbf{Y}^*$ are the c column vectors of $\mathbf{U}_Q\mathbf{V}_\Phi$ which correspond to the c biggest eigenvalues in Σ_Φ.

In the new method, we only perform the reduced SVD of $\mathbf{Q}$ and eigendecomposition of Φ. It can be verified that

$$\mathbf{Q}'\mathbf{Q} = \mathbf{V_Q}\Sigma_\mathbf{Q}^2\mathbf{V_Q}^T \tag{35}$$

which indicates that the column vectors of matrix $\mathbf{V_Q}$ are the eigenvectors of matrix $\mathbf{Q}'\mathbf{Q}$ and the diagonal elements in matrix $\Sigma_\mathbf{Q}^2$ are eigenvalues. Since $\mathbf{Q}'\mathbf{Q}$ is a $(m+1)\times(m+1)$ matrix, we only need $O(m^3)$ time to obtain $\mathbf{V_Q}$ and $\Sigma_\mathbf{Q}$. Then, $\mathbf{U_Q}$ can be obtained in $O(nm^2)$ time by

$$\mathbf{U}_Q = \mathbf{Q}\mathbf{V}_Q\Sigma_Q^{-1}. \tag{36}$$

Then it takes $O(m^3)$ time to perform eigendecomposition of Φ. Therefore, the overall time is $O(nm^2)$, which is a significant reduction from $O(n^3)$ considering $n \gg m$.

After we have obtained $\mathbf{Y}^*$, the next step is to learn a suitable orthogonal matrix $\mathbf{R}$ and a cluster indicator matrix $\mathbf{Y}$ such that $\mathbf{Y}^*\mathbf{R}$ is closest to $\mathbf{Y}$ by solving the following problem:

$$\min_{\mathbf{Y}\in\Phi^{n\times c},\ \mathbf{R}\in\mathbb{R}^{c\times c},\ \mathbf{Y1}_c=\mathbf{1}_n,\mathbf{R}^T\mathbf{R}=\mathbf{I}_c} \|\mathbf{Y} - \mathbf{Y}^*\mathbf{R}\|_F^2. \tag{37}$$

Note that $\|\mathbf{Y} - \mathbf{Y}^*\mathbf{R}\|_F^2 = 2n - 2\mathrm{Tr}(\mathbf{Y}^T\mathbf{Y}^*\mathbf{R})$, the above problem can be rewritten as

$$\max_{\mathbf{Y}\in\Phi^{n\times c},\ \mathbf{R}\in\mathbb{R}^{c\times c},\ \mathbf{Y1}_{c\times 1}=\mathbf{1}_n,\ \mathbf{R}^T\mathbf{R}=\mathbf{I}_c} \mathrm{Tr}(\mathbf{Y}^T\mathbf{Y}^*\mathbf{R}). \tag{38}$$

We can apply the alternative optimization approach to solve problem (38).

1) Update $\mathbf{R}$ *With* $\mathbf{Y}$ *Fixed:* If $\mathbf{Y}$ is fixed, supposing the SVD of $\mathbf{Y}^T\mathbf{Y}^*$ is $\mathbf{Y}^T\mathbf{Y}^* = \mathbf{U}\Sigma\mathbf{V}^T$, we have

$$\mathrm{Tr}(\mathbf{Y}^T\mathbf{Y}^*\mathbf{R}) = \mathrm{Tr}(\mathbf{R}\mathbf{U}\Sigma\mathbf{V}^T) = \mathrm{Tr}(\Sigma\mathbf{E}) = \sum_{i=1}^{c}\lambda_{ii}e_{ii} \tag{39}$$

where $\mathbf{E} = \mathbf{V}^T\mathbf{R}\mathbf{U}$, λ_{ii} and e_{ii} are the (i,i)th diagonal elements of matrix of Σ and $\mathbf{E}$, respectively.

It can be verified that $\mathbf{E}^T\mathbf{E} = \mathbf{U}^T\mathbf{R}\mathbf{V}\mathbf{V}^T\mathbf{R}^T\mathbf{U} = \mathbf{I}_c$, i.e., $\sum_{j=1}^{c} e_{ji}^2 = 1$, then we know that $e_{ii} \le 1 (1 \le i \le c)$. On the other hand, $\lambda_{ii} \ge 0$ since λ_{ii} is singular value. Therefore, $\mathrm{Tr}(\Sigma\mathbf{E}) = \sum_{i=1}^{c}\lambda_{ii}e_{ii} \le \sum_{i=1}^{c}\lambda_{ii}$, and the equality holds if and only if $e_{ii} = 1$ $(1 \le i \le c)$. This is to say, $\mathrm{Tr}(\Sigma\mathbf{E})$ reaches its maximum when $\mathbf{E} = \mathbf{I}_c$. Finally, we obtain the optimal solution of $\mathbf{R}$ as

$$\mathbf{R} = \mathbf{V}\mathbf{U}^T. \tag{40}$$

2) Update $\mathbf{Y}$ *with* $\mathbf{R}$ *fixed:* Let $\mathbf{G} = \mathbf{Y}^*\mathbf{R}$. According to problem (38), the optimal solution of $\mathbf{Y}$ can be obtained by solving the following problem:

$$\max_{\mathbf{Y}\in\Phi^{n\times c},\ \mathbf{Y1}_c=\mathbf{1}_n} \mathrm{Tr}(\mathbf{G}\mathbf{Y}) \tag{41}$$

which can be rewritten as

$$\max_{\mathbf{Y}\in\Phi^{n\times c},\ \mathbf{Y1}_c=\mathbf{1}_n} \sum_{j=1}^{c}\sum_{i=1}^{n} y_{ij}g_{ij}. \tag{42}$$

The optimal solution of $\mathbf{y}^i$ can be obtained as

$$y_{ij} = < l = \arg\max_{j'\in[1,c]} g_{ij'} > \tag{43}$$

where $< . >$ is 1 if the argument is true or 0 otherwise, and $\mathbf{G} = \mathbf{Y}^*\mathbf{R}$.

B. Solving s With $\mathbf{Y}$ *Fixed*

If $\mathbf{Y}$ fixed, problem (26) becomes

$$\min_{s>0}\left(s - \frac{\mathrm{Tr}(\mathbf{Y}^T\mathbf{A}\mathbf{Y})}{\|\mathbf{Y}\|_b}\right)^2. \tag{44}$$

The optimal solution of s is

$$s = \frac{\mathrm{Tr}(\mathbf{Y}^T\mathbf{A}\mathbf{Y})}{\|\mathbf{Y}\|_b}. \tag{45}$$

Authorized licensed use limited to: SHENZHEN UNIVERSITY. Downloaded on April 14,2023 at 16:50:09 UTC from IEEE Xplore. Restrictions apply.

