Robust Spectral Ensemble Clustering via Rank Minimization

ZHIQIANG TAO, Northeastern University
HONGFU LIU, Brandeis University
SHENG LI, University of Georgia
ZHENGMING DING, Indiana University - Purdue University Indianapolis
YUN FU, Northeastern University

Ensemble Clustering (EC) is an important topic for data cluster analysis. It targets to integrate multiple Basic Partitions (BPs) of a particular dataset into a consensus partition. Among previous works, one promising and effective way is to transform EC as a graph partitioning problem on the co-association matrix, which is a pair-wise similarity matrix summarized by all the BPs in essence. However, most existing EC methods directly utilize the co-association matrix, yet without considering various noises (e.g., the disagreement between different BPs and the outliers) that may exist in it. These noises can impair the cluster structure of a co-association matrix, and thus mislead the final graph partitioning process. To address this challenge, we propose a novel Robust Spectral Ensemble Clustering (RSEC) algorithm in this article. Specifically, we learn low-rank representation (LRR) for the co-association matrix to uncover its cluster structure and handle the noises, and meanwhile, we perform spectral clustering with the learned representation to seek for a consensus partition. These two steps are jointly proceeded within a unified optimization framework. In particular, during the optimizing process, we leverage consensus partition to iteratively enhance the block-diagonal structure of LRR, in order to assist the graph partitioning. To solve RSEC, we first formulate it by using nuclear norm as a convex proxy to the rank function. Then, motivated by the recent advances in non-convex rank minimization, we further develop a non-convex model for RSEC and provide it a solution by the majorization–minimization Augmented Lagrange Multiplier algorithm. Experiments on 18 real-world datasets demonstrate the effectiveness of our algorithm compared with state-of-the-art methods. Moreover, several impact factors on the clustering performance of our approach are also explored extensively.

CCS Concepts: • **Computing methodologies → Machine learning**; • **Theory of computation → Unsupervised learning and clustering**; • **Information systems → Clustering**;

Additional Key Words and Phrases: Ensemble clustering, spectral clustering, co-association matrix, low-rank representation, non-convex relaxation

ACM Reference format:
Zhiqiang Tao, Hongfu Liu, Sheng Li, Zhengming Ding, and Yun Fu. 2019. Robust Spectral Ensemble Clustering via Rank Minimization. *ACM Trans. Knowl. Discov. Data* 13, 1, Article 4 (January 2019), 25 pages.
https://doi.org/10.1145/3278606

This work is supported in part by the NSF IIS award 1651902 and U.S. Army Research Office Award W911NF-17-1-0367.

Authors' addresses: Z. Tao, Department of Electrical and Computer Engineering, Northeastern University, Boston, MA 02115; email: zqtao@ece.neu.edu; H. Liu, Michtom School of Computer Science, Brandeis University, Waltham, MA 02453; email: hongfuliu@brandeis.edu; S. Li, Department of Computer Science, University of Georgia, Athens, GA 30602; email: sheng.li@uga.edu; Z. Ding, Department of Computer, Information and Technology, Indiana University - Purdue University Indianapolis, Indianapolis, IN 46202; email: zd2@iu.edu; Y. Fu, Department of Electrical and Computer Engineering, College of Computer and Information Science, Northeastern University, Boston, MA 02115; email: yunfu@ece.neu.edu.

1556-4681/2019/01-ART4 $15.00
https://doi.org/10.1145/3278606

1 INTRODUCTION

Data clustering is a core technique for data analysis, the objective of which is to partition a set of unlabeled patterns, points, or objects into natural groups or clusters (Jain 2010). It has been attracting a lot of attention throughout the fields of data mining, machine learning, information retrieval, biology and computer vision. In general, numerous clustering algorithms are based on different assumptions, such as connectivity, centroid, distribution, density and subspace, leading to the fact that they usually generate different clustering results. Thus, it is difficult to employ a single method to handle various cluster structures in real-world datasets, and even hard to decide which one to use for a particular dataset. In light of this, Ensemble Clustering (EC) comes into being (Strehl and Ghosh 2003; Fred and Jain 2005), which is also known as *consensus clustering* (Topchy et al. 2005) or *clustering aggregation* (Gionis et al. 2007).