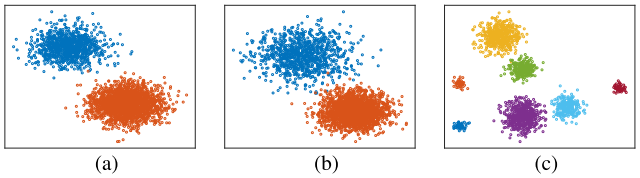

Fig. 1. Plots of three synthetic data sets. (a) Plot of D_{11}. (b) Plot of D_{21}. (c) Plot of D_{31}.

C. Optimization Algorithm

The detailed algorithm to solve problem (26), namely, the LABIN, is summarized in Algorithm 2. Given a data set, a sample-anchor similarity matrix is constructed first, and then s and $\mathbf{Y}$ are iteratively updated until convergence. In the new algorithm, we need $O(r_1(nm^2 + r_2nc^2))$ time to iteratively solve s and $\mathbf{Y}$ where r_1 is the number of iterations to update 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, LABIN has a time complexity of $O(nm^2)$ ($m \gg c$), which has a significant reduction in computation compared with the conventional spectral clustering methods which have a time complexity of $O(n^3)$.

Algorithm 2 Algorithm to Solve Problem (26): LABIN

Input: Data matrix $\mathbf{X} \in \mathbb{R}^{d\times n}$, number of nearest neighbors k, number of anchors m.
Find m anchors $\mathbf{W}$ with k-means, and construct a sparse representation matrix $\mathbf{B} \in \mathbb{R}^{n\times m}$ according to Eq. (16).
Compute $\mathbf{P} \in \mathbb{R}^{n\times m} = \mathbf{B}\Delta^{-\frac{1}{2}}$, where $\Delta \in \mathbb{R}^{m\times m}$ is the degree matrix of $\mathbf{B}$.
Perform the reduced SVD on $\mathbf{Q} = [\mathbf{P},\ \mathbf{1}_n]$ such that $\mathbf{Q} = \mathbf{U}_Q\Sigma_Q\mathbf{V}_Q^T$.
Randomly initialize $\mathbf{Y}$.
repeat
  Update $s = \frac{Tr(\mathbf{Y}^T\mathbf{P}\mathbf{P}^T\mathbf{Y})}{\|\mathbf{Y}\|_b}$.
  Update Φ according to Eq. (32) and perform eigendecomposition on Φ such that $\Phi = \mathbf{V}_\Phi\Sigma_\Phi\mathbf{V}_\Phi^{-1}$.
  Update $\mathbf{Y}^*$ as the c column vectors of $\mathbf{U}_Q\mathbf{V}_\Phi$ which correspond to the c biggest eigenvalues in Σ_Φ.
  repeat
    Update $\mathbf{R}$ according to Eq. (40).
    Update $\mathbf{Y}$ according to Eq. (43).
  until problem (38) converges
until problem (26) converges
Output: The clustering result $\mathbf{Y}$.

IV. Experimental Results and Analysis on Synthetic Data Sets

In this section, we present the experimental results on synthetic data sets to demonstrate the efficiency and effectiveness of the proposed methods.

A. Data Sets and Evaluation Metric

We generated three 2-D synthetic data sets (shown in Fig. 1), i.e., D_{11}, D_{12}, and D_{13}. D_{11} consists of two Gaussian-shaped clusters that contain the same numbers of objects, whereas D_{12} consists of two Gaussian-shaped clusters in which the larger cluster consists of two times of objects than the smaller cluster. D_{31} consists of seven clusters with different sizes, in which the largest cluster is nine times larger than the smallest cluster. We increased proportionately the number of objects in the clusters of the 3 data sets and generated 11 synthetic data sets for this experiment. The characteristics of these data sets are given in Table I.

TABLE I
Characteristic of 11 Synthetic Data Sets

Datasets	No. of objects	No. of clusters
D_{11}	3000	
D_{12}	4000	2
D_{13}	6000	
D_{21}	3000	
D_{22}	4000	2
D_{23}	6000	
D_{31}	1650	
D_{32}	3300	
D_{33}	6600	7
D_{34}	13200	
D_{35}	19800	

The clustering result is evaluated by comparing the obtained label of each sample with the label provided by the data set. We use both the accuracy (ACC) and normalized mutual information (NMI) metric to measure the clustering performance.

B. Results and Analysis

With all 11 data sets given in Table I, we compared LABIN with four approximate spectral clustering methods, i.e., KASP [24], Nyström approximation-based spectral clustering (Nyström) [23], LSC [26], and SNC [27]. We also compared LABIN with NCut with k-means [9], RCut with k-means [8], in order to check whether LABIN can produce comparable result with nonapproximate methods. In this experiment, the number of clusters was set to 500 for KASP and random selection was used for Nyström. We used the similarity construction method in [13] to construct an affinity matrix for each data set to run NCut, Rcut, and SBMC, where the nearest neighbors were set to 50. We also used the method in [27] to construct sample-anchor similarity matrix for LSC, SNC, and LABIN, where the number of nearest neighbors was set to 50, and the number of anchors was set to eight integers from 150 to 500 for PA25, CA16, and CA28 and nine integers from 100 to 500. The number of clusters for KASP and the number of samples for Nyström were also set to eight integers from 150 to 500 for PA25, CA16, and CA28 and nine integers from 100 to 500.

The average clustering performance of eight clustering methods are given in Tables II and III. From these two tables, we can see that LABIN outperformed other four approximate spectral clustering methods in accuracy and NMI on 9 of 11 data sets. To be specific, on D_{34}, LABIN achieves a greater than 18% improvement compared to the second-best method SNC in terms of ACC, and a greater than 14% improvement compared to the second-best method LSC in terms of NMI. Especially on D_{35}, LABIN achieves a greater than 18% improvement compared to the second-best method SNC in

Authorized licensed use limited to: SHENZHEN UNIVERSITY. Downloaded on April 14,2023 at 16:50:09 UTC from IEEE Xplore. Restrictions apply.

TABLE II

ACC ± STANDARD DEVIATION (%) OF 7 CLUSTERING METHODS ON 11 SYNTHETIC DATA SETS. THE BEST RESULT ON EACH DATA SET (EXCLUDING THOSE OF NCUT, RCUT, AND SBMC) IS HIGHLIGHTED IN BOLD

Data	NCut	RCut	SBMC	Nyström	KASP	LSC	SNC	LABIN
D_{11}	99.63 ± 0.23	99.63 ± 0.23	99.71 ± 0.04	74.73 ± 12.89	99.7 ± 0.04	99.71 ± 0.04	87.49 ± 15.64	**99.74 ± 0.02**
D_{12}	99.9 ± 0.01	99.9 ± 0.01	99.66 ± 0.01	72.33 ± 17.65	99.52 ± 0.23	99.79 ± 0.11	81.2 ± 1.9	**99.86 ± 0.04**
D_{13}	99.45 ± 0.02	99.45 ± 0.02	99.17 ± 0.02	75.8 ± 16.5	97.97 ± 0.42	**99.44 ± 0.29**	88.4 ± 15.54	99.43 ± 0.2
D_{21}	99.97 ± 0.01	99.97 ± 0.01	99.97 ± 0	78.23 ± 15.76	99.96 ± 0.02	99.96 ± 0.02	88.78 ± 15.82	**99.97 ± 0.01**
D_{22}	99.93 ± 0.01	99.93 ± 0.01	99.93 ± 0	79.99 ± 19.93	99.92 ± 0.01	99.92 ± 0.02	79.94 ± 4.93	**99.93 ± 0.01**
D_{23}	99.93 ± 0.01	99.93 ± 0.01	99.93 ± 0.01	85.55 ± 16.35	99.89 ± 0.07	**99.94 ± 0.01**	96.17 ± 10.55	99.94 ± 0.01
D_{31}	71.60 ± 16.07	99.89 ± 0.05	85.12 ± 10.23	51.31 ± 12.64	75.1 ± 6.69	67.36 ± 6.3	73.56 ± 9.49	**85.99 ± 10.8**
D_{32}	67.48 ± 8.28	99.67 ± 0.04	85.19 ± 14.88	67.81 ± 12.00	70.29 ± 8.14	69.94 ± 6.56	76.15 ± 10.97	**86.62 ± 9.15**
D_{33}	92.49 ± 9.93	99.69 ± 0.09	80.98 ± 6.09	59.09 ± 12.32	67.61 ± 3.69	67.37 ± 5.42	70.24 ± 10.92	**85.6 ± 7.94**
D_{34}	81.70 ± 0.14	99.38 ± 0.14	83.53 ± 12.65	68.8 ± 11.40	67.87 ± 4.64	70.32 ± 7.15	73.74 ± 13.14	**87.23 ± 11.21**
D_{35}	92.15 ± 9.92	99.37 ± 0.05	77.31 ± 11.67	68.9 ± 11.23	70.95 ± 8.8	67.98 ± 5.54	76.26 ± 10.76	**90.28 ± 8.02**

TABLE III

NMI ± STANDARD DEVIATION (%) OF 7 CLUSTERING METHODS ON 11 SYNTHETIC DATA SETS. THE BEST RESULT ON EACH DATA SET (EXCLUDING THOSE OF NCUT, RCUT, AND SBMC) IS HIGHLIGHTED IN BOLD

Data	NCut	RCut	SBMC	Nyström	KASP	LSC	SNC	LABIN
D_{11}	96.45 ± 1.73	96.45 ± 1.73	96.89 ± 0.4	29.38 ± 21.8	96.86 ± 0.34	96.9 ± 0.33	58.96 ± 44.29	**97.18 ± 0.16**
D_{12}	98.81 ± 0.11	98.81 ± 0.11	96.9 ± 0.08	28.47 ± 20.16	96.05 ± 1.47	97.95 ± 0.84	41.89 ± 3.68	**98.54 ± 0.31**
D_{13}	94.89 ± 0.22	94.89 ± 0.22	92.96 ± 0.15	37.06 ± 31.70	86.08 ± 2.18	**94.93 ± 1.85**	62.41 ± 44.68	94.79 ± 1.33
D_{21}	99.55 ± 0.01	99.55 ± 0.01	99.55 ± 0.01	40.05 ± 30.37	99.47 ± 0.24	99.52 ± 0.17	65.45 ± 46.81	**99.55 ± 0.01**
D_{22}	99.12 ± 0.01	99.12 ± 0.01	99.12 ± 0.01	54.29 ± 36.70	99.06 ± 0.11	99.07 ± 0.15	40.00 ± 8.14	**99.12 ± 0.01**
D_{23}	99.17 ± 0.01	99.17 ± 0.01	99.14 ± 0.02	57.06 ± 36.15	98.76 ± 0.69	**99.20 ± 0.11**	87.46 ± 31.27	99.20 ± 0.13
D_{31}	80.68 ± 10.68	99.54 ± 0.20	89.03 ± 5.31	42.62 ± 16.03	80.74 ± 3.78	76.91 ± 4.71	71.53 ± 10.28	**89.67 ± 6.79**
D_{32}	76.65 ± 6.73	98.74 ± 0.11	89.23 ± 8.93	60.82 ± 10.91	76.89 ± 4.48	77.29 ± 4.40	73.63 ± 10.82	**88.55 ± 6.01**
D_{33}	94.13 ± 6.5	98.71 ± 0.30	84.10 ± 6.21	55.77 ± 12.84	75.47 ± 2.13	75.43 ± 3.39	70.21 ± 10.74	**86.84 ± 6.23**
D_{34}	86.67 ± 0.42	97.81 ± 0.40	87.13 ± 6.72	65.88 ± 11.56	75.52 ± 2.24	77.35 ± 4.82	71.42 ± 12.25	**88.56 ± 7.31**
D_{35}	93.13 ± 6.42	97.73 ± 0.14	83.51 ± 6.33	68.52 ± 11.75	77.22 ± 3.31	75.65 ± 3.79	74.23 ± 10.99	**90.39 ± 5.27**

terms of ACC and a greater than 17% improvement compared to the second-best method KASP in terms of NMI. LABIN even outperformed two nonapproximate spectral clustering methods on D_{11} and D_{23}. We also noted that SBMC produced bad results on five large data sets $\{D_{31}, \ldots, D_{35}\}$ that indicates that SBMC is sensitive to the initial partition and unstable on the large data sets. LABIN outperformed SBMC on all five large data sets $\{D_{31}, \ldots, D_{35}\}$.

The time costs of eight clustering methods are shown in Fig. 2. This figure shows that LABIN spent comparable time with KASP, LSC, and SNC that is much less than the time costs of NCut and RCut. It increases linearly with the increase in data size. This experiment result indicates that LABIN is scalable to large scale data.

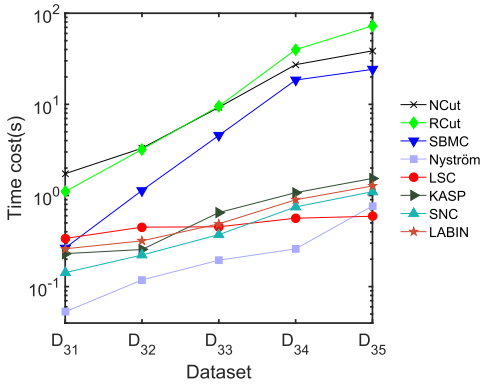

Fig. 2. Time costs of eight clustering methods on five synthetic data sets, when $k = 10$ and $m = 300$.

V. EXPERIMENTAL RESULTS AND ANALYSIS ON THE REAL-WORLD DATA SETS

In this section, we present the experimental results on real-world data sets to demonstrate the efficiency and effectiveness of the proposed method.