EC emerges as a powerful alternative to the traditional clustering method, which has the ability of boosting clustering performance, improving robustness and stability, discovering novel clusters, and fusing multi-source information (Wu et al. 2015). It takes as input a set of Basic Partitions (BPs) and targets to integrate them into a consensus one. Hence, BPs *generation* and *aggregation* are two fundamental steps for developing an EC system. According to Topchy et al. (2005), BPs could be generated by three common strategies as follows: (1) running different clustering algorithms on the same dataset (Fern and Brodley 2004; Yi et al. 2012); (2) generating BPs with random sampling (Minaei-Bidgoli et al. 2004); (3) performing the same clustering method with various settings (e.g., different parameters or initialization), such as the Random Parameter Selection (RPS) strategy (Fred and Jain 2005; Wu et al. 2015), which obtains BPs by running K-means algorithm with different cluster numbers. RPS is one of the most successful BPs generation strategies (Kuncheva and Vetrov 2006; Iam-on et al. 2011), as it provides a highly efficient way to generate diverse clustering results for uncovering arbitrary cluster structures (Fred and Jain 2005). In this article, we mainly adopt the RPS strategy and focus on the BPs integration.

In terms of BPs aggregation manner, EC methods can be roughly divided into two classes, first, the utility function based methods and, second, the co-association matrix based ones. The first class (Topchy et al. 2003, 2005; Wu et al. 2015) usually employs a utility function to measure the similarity between consensus partition and multiple BPs, and then achieves ensemble by solving a maximization problem. The second class (Fred and Jain 2005; Zheng et al. 2014; Liu et al. 2015a) generally summarizes BPs as a co-association matrix, and transforms EC into a graph partitioning problem. The co-association matrix is actually a pair-wise similarity matrix on an instance-level. It calculates the times of any two instances being divided into the same cluster upon the categorical data provided by BPs. Among previous works, co-association matrix is of particular interest, as it allows to solve EC problem with an easy and effective way, i.e., by conducting any graph partitioning algorithms, such as agglomerative clustering (Fred and Jain 2005), hierarchical clustering (Zheng et al. 2014), and spectral clustering (Liu et al. 2015a).

Most existing EC methods directly utilize the co-association matrix, yet without considering various noises from BPs. These noises are induced by disagreement among multiple BPs or caused by the outlier ones, which can severely undermine the structure of a co-association matrix, mislead the graph partitioning process, and hence degrade the clustering performance. Unfortunately, this issue is still under explored. Only a few research efforts (Yi et al. 2012; Zhou et al. 2015) have been made to address it. For example, Yi et al. (2012) recovered the co-association matrix as a low-rank one by using matrix completion, where the disagreement between BPs was treated as uncertain data pairs and labeled as missing values. For another example, Zhou et al. (2015) captured the outliers from BPs by minimizing the Kullback–Leibler (KL) divergence between co-association matrix and each BP with a low-rank constraint. It is worth noting that, although these two reasonable methods improve the robustness of a co-association matrix, they both take more attention on a denoising task, rather than enhancing the cluster structure inside co-association matrix. Moreover, their model is learned away from consensus clustering, which lacks the guidance of consensus partition and neglects to provide a unified framework.

In this article, we propose a novel Robust Spectral Ensemble Clustering (RSEC) approach to address the above challenges. We aim to learn a robust co-association matrix and find the consensus partition simultaneously. Previous research efforts (Liu et al. 2013; Yin et al. 2016; Li and Fu 2014) have shown that Low-Rank Representation (LRR) can handle various noises and uncover the membership between data points with a block-diagonal form. This nice property naturally enables RSEC to learn a robust representation for co-association matrix by leveraging LRR, which could reveal the cluster structure and facilitate the graph partitioning process. Moreover, ensemble clustering is seamlessly incorporated into our learning process. Specifically, we obtain the consensus partition by running spectral clustering on the learned LRR codes with a trace minimization form. By this means, our optimization framework jointly performs LRR learning and ensemble clustering task. In particular, since the consensus partition usually exhibits a high clustering performance, we employ it to iteratively enhance the block-diagonal structure of the learned representation. To solve RSEC, we formulate it by using nuclear norm as a convex proxy to the rank function. However, since the nuclear norm works as the ℓ_1 norm of singular values, it may give a biased estimation to the matrix rank (Wang et al. 2013; Lu et al. 2014; Peng et al. 2015). To alleviate this problem, we further develop a non-convex model for RSEC and provide a solution with the majorization–minimization Augmented Lagrange Multiplier (MM-ALM) method. The contributions of this article are summarized as the following four-folds:

- We propose a unified optimization framework (see Figure 1) to simultaneously learn a robust representation for the co-association matrix and find the final consensus partition.
- The spectral graph on a co-association matrix is constructed by its LRR along with the consensus partition. By this means, we introduce an interaction between representation learning and ensemble clustering. The block-diagonal structure of the learned representation is iteratively enhanced by consensus partition during the optimization process, which better uncovers the cluster structure of a co-association matrix.
- Two effective solutions are provided for RSEC, where we adopt two strategies to address the rank minimization problem, i.e., a convex formulation and a non-convex one. Empirical evidence shows the non-convex RSEC exhibits more robustness.
- Experiments conducted on 18 real-world datasets demonstrate the effectiveness of our approach over the state-of-the-art EC methods. Moreover, we also explore several impact factors that may affect the final clustering performance, including BPs number, incomplete BPs and different BPs generation strategy.