A. Benchmark Data Sets

Eight large scale benchmark data sets were selected from Xiaojun Chen's page,[1] for this experiment. Table IV summarizes the characteristics of these eight data sets, and Fig. 3 shows some example images in three data sets.

[1] http://www.escience.cn/people/chenxiaojun/data.html/#

B. Results and Analysis

With all eight data sets given in Table IV, we have implemented LABIN[1] and compared it with four approximate spectral clustering methods KASP [24], Nyström [23], LSC [26], SNC [27], and three nonapproximate methods NCut [9], RCut [8], and SBMC [14]. This experiment was conducted in the same way as the experiment in Section IV-B.

Authorized licensed use limited to: SHENZHEN UNIVERSITY. Downloaded on April 14,2023 at 16:50:09 UTC from IEEE Xplore. Restrictions apply.

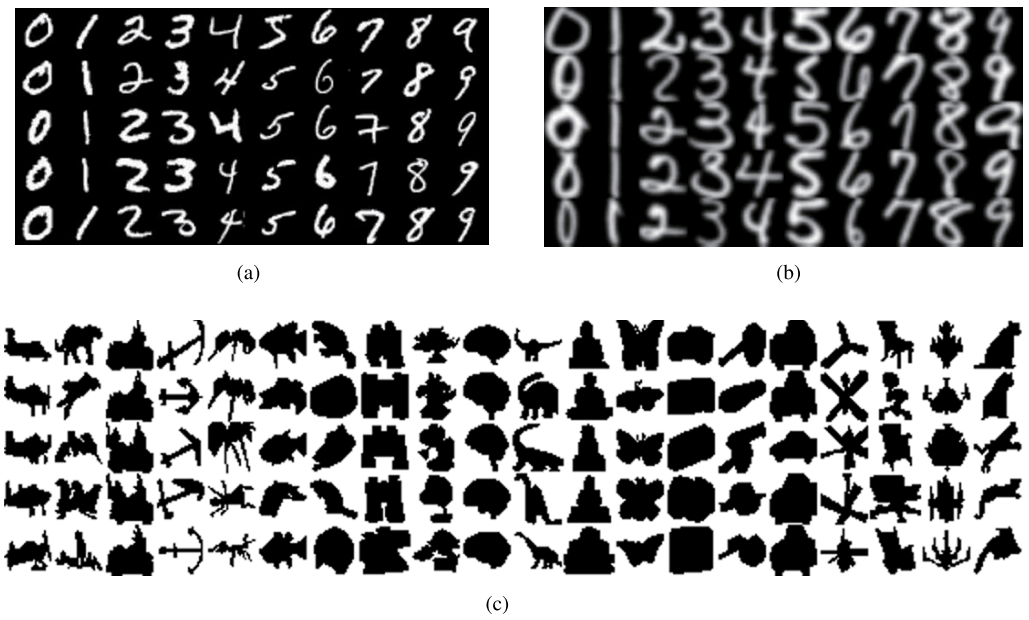

Fig. 3. Some example images. (a) MNIST data set images from 10 digit classes: 0 to 9. (b) USPS data set images from 10 digit classes: 0 to 9. (c) CA28 data set images from 20 classes.

TABLE IV
CHARACTERISTICS OF EIGHT REAL-WORLD DATA SETS

Data sets	Abbr.	#Samples	#Features	#Classes
PalmData25_uni	$PA25$	2,000	256	100
isolet5	$ISO5$	7,797	617	26
caltech101-silhouettes-16	$CA16$	8,641	256	101
caltech101-silhouettes-28	$CA28$	8,641	784	101
USPSdata_uni	$USPS$	9,298	256	10
MnistData_70k	$MNIST$	70,000	784	10
seismic	$SEIS$	98,528	50	3
covtype	$COVT$	58,1012	54	7

The ACC and NMI results of nine clustering methods on eight real-world data sets are given in Tables V and VI. From these results, we can see that LABIN outperformed all the other methods on five of total eight data sets. To be specific, LABIN achieves a nearly 11% average improvement in terms of NMI on the CA28 data set, compared to the second-best method KASP. LABIN even achieves a nearly 15% average improvement in terms of ACC on the PA25 data set, compared to the second-best method KASP. We also noted that, especially on the three data sets consisting of greater than or equal to 100 clusters, LABIN outperformed all four approximate spectral clustering methods, i.e., KASP, Nyström, LSC, and SNC. This result verifies that LABIN is able to uncover high-quality cluster structure from a data set with a large number of clusters since it can automatically balance graph cut and cluster size in order to produce high-quality clusters. Compared with the two nonapproximate spectral clustering methods NCut and RCut, LABIN produced comparable results with them and even outperformed them on the CA16 and CA28 data sets. LABIN even outperformed SBMC on the PA25, ISO5, USPS, and MNIST data sets. These results show the superior performance of LABIN.

C. Parameter Selection

In this section, we investigate how the clustering results of the approximate spectral clustering methods vary with the change of k and m. Fig. 4 shows the result of KASP, Nyström, LSC, SNC, and LABIN versus the number of selected anchors m, and Fig. 5 shows the result of NCut, RCut, SBMC, LSC, SNC, and LABIN versus the number of nearest neighbors k. Fig. 4 shows that the results of three anchor strategy-based methods LSC, SNC, and LABIN increased with the increase in m, while the results of the other two methods are unstable with the change in m. Fig. 5 shows that smaller k results in better results. Therefore, we can select big m and small k to produce good clustering results. In real applications, we can perform hierarchy grid search to select the proper m and k for better results.

D. Scalability Analysis

Fig. 6 shows the time costs of KASP, Nyström, LSC, SNC, and LABIN versus the number of selected anchors m, and Fig. 7 shows the time costs of NCut, RCut, SBMC, LSC, SNC, and LABIN versus the number of selected anchors k. These results show that the time costs of these methods increased with the increase in the number of anchors m and were nearly stable as the number of nearest neighbors k increased. The time costs of three fast spectral clustering methods LSC, SNC, and LABIN are much less than the time costs of three nonapproximate spectral clustering methods NCut, RCut, and SBMC. We also noted that the time costs of LABIN are comparable to SNC and less than the time costs of LSC. This indicates that LABIN scales well to large scale data sets.

Authorized licensed use limited to: SHENZHEN UNIVERSITY. Downloaded on April 14,2023 at 16:50:09 UTC from IEEE Xplore. Restrictions apply.

TABLE V

ACC ± STANDARD DEVIATION (%) OF 8 CLUSTERING METHODS ON 8 REAL-WORLD DATA SETS. THE BEST RESULT ON EACH DATA SET IS HIGHLIGHTED IN BOLD

Data	NCut	RCut	SBMC	Nyström	KASP	LSC	SNC	LABIN
$PA25$	76.06 ± 2.24	75.94 ± 0.88	68.44 ± 1.53	50.31 ± 16.25	**64.36 ± 6.56**	57.1 ± 6.28	50.88 ± 5.98	**73.91 ± 2.07**
$ISO5$	55.99 ± 1.23	56.39 ± 1.14	49.46 ± 5.57	**55.18 ± 2.53**	54.36 ± 5.11	52.37 ± 6.93	43.7 ± 5.78	**55.8 ± 3.38**
$CA16$	25.36 ± 0.55	26.45 ± 0.35	31.55 ± 1.31	21.06 ± 2.62	**26.19 ± 2.02**	22.05 ± 2.15	26.07 ± 1.71	**30.34 ± 1.30**
$CA28$	26.58 ± 0.54	26.85 ± 0.36	30.30 ± 0.75	22.92 ± 1.50	**25.82 ± 1.67**	22.54 ± 2.2	25.22 ± 2.16	**30.38 ± 1.34**
$USPS$	66.33 ± 0.41	66.41 ± 0.32	66.77 ± 2.26	46.23 ± 11.81	**69.18 ± 2.69**	64.03 ± 6.11	55.82 ± 5.19	**67.72 ± 2.94**
$MNIST$	68.4 ± 1.37	69.03 ± 1.43	54.31 ± 7.77	52.39 ± 1.96	56.83 ± 2.99	**57.34 ± 4.94**	46.38 ± 6.50	**59.00 ± 5.09**
$SEIS$	57.74 ± 0.33	57.73 ± 0.32	43.30 ± 4.33	42.05 ± 3.70	**49.81 ± 1.82**	42.29 ± 2.08	**50.02 ± 6.42**	43.84 ± 4.10
$COVT$	30.13 ± 3.06	34.10 ± 9.82	20.10 ± 6.82	**27.62 ± 3.45**	23.12 ± 1.02	21.82 ± 0.69	**24.25 ± 2.27**	23.37 ± 1.20

TABLE VI

NMI ± STANDARD DEVIATION (%) OF 8 CLUSTERING METHODS ON 8 REAL-WORLD DATA SETS. THE BEST RESULT ON EACH DATA SET IS HIGHLIGHTED IN BOLD

Data	NCut	RCut	SBMC	Nyström	KASP	LSC	SNC	LABIN
$PA25$	91.75 ± 0.6	91.65 ± 0.49	87.74 ± 0.62	72.75 ± 12.26	**85.89 ± 2.82**	80.08 ± 4.11	76.08 ± 4.52	**90.47 ± 0.78**
$ISO5$	75.66 ± 0.57	75.75 ± 0.69	70.83 ± 2.58	**70.78 ± 1.37**	69.65 ± 2.90	66.16 ± 6.29	57.7 ± 6.52	**71.38 ± 2.22**
$CA16$	50.52 ± 0.35	51.2 ± 0.45	53.93 ± 0.18	44.36 ± 3.45	**48.62 ± 1.03**	47.19 ± 2.01	44.77 ± 1.76	**53.31 ± 0.73**
$CA28$	51.44 ± 0.38	51.55 ± 0.40	53.62 ± 0.29	47.21 ± 1.93	**48.4 ± 0.98**	47.30 ± 2.11	44.98 ± 1.91	**53.45 ± 0.80**
$USPS$	80.65 ± 1.35	80.55 ± 1.38	62.82 ± 0.59	38.19 ± 14.26	**67.16 ± 2.43**	61.08 ± 6.63	53.56 ± 6.94	**68.51 ± 4.44**
$MNIST$	73.78 ± 0.79	74.22 ± 0.59	50.35 ± 3.62	43.49 ± 3.32	**53.94 ± 2.71**	52.24 ± 7.31	41.9 ± 7.62	**58.98 ± 5.33**
$SEIS$	9.68 ± 0.82	9.72 ± 0.82	6.23 ± 3.51	2.73 ± 3.15	**13.82 ± 0.76**	4.57 ± 1.58	**7.71 ± 4.36**	7.41 ± 5.32
$COVT$	8.83 ± 2.60	5.73 ± 3.76	5.02 ± 2.71	**7.45 ± 1.44**	6.02 ± 0.20	6.17 ± 0.36	4.53 ± 1.01	**6.38 ± 0.18**

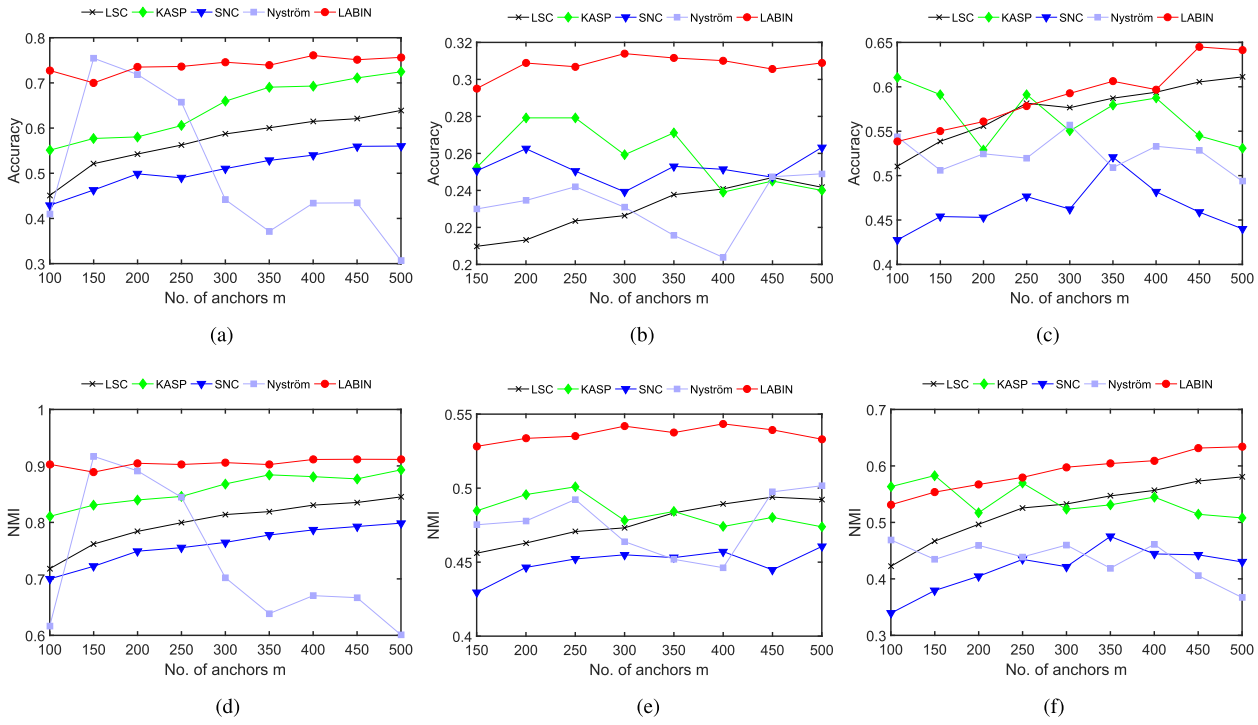

Fig. 4. Average ACC and NMI of LSC, SNC, and LABIN versus the number of anchors m, average ACC, and NMI of KASP versus the number of clusters m and average ACC and NMI of Nyström versus the number of samples m on three data sets. (a) ACC versus m on the PA25 data set. (b) ACC versus m on the CA28 data set. (c) ACC versus m on the MNIST data set. (d) NMI versus m on the PA25 data set. (e) NMI versus m on the CA28 data set. (f) NMI versus m on the MNIST data set.