This article is an extension to our previous work (Tao et al. 2016). Compared with Tao et al. (2016), we propose a general model to our approach, and provide a non-convex alternative formulation, as well as more theoretical analysis, model discussion and experimental evaluations.

The remainder of this article is organized as follows. In Section 2, we give a brief introduction to ensemble clustering and low-rank matrix analysis. In Section 3, we present the proposed RSEC model and provide a solution to its convex formulation. In Section 4, we propose a non-convex relaxation to our model and develop the MM-ALM algorithm to solve it. Extensive experimental results and discussions are reported in Section 5, followed by a final conclusion in Section 6.

Notations. A vector (matrix) is denoted by a lowercase (uppercase) letter in boldface. Given a matrix $\mathbf{U}$, $\mathbf{U}^{\mathrm{T}}$, $\mathbf{U}^{-1}$, $\mathrm{rank}(\mathbf{U})$ and $\mathrm{tr}(\mathbf{U})$ represent its transpose, inverse, rank, and trace, respectively. Let $\mathbf{U}_i$ be the ith column of $\mathbf{U}$ and $\mathbf{U}_{ij}$ the jth element in $\mathbf{U}_i$, then we have the definitions of matrix norms as the following. $\|\mathbf{U}\|_0$, $\|\mathbf{U}\|_1$, $\|\mathbf{U}\|_{\mathrm{F}}$, and $\|\mathbf{U}\|_*$ stand for the ℓ_0 norm (number of all the

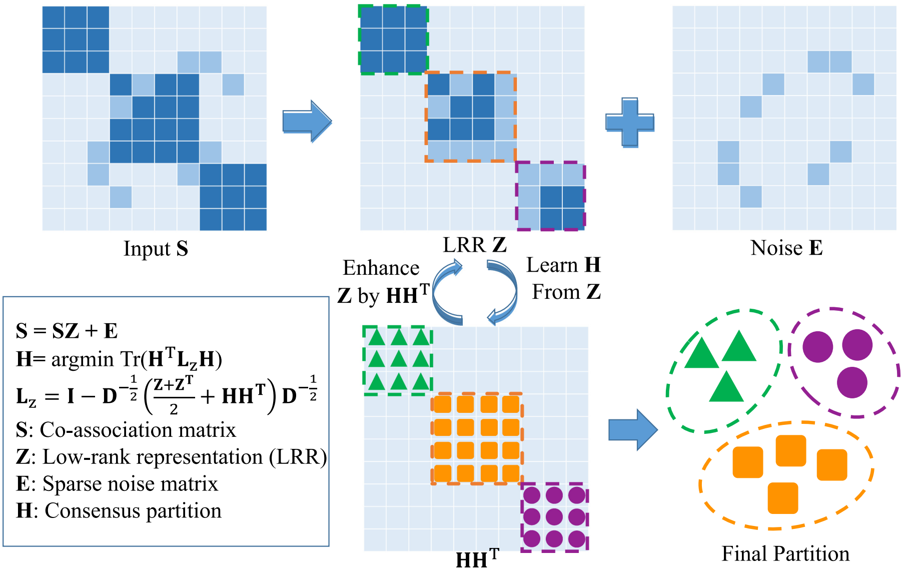

Input **S**
LRR **Z**
Noise **E**
Enhance **Z** by $\mathbf{HH}^{\mathrm{T}}$
Learn **H** From **Z**
$\mathbf{S} = \mathbf{SZ} + \mathbf{E}$
$\mathbf{H}$= argmin $\mathrm{Tr}(\mathbf{H}^{\mathrm{T}}\mathbf{L}_{\mathbf{Z}}\mathbf{H})$
$\mathbf{L}_{\mathbf{Z}} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\left(\frac{\mathbf{Z}+\mathbf{Z}^{\mathrm{T}}}{2} + \mathbf{HH}^{\mathrm{T}}\right)\mathbf{D}^{-\frac{1}{2}}$
S: Co-association matrix
Z: Low-rank representation (LRR)
E: Sparse noise matrix
H: Consensus partition
$\mathbf{HH}^{\mathrm{T}}$
Final Partition

Fig. 1. Illustration of the proposed RSEC, where we jointly learn low-rank representation **Z** for the co-association matrix **S** and find consensus partition **H** by running spectral clustering with $\mathbf{L}_Z$. We employ **Z** to reveal the cluster structure of **S**, and capture the noises inside **S** with a sparse error matrix **E**. During the learning process, **H** is used to iteratively enhance the block-diagonal structure of **Z**. The final clustering result could be obtained either from **H** or **Z**.