E. Balance Parameter

In this experiment, we check whether the learned balance parameter s improves the performance of LABIN. We ran LABIN with fixed s (LABIN-s) which varies from 10^{-10} to 1 and compared their clustering results with those of LABIN. The ACC and NMI results are shown in Figs. 8 and 9. These results show that LABIN with the learned s outperformed almost all results with fixed s. Although we may obtain better result by performing hierarchy grid

Authorized licensed use limited to: SHENZHEN UNIVERSITY. Downloaded on April 14,2023 at 16:50:09 UTC from IEEE Xplore. Restrictions apply.

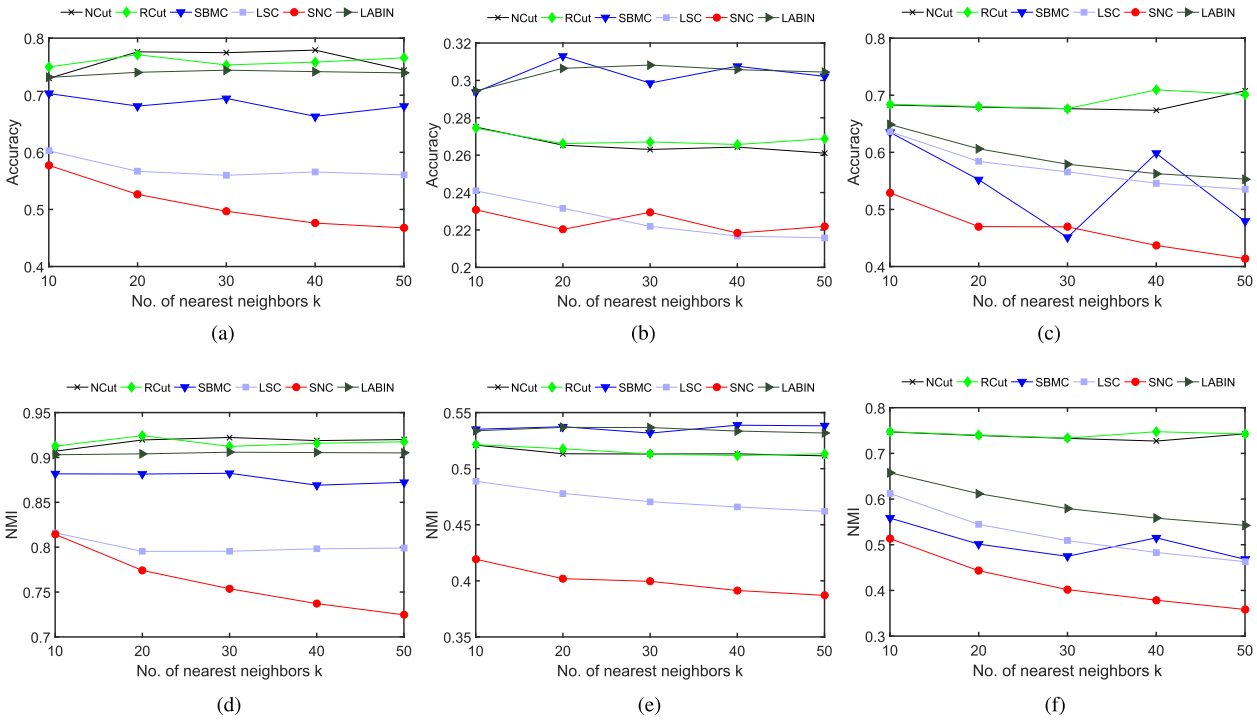

Fig. 5. Average ACC and NMI of NCut, RCut, SBMC, LSC, SNC, and LABIN versus the number of nearest neighbors k. (a) ACC versus k on the PA25 data set. (b) ACC versus k on the CA28 data set. (c) ACC versus k on the MNIST data set. (d) NMI versus k on the PA25 data set. (e) NMI versus k on the CA28 data set. (f) NMI versus k on the MNIST data set.

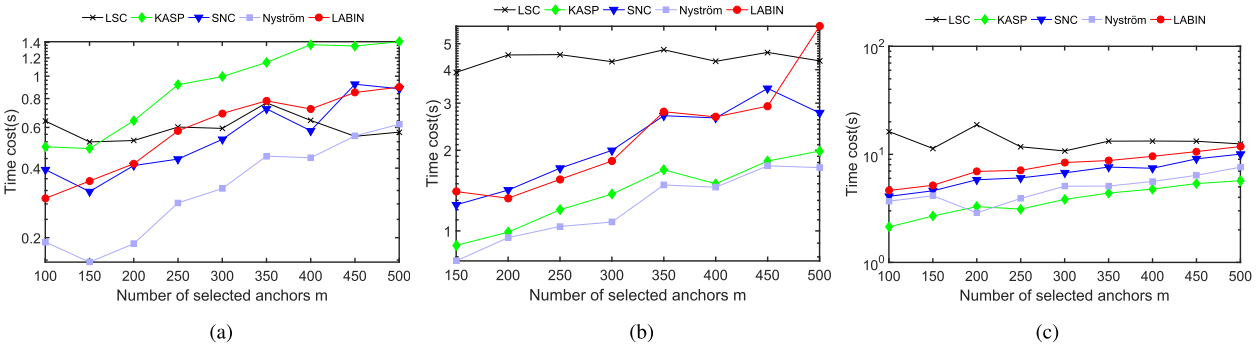

Fig. 6. Average time cost of LSC, SNC, and LABIN versus the number of anchors m, average time cost of KASP versus the number of clusters m, and average time cost of Nyström versus the number of samples m on three data sets. (a) PA25. (b) CA28. (c) MNIST.