non-zero entries), ℓ_1 norm ($\sum_{i,j} |\mathbf{U}_{i,j}|$), Frobenius norm ($\sqrt{\sum_{i,j} \mathbf{U}_{i,j}^2}$) and nuclear norm (sum of all the singular values of **U**), respectively. Besides, $||\mathbf{U}||_{2,0}$ represents the $\ell_{2,0}$ norm ($\#\{i : \sum_j \mathbf{U}_{i,j}^2 \neq 0\}$) and $||\mathbf{U}||_{2,1}$ denotes the $\ell_{2,1}$ norm ($\sum_i \sqrt{\sum_j \mathbf{U}_{i,j}^2}$). $\langle \mathbf{U}, \mathbf{V} \rangle$ is the inner product of two matrices with appropriate dimensions, which is equal to $\mathrm{tr}(\mathbf{U}^{\mathrm{T}}\mathbf{V})$. Moreover, $\mathrm{diag}(\cdot)$ is the diagonalization operator that converts a vector into a diagonal matrix, and **I** is the identity matrix with compatible size.

2 RELATED WORK

In general, our approach improves the robustness of ensemble clustering by learning a robust representation for co-association matrix through low-rank constraint. In this section, we give a brief introduction to ensemble clustering and low-rank matrix analysis, respectively.

2.1 Ensemble Clustering

Ensemble clustering has attracted a lot of research efforts, which can be roughly divided into two categories, such as the utility function based methods and co-association matrix based ones.

The utility function based methods compute the similarity between consensus partition and multiple BPs by an explicit objective function. They usually obtain the final clustering result by solving a maximization problem. For instance, Topchy et al. (2003) proposed a quadratic mutual information based objective function for consensus clustering, which actually employed the category utility function (Mirkin 2001) and found a solution by using *K*-means clustering. This idea was further extended in their work (Topchy et al. 2004, 2005), where they solved ensemble clustering by using expectation-maximization (EM) algorithm with a finite mixture of multinomial distributions. Along this line, Wu et al. (2015) transferred the ensemble clustering into a *K*-means clustering problem with KCC utility functions and gave the necessary and sufficient conditions of using them. In addition, there are some other interesting objective functions for ensemble clustering, such as the ones based on non-negative matrix factorization (Li et al. 2007), simulated annealing (Lu et al. 2008), and kernel-based methods (Vega-Pons et al. 2010).

The co-association matrix based methods summarize the categorical data of BPs into a pairwise similarity matrix, which counts the times of any two instances being divided into the same cluster. The co-association matrix actually represents the membership of all the data points in a partition space. Thus, any graph partitioning algorithms can be conducted on it to obtain the final clustering result. Strehl and Ghosh (2003) developed three graph-based algorithms for consensus clustering (i.e., cluster-based similarity graph partitioning, hyper-graph partitioning and meta-clustering algorithm) and returned the best result according to the normalized mutual information measurement. Following this line, Fern and Brodley (2004) built a bipartite graph to further improve the clustering quality. These two representative works both involved a pairwise similarity matrix computed by BPs. To our best knowledge, the definition of co-association matrix was first introduced by Fred and Jain (2005), where they applied the agglomerative hierarchical clustering to find consensus partition. After that, a fully discussion about hierarchical ensemble clustering was given by Zheng et al. (2014). Recently, Liu et al. (2015a) proposed a spectral ensemble clustering method, which ran spectral clustering on the co-association matrix and transformed it as a weighted *K*-means problem to achieve high efficiency. Other methods include genetic algorithm based methods (Yoon et al. 2006), relabeling and voting (Ayad and Kamel 2008), locally adaptive cluster based methods (Domeniconi and Al-Razgan 2009), simultaneous clustering and ensemble (Tao et al. 2017a), multi-view ensemble clustering (Tao et al. 2017b), and infinite ensemble (Liu et al. 2016, 2018).

Most existing ensemble clustering methods directly integrate BPs without considering the detrimental effect of noises. A few efforts (Yi et al. 2012; Zhou et al. 2015) have been made to improve the robustness of co-association matrix. They mainly target at a denoising task by using low-rank matrix recovery. In contrast, our approach learns a LRR for the co-association matrix, which not only reduces its noises, but also highlight its block-diagonal structure. Moreover, we find the consensus partition within a unified optimization framework.