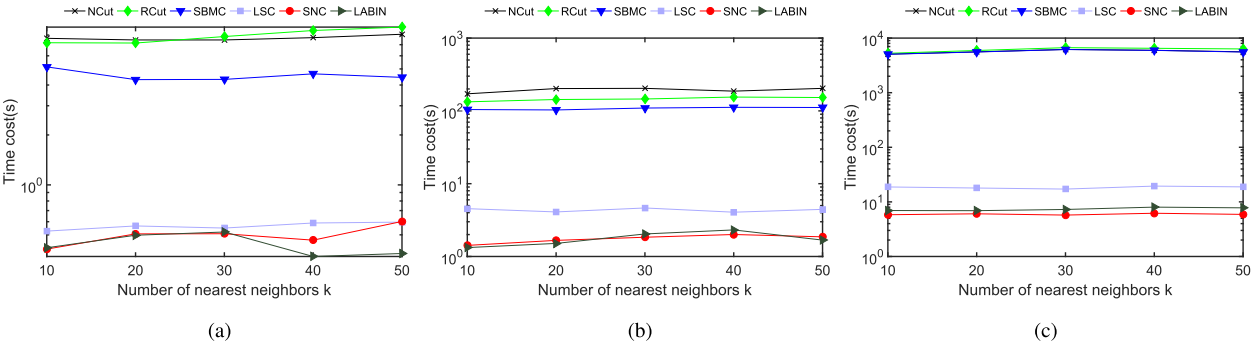

Fig. 7. Average time cost of NCut, RCut, SBMC, LSC, SNC, and LABIN versus the number of nearest neighbors k on three data sets. (a) PA25. (b) CA28. (c) MNIST.

search to select proper s, it is time-consuming and inefficient for large-scale data. However, LABIN can obtain the clustering result that is very close to the best result with fixed s by running once. Therefore, learning the balance parameter is both effective and efficient for large-scale data.

Authorized licensed use limited to: SHENZHEN UNIVERSITY. Downloaded on April 14,2023 at 16:50:09 UTC from IEEE Xplore. Restrictions apply.

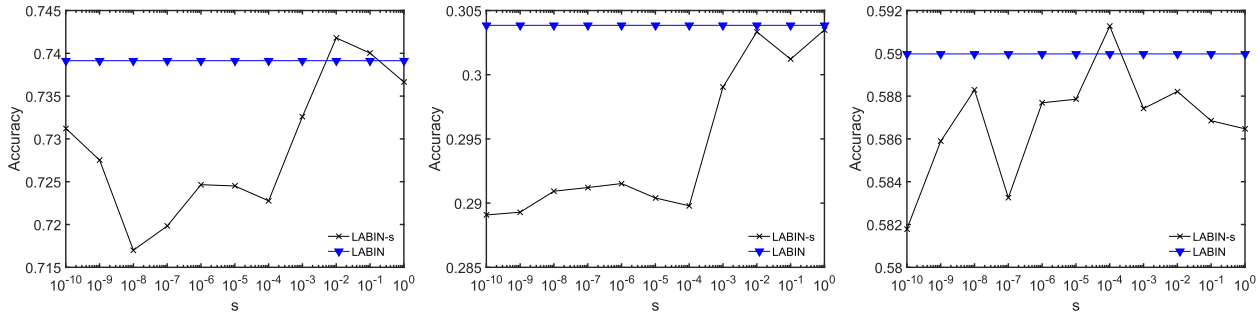

Fig. 8. Accuracy results versus s on three data sets. (a) PA25. (b) CA28. (c) MNIST.

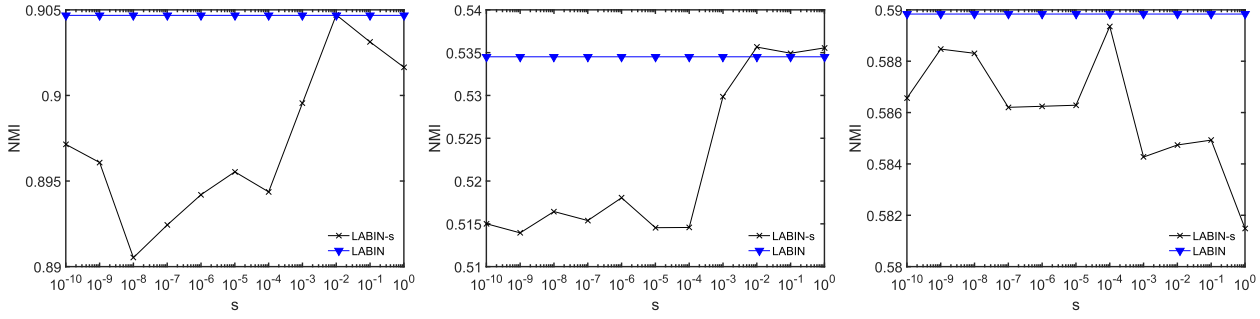

Fig. 9. NMI results versus s on three data sets. (a) PA25. (b) CA28. (c) MNIST.

VI. Conclusion

In this paper, we have proposed a new spectral clustering method for large-scale data, namely, LABIN. In the new method, the anchor-based strategy is used to construct a sample-anchor similarity matrix and the clustering is performed on the sample-anchor similarity matrix instead of the full similarity matrix, in which the ISR is used to optimize the new model and an efficient method is proposed to perform eigendecomposition with both linear time and space computational complexities. Compared with the original SBMC, the new method has significant reduction on space computational complexity and is more robust due to the use of ISR. Extensive experimental results show the effectiveness and efficiency of the new method compared to the state-of-the-art methods.

However, LABIN still uses a two-stage optimization method to obtain the final clustering result. In the future work, we will study new optimization method to directly solve LABIN. Extending LABIN to semi-supervised learning task is also our future work.

References

[1] U. von Luxburg, "A tutorial on spectral clustering," *Statist. Comput.*, vol. 17, no. 4, pp. 395–416, 2007.

[2] H.-P. Kriegel, P. Kröger, and A. Zimek, "Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering," *ACM Trans. Knowl. Discovery Data*, vol. 3, no. 1, pp. 1–58, 2009.

[3] X. Chen, Y. Ye, X. Xu, and J. Z. Huang, "A feature group weighting method for subspace clustering of high-dimensional data," *Pattern Recognit.*, vol. 45, no. 1, pp. 434–446, 2012.

[4] X. Cai, F. Nie, H. Huang, and F. Kamangar, "Heterogeneous image feature integration via multi-modal spectral clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2011, pp. 1977–1984.

[5] X. Chen, X. Xu, J. Z. Huang, and Y. Ye, "TW-k-means: Automated two-level variable weighting clustering algorithm for multiview data," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 4, pp. 932–944, Apr. 2013.

[6] X. Chen, J. Z. Huang, Q. Wu, and M. Yang, "Subspace weighting co-clustering of gene expression data," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 16, no. 2, pp. 352–364, Mar./Apr. 2017.

[7] X. Chen, M. Yang, J. Z. Huang, and Z. Ming, "TWCC: Automated two-way subspace weighting partitional co-clustering," *Pattern Recognit.*, vol. 76, pp. 404–415, Apr. 2018.