2.2 Low-Rank Matrix Analysis

Low-rank matrix analysis has been widely used in the fields of machine learning, data mining and computer vision, such as matrix completion (Wright et al. 2009; Candès et al. 2011), subspace learning (Liu and Yan 2011; Liu et al. 2013; Li and Fu 2014, 2016; Li et al. 2018a), transfer learning (Shao et al. 2014; Ding and Fu 2014; Ding et al. 2015), multi-view learning (Li et al. 2017a, 2018b), and image segmentation (Cheng et al. 2011; Cao et al. 2014; Li et al. 2016a). It aims to recover clean data from the samples containing various noises. Generally speaking, it has two representative forms as Robust PCA (Wright et al. 2009; Candès et al. 2011) and LRR (Liu et al. 2010, 2013). Given an observed data matrix $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n] \in \mathbb{R}^{d\times n}$, of which each column vector $\mathbf{x}_i \in \mathbb{R}^d$ denotes a sample, Robust PCA decomposes $\mathbf{X}$ into a low-rank matrix $\mathbf{A}$ and a sparse residual matrix $\mathbf{E}$, where $\mathbf{A}$ represents the clean data and $\mathbf{E}$ captures noises. Clearly, robust PCA implies data lying in a single subspace and is mainly used for low-rank matrix recovery and completion. On the other hand, LRR assumes data are drawn from a union of low-dimensional subspaces and tries to segment these subspaces by learning the lowest rank representation $\mathbf{Z}$ for $\mathbf{X}$: $\min_{\mathbf{Z},\mathbf{E}} \ \mathrm{rank}(\mathbf{Z}) + \lambda\|\mathbf{E}\|_0$, subject to $\mathbf{X} = \mathbf{X}\mathbf{Z} + \mathbf{E}$, where $\lambda > 0$ balances the rankness of $\mathbf{Z}$ and the sparseness of $\mathbf{E}$. In practice, since solving LRR is an NP-hard problem, we usually tackle its convex relaxation, i.e., using nuclear norm to estimate $\mathrm{rank}(\mathbf{Z})$ and ℓ_1 or $\ell_{2,1}$ norm to approximate $\|\mathbf{E}\|_0$. It is worth noting that, LRR is in essence a generalization for Robust PCA, as $\mathbf{XZ}$ also exhibits low-rank property with the minimizer of $\mathbf{Z}$ (Liu et al. 2013). By expressing $\mathbf{X}$ with itself, $\mathbf{Z}$ works as a pairwise affinity matrix that computes the similarity between data points. Moreover, it has been proven that **Z** enjoys a nice block-diagonal property (Liu et al. 2013; Yin et al. 2016), which is able to uncover the global structure of data and further facilitate the clustering process. In light of this, we employ LRR to learn a robust representation for the co-association matrix, rather than directly recover it as a low-rank one. However, one drawback of LRR is that the nuclear norm is a biased estimator to matrix rank, as it over-penalizes the large singular values (Wang et al. 2013; Lu et al. 2014; Peng et al. 2015). Thus, in this article, we also present a non-convex RSEC model to achieve more robustness for rank approximation.

3 ROBUST SPECTRAL ENSEMBLE CLUSTERING

In this section, we first propose a general model to our RSEC method and formulate it as a convex optimization problem. Then, the solution based on Augmented Lagrange Multiplier (ALM) is elaborated, followed by an analysis of convergence and complexity to our algorithm.

3.1 Preliminary

Let $\mathcal{X} = \{x_1, x_2, \dots, x_n\}$ be a set of n data points independently sampled from K clusters, represented as $C = \{C_1, \dots, C_K\}$. Denote $\Pi = \{\pi_1, \pi_2, \dots, \pi_m\}$ as m input BPs, each of which divides $\mathcal{X}$ into K_i crisp partitions and maps $\mathcal{X}$ into a label set $\pi_i = \{\pi_i(x_1), \pi_i(x_2), \dots, \pi_i(x_n)\}$, where K_i is the cluster number for the i^{th} BP, $1 \le \pi_i(x_j) \le K_i$, and $1 \le i \le m$, $1 \le j \le n$. Note that, the cluster number of each BP is usually set to be different for achieving the diversity among input BPs, which has been recognized as an efficient manner to ensure the success for ensemble clustering (Fred and Jain 2005; Wu et al. 2015).

The goal of ensemble clustering is to find the consensus partition that agrees with input BPs most and divides $\mathcal{X}$ into its original K clusters. Commonly, EC methods summarize m BPs as a co-association matrix, and then conduct a graph partitioning algorithm to obtain the final consensus clustering, denoted as π. The co-association matrix $\mathbf{S} \in \mathbb{R}^{n \times n}$ actually calculates the times of two instances occurring in the same cluster based on Π, which is defined by Fred and Jain (2005) as

$$\mathbf{S}(x_p, x_q) = \sum_{i=1}^{m} \delta(\pi_i(x_p), \pi_i(x_q)), \tag{1}$$

where $x_p, x_q \in \mathcal{X}$ and $\delta(a, b)$ is 1 if $a = b$; 0 otherwise.

Obviously, $\mathbf{S}$ could be normalized by $\mathbf{S} = \mathbf{S}/m$. Inspired by Liu et al. (2015a), we apply the spectral clustering on the co-association matrix $\mathbf{S}$, and have its trace minimization form by following Luxburg (2007) as

$$\min_{\mathbf{H}} \operatorname{tr}(\mathbf{H}^{\mathrm{T}}\mathbf{L}_s\mathbf{H}) \text{ s.t. } \mathbf{H}^{\mathrm{T}}\mathbf{H} = \mathbf{I}, \tag{2}$$

where $\mathbf{L}_s = \mathbf{I} - \mathbf{D}_s^{-1/2}\mathbf{S}\mathbf{D}_s^{-1/2}$ is the normalized Laplacian matrix of $\mathbf{S}$, with degree matrix $\mathbf{D}_s \in \mathbb{R}^{n \times n}$ being a diagonal matrix whose jth diagonal element is the sum of the jth row of $\mathbf{S}$, and $\mathbf{H} \in \mathbb{R}^{n \times K}$ is defined as the scaled partition matrix of the consensus partition π:

$$\mathbf{H}_{jk} = \begin{cases} 1/\sqrt{|C_k|}, & \text{if } x_j \in C_k \text{ in } \pi \\ 0, & \text{otherwise} \end{cases}. \tag{3}$$

3.2 Problem Formulation

According to Equation (2), the intrinsic structure of co-association matrix $\mathbf{S}$ is the key factor to ensemble clustering. Ideally, $\mathbf{S}$ should be a low-rank matrix ($K \ll n$), and enjoys a clear cluster structure of K block-diagonal. However, since co-association matrix is directly computed by multiple BPs, its cluster structure could be easily undermined by the noises from BPs.

LRR is capable of handling noises and revealing the membership between data points with a block-diagonal form Liu et al. (2013); thus, it is natural to learn a robust representation for the co-association matrix S by leveraging LRR. However, LRR only guarantees the low-rank property of the learned representation, yet without considering the ensemble clustering task. Therefore, we propose our RSEC method to simultaneously learn a robust representation for S and find the consensus partition with a unified optimization framework. Given a normalized co-association matrix S, the objective function of our RSEC is formulated as

$$\begin{aligned} &\min_{\mathbf{H},\mathbf{Z},\mathbf{E}} \operatorname{tr}(\mathbf{H}^{\mathrm{T}}\mathbf{L}_z\mathbf{H}) + \lambda_1 \operatorname{rank}(\mathbf{Z}) + \lambda_2\|\mathbf{E}\|_{\ell} \\ &\text{s.t. } \mathbf{H}^{\mathrm{T}}\mathbf{H} = \mathbf{I},\ \mathbf{S} = \mathbf{S}\mathbf{Z} + \mathbf{E}, \end{aligned} \tag{4}$$

with

$$\mathbf{L}_z = \mathbf{I} - \mathbf{D}_z^{-1/2}((\mathbf{Z}+\mathbf{Z}^{\mathrm{T}})/2 + \mathbf{H}\mathbf{H}^{\mathrm{T}})\mathbf{D}_z^{-1/2}, \tag{5}$$

where $\mathbf{H}$ denotes the consensus partition, $\mathbf{Z} \in \mathbb{R}^{n\times n}$ is the learned representation, $\mathbf{E} \in \mathbb{R}^{n\times n}$ is an error matrix that tries to capture various noises inside $\mathbf{S}$, $\|\cdot\|_{\ell}$ indicates a specific sparse regularization strategy (e.g., ℓ_0 norm or $\ell_{2,0}$ norm), and $\lambda_1, \lambda_2 > 0$ are two penalty parameters that balance the corresponding terms. In Equation (5), $\mathbf{L}_z$ is designed as a normalized Laplacian matrix of the graph constructed by $\mathbf{Z}$ and $\mathbf{H}$, and the degree matrix $\mathbf{D}_z$ is computed by

$$\mathbf{D}_z = \operatorname{diag}([d_1, \ldots, d_n]), \tag{6}$$

where d_j, $1 \le j \le n$, is the sum of the jth row of the matrix $(\mathbf{Z}+\mathbf{Z}^{\mathrm{T}})/2 + \mathbf{H}\mathbf{H}^{\mathrm{T}}$. Here, $(\mathbf{Z}+\mathbf{Z}^{\mathrm{T}})/2$ is employed to obtain a symmetric graph. Moreover, since $\mathbf{H}$ is a high-quality clustering result, $\mathbf{H}\mathbf{H}^{\mathrm{T}}$ also works as an affinity matrix with clear cluster structure. Thus, we use it to iteratively enhance the block-diagonal structure of $\mathbf{Z}$ during the optimization process.