[8] L. Hagen and A. B. Kahng, "New spectral methods for ratio cut partitioning and clustering," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 11, no. 9, pp. 1074–1085, Sep. 1992.

[9] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 2. 2001, pp. 849–856.

[10] S. X. Yu and J. Shi, "Multiclass spectral clustering," in *Proc. 9th IEEE Int. Conf. Comput. Vis.*, vol. 1. Oct. 2003, pp. 313–319.

[11] F. Nie, Z. Zeng, I. W. Tsang, D. Xu, and C. Zhang, "Spectral embedded clustering: A framework for in-sample and out-of-sample spectral clustering," *IEEE Trans. Neural Netw.*, vol. 22, no. 11, pp. 1796–1808, Nov. 2011.

[12] F. Nie, X. Wang, and H. Huang, "Clustering and projected clustering with adaptive neighbors," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2014, pp. 977–986.

[13] F. Nie, X. Wang, M. Jordan, and H. Huang, "The constrained Laplacian rank algorithm for graph-based clustering," in *Proc. 13th AAAI Conf. Artif. Intell.*, 2016, pp. 1969–1976.

[14] X. Chen, J. Z. Huang, F. Nie, R. Chen, and Q. Wu, "A self-balanced min-cut algorithm for image clustering," in *Proc. Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2080–2088.

[15] C. Hou, F. Nie, D. Yi, and D. Tao, "Discriminative embedded clustering: A framework for grouping high-dimensional data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 6, pp. 1287–1299, Jun. 2015.

[16] Q. Wang, Z. Qin, F. Nie, and X. Li, "Spectral embedded adaptive neighbors clustering," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 4, pp. 1265–1271, Apr. 2018.

Authorized licensed use limited to: SHENZHEN UNIVERSITY. Downloaded on April 14,2023 at 16:50:09 UTC from IEEE Xplore. Restrictions apply.

[17] Z. Li, F. Nie, X. Chang, L. Nie, H. Zhang, and Y. Yi, "Rank-constrained spectral clustering with flexible embedding," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 12, pp. 6073–6082, Dec. 2018.
[18] Z. Kang, L. Wen, W. Chen, and Z. Xu, "Low-rank kernel learning for graph-based clustering," *Knowledge-Based Syst.*, vol. 163, pp. 510–517, Jan. 2019.
[19] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.
[20] M. C. P. de Souto, I. G. Costa, D. S. A. de Araujo, T. B. Ludermir, and A. Schliep, "Clustering cancer gene expression data: A comparative study," *BMC Bioinform.*, vol. 9, no. 1, p. 497, 2008.
[21] C. Fowlkes, S. Belongie, F. Chung, and J. Malik, "Spectral grouping using the Nyström method," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 2, pp. 214–225, Feb. 2004.
[22] M. Li, J. T. Kwok, and B.-L. Lu, "Making large-scale Nyström approximation possible," in *Proc. Int. Conf. Mach. Learn.*, 2010, pp. 631–638.
[23] W.-Y. Chen, Y. Song, H. Bai, C.-J. Lin, and E. Y. Chang, "Parallel spectral clustering in distributed systems," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 3, pp. 568–586, Mar. 2011.
[24] D. Yan, L. Huang, and M. I. Jordan, "Fast approximate spectral clustering," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2009, pp. 907–916.
[25] H. Shinnou and M. Sasaki, "Spectral clustering for a large data set by reducing the similarity matrix size," in *Proc. Int. Conf. Lang. Resour. Eval. (LREC)*, Marrakech, Morocco, May/Jun. 2008, pp. 201–204.
[26] D. Cai and X. Chen, "Large scale spectral clustering via landmark-based sparse representation," *IEEE Trans. Cybern.*, vol. 45, no. 8, pp. 1669–1680, Aug. 2015.
[27] X. Chen, F. Nie, J. Z. Huang, and M. Yang, "Scalable normalized cut with improved spectral rotation," in *Proc. 26th Int. Joint Conf. Artif. Intell. (IJCAI)*, 2017, pp. 1518–1524.
[28] W. Liu, J. He, and S.-F. Chang, "Large graph construction for scalable semi-supervised learning," in *Proc. Int. Conf. Mach. Learn.*, 2010, pp. 679–686.

Xiaojun Chen (M'16) received the Ph.D. degree from the Harbin Institute of Technology, Harbin, China, in 2011.

He is currently an Assistant Professor with the College of Computer Science and Software, Shenzhen University, Shenzhen, China. His current research interests include clustering, feature selection, and massive data mining.

Renjie Chen is currently pursuing the master's (M.A.) degree with the School of Software Engineering, South China University of Technology, Guangzhou, China.

His current research interests include clustering and feature selection.

Qingyao Wu received the Ph.D. degree in computer science from the Harbin Institute of Technology, Harbin, China, in 2013.

He is currently a Professor with the School of Software Engineering, South China University of Technology, Guangzhou, China. His current research interests include computer vision, natural language processing, and big data learning.

Yixiang Fang received the B.Eng. degree from Harbin Engineering University, Harbin, China, in 2010, and the M.S. degree from the Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen, China, in 2013. He is currently pursuing the Ph.D. degree with the Department of Computer Science, University of Hong Kong (HKU), Hong Kong, under the supervision of Dr. R. Cheng.

His current research interests include Big Data analytics on spatial–temporal databases, graph databases, uncertain databases, and web data mining.

Feiping Nie received the Ph.D. degree in computer science from Tsinghua University, Beijing, China, in 2009.

He has authored or coauthored more than 100 papers in the following top journals and conferences: the IEEETRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, *International Journal of Computer Vision*, the IEEE TRANSACTIONS ON IMAGE PROCESSING, the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS/the IEEE TRANSACTIONS ON NEURAL NETWORKS, the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, it Transactions on Knowledge Discovery from Data, it Bioinformatics, ICML, NIPS, KDD, IJCAI, AAAI, ICCV, CVPR, and ACM MM. His papers have been cited more than 5000 times (Google scholar). His current research interests include machine learning and its applications, such as pattern recognition, data mining, computer vision, image processing, and information retrieval.

Dr. Nie currently serves as a PC Member or an Associate Editor for several prestigious journals and conferences in the related fields.

Joshua Zhexue Huang received the Ph.D. degree from the Royal Institute of Technology, Stockholm, Sweden, in 1993.

He is currently a Professor with the College of Computer Science and Software, Shenzhen University, Shenzhen, China, a Professor and Chief Scientist with the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, and an Honorary Professor with the Department of Mathematics, The University of Hong Kong, Hong Kong. His current research interests include data mining, machine learning, and clustering algorithms.

Authorized licensed use limited to: SHENZHEN UNIVERSITY. Downloaded on April 14,2023 at 16:50:09 UTC from IEEE Xplore. Restrictions apply.