In the objective function of RSEC, we minimize $\operatorname{rank}(\mathbf{Z})$ subject to the *self-expressiveness* property (Liu et al. 2013) (i.e., $\mathbf{S} = \mathbf{S}\mathbf{Z} + \mathbf{E}$) to seek for the LRR of $\mathbf{S}$. It is worth noting that, either ℓ_0 norm or $\ell_{2,0}$ norm could be used as a sparse regularization on $\mathbf{E}$, since it is reasonable to assume the noises randomly appear on $\mathbf{S}$ due to disagreements between BPs, or mainly occur on some specific instances (i.e., rows in $\mathbf{S}$) that may easily induce outlier partition results. The spectral clustering is conducted on $\mathbf{L}_z$ to find the consensus partition $\mathbf{H}$. By using $\mathbf{L}_z$ in Equation (5), RSEC incorporates an interaction between learning $\mathbf{Z}$ and finding $\mathbf{H}$.

Remark 3.1. Different from the Laplacian Regularized LRR problem (Yin et al. 2016), where the graph is fixed by the predefined affinity matrix, the graph defined by Equation (5) is iteratively updated in our method. Moreover, the Laplacian term is used for different purposes. In (Yin et al. 2016), it works as a regularization for LRR to hold the structure of the graph. However, the proposed RSEC employs the Laplacian term to conduct spectral clustering on the learned representation $\mathbf{Z}$, which finds the final consensus partition $\mathbf{H}$.

Clearly, Equation (4) is an NP-hard problem. By following Liu et al. (2013), we relax RSEC as

$$\begin{aligned} &\min_{\mathbf{H},\mathbf{Z},\mathbf{E}} \operatorname{tr}(\mathbf{H}^{\mathrm{T}}\mathbf{L}_z\mathbf{H}) + \lambda_1\|\mathbf{Z}\|_* + \lambda_2\|\mathbf{E}\|_{2,1} \\ &\text{s.t. } \mathbf{H}^{\mathrm{T}}\mathbf{H} = \mathbf{I},\ \mathbf{S} = \mathbf{S}\mathbf{Z} + \mathbf{E}, \end{aligned} \tag{7}$$

where the nuclear norm $\|\mathbf{Z}\|_*$ is a convex envelope of $\operatorname{rank}(\mathbf{Z})$, and the $\ell_{2,1}$ norm $\|\mathbf{E}\|_{2,1}$ is a convex proxy to $\|\mathbf{E}\|_{2,1}$. In the next, we will give more details about how to solve Equation (7) with an iterative optimization manner.

3.3 Optimization

The problem of Equation (7) is hard to solve, since it is not jointly convex for **Z** and **H**. However, we may divide it into several subproblems and optimize each of them by fixing the other variables. The ALM method (Lin et al. 2010, 2011) comes to mind as an efficient and effective solver to our problem. To apply ALM, we first introduce an auxiliary variable **J** to make Equation (7) separable, and equivalently convert it as

$$\begin{aligned} &\min_{\mathbf{H},\mathbf{Z},\mathbf{E}} \operatorname{tr}(\mathbf{H}^{\mathrm{T}}\mathbf{L}_z\mathbf{H}) + \lambda_1\|\mathbf{J}\|_* + \lambda_2\|\mathbf{E}\|_{2,1} \\ &\text{s.t. } \mathbf{H}^{\mathrm{T}}\mathbf{H} = \mathbf{I},\ \mathbf{S} = \mathbf{S}\mathbf{Z} + \mathbf{E},\ \mathbf{Z} = \mathbf{J}. \end{aligned} \tag{8}$$

Following Dhillon et al. (2004), the constraint $\mathbf{H}^{\mathrm{T}}\mathbf{H} = \mathbf{I}$ is relaxed to avoid hard partition during the optimization process. The augmented Lagrangian function of Equation (8) could be written as

$$\begin{aligned} \mathcal{L} = {} & \operatorname{tr}(\mathbf{H}^{\mathrm{T}}\mathbf{L}_z\mathbf{H}) + \lambda_1\|\mathbf{J}\|_* + \lambda_2\|\mathbf{E}\|_{2,1} + \langle \mathbf{Y}_1, \mathbf{S} - \mathbf{S}\mathbf{Z} - \mathbf{E}\rangle \\ & + \langle \mathbf{Y}_2, \mathbf{Z} - \mathbf{J}\rangle + \frac{\mu}{2}\left(\|\mathbf{S} - \mathbf{S}\mathbf{Z} - \mathbf{E}\|_{\mathrm{F}}^2 + \|\mathbf{Z} - \mathbf{J}\|_{\mathrm{F}}^2\right), \end{aligned} \tag{9}$$

where $\mathbf{Y}_1$ and $\mathbf{Y}_1$ are two Lagrangian multipliers, and $\mu > 0$ is a penalty parameter.

The ALM solver solves Equation (9) with an iterative update manner, which addresses **J**, **Z**, **E**, and **H** in sequence and optimizes one variable by fixing the others. More details are given in the following.

Update **J**. We first minimize $\mathcal{L}$ with respect to **J**, and obtain $\mathbf{J}^{(t+1)}$ by

$$\begin{aligned} \mathbf{J}^{(t+1)} &= \operatorname*{argmin}_{\mathbf{J}} \lambda_1\|\mathbf{J}\|_* + \left\langle \mathbf{Y}_2^{(t)}, \mathbf{Z}^{(t)} - \mathbf{J}\right\rangle + \frac{\mu^{(t)}}{2}\|\mathbf{Z}^{(t)} - \mathbf{J}\|_{\mathrm{F}}^2 \\ &= \operatorname*{argmin}_{\mathbf{J}} \frac{\lambda_1}{\mu^{(t)}}\|\mathbf{J}\|_* + \frac{1}{2}\left\|\mathbf{J} - \left(\mathbf{Z}^{(t)} + \frac{1}{\mu^{(t)}}\mathbf{Y}_2^{(t)}\right)\right\|_{\mathrm{F}}^2. \end{aligned} \tag{10}$$

Equation (10) could be solved with a closed-form solution as

$$\mathbf{J}^{(t+1)} = \Theta_{\frac{\lambda_1}{\mu^{(t)}}}\left(\mathbf{Z}^{(t)} + \frac{1}{\mu^{(t)}}\mathbf{Y}_2^{(t)}\right), \tag{11}$$

where $\Theta(\cdot)$ denotes the Singular Value Thresholding (SVT) operator (Cai et al. 2010).

Update **Z**. By substituting Equation (5) into $\mathcal{L}$ and dropping unrelated terms, the subproblem for updating **Z** is equivalent to the following:

$$\begin{aligned} \mathbf{Z}^{(t+1)} = \operatorname*{argmin}_{\mathbf{Z}} & -\frac{1}{2}\operatorname{tr}\left(\mathbf{H}^{(t)\mathrm{T}}\mathbf{D}_z^{-1/2}(\mathbf{Z} + \mathbf{Z}^{\mathrm{T}})\mathbf{D}_z^{-1/2}\mathbf{H}^{(t)}\right) + \left\langle \mathbf{Y}_1^{(t)}, \mathbf{S} - \mathbf{S}\mathbf{Z} - \mathbf{E}^{(t)}\right\rangle \\ & + \left\langle \mathbf{Y}_2^{(t)}, \mathbf{Z} - \mathbf{J}^{(t+1)}\right\rangle + \frac{\mu^{(t)}}{2}\left(\|\mathbf{S} - \mathbf{S}\mathbf{Z} - \mathbf{E}^{(t)}\|_{\mathrm{F}}^2 + \|\mathbf{Z} - \mathbf{J}^{(t+1)}\|_{\mathrm{F}}^2\right). \end{aligned} \tag{12}$$

Note that the derivative of $\mathbf{D}_z$ with respect to **Z** is relatively complex, which actually complicates the solution of obtaining $\mathbf{Z}^{(t+1)}$. To simplify the solution, the trick employed here is to fix $\mathbf{D}_z$ as a constant, and update it later by using its definition with $\mathbf{Z}^{(t+1)}$ and $\mathbf{H}^{(t)}$. Though it is an approximate way to take the derivative for **Z**, it has very little effect on the convergence property to our method, as shown in the experiments.

By fixing $\mathbf{D}_z$, Equation (12) becomes a quadratic problem of $\mathbf{Z}$. Thus, taking the derivative of $\mathcal{L}$ with respect to $\mathbf{Z}$ as zero, we obtain

$$\begin{aligned}\mathbf{Z}^{(t+1)} = {} & (\mathbf{S}\mathbf{S}^{\mathrm{T}} + \mathbf{I})^{-1}\Big(\mathbf{S}^{\mathrm{T}}\mathbf{S} + \mathbf{J}^{(t+1)} - \mathbf{S}^{\mathrm{T}}\mathbf{E}^{(t)} \\ & + \frac{1}{\mu^{(t)}}\Big(\mathbf{S}^{\mathrm{T}}\mathbf{Y}_1^{(t)} - \mathbf{Y}_2^{(t)} + \mathbf{D}_z^{-1/2}\mathbf{H}^{(t)}\mathbf{H}^{(t)\mathrm{T}}\mathbf{D}_z^{-1/2}\Big)\Big).\end{aligned} \tag{13}$$

Update $\mathbf{E}$. The subproblem of updating $\mathbf{E}$ can be recast as

$$\begin{aligned}\mathbf{E}^{(t+1)} &= \underset{\mathbf{E}}{\operatorname{argmin}}\, \lambda_2\|\mathbf{E}\|_{2,1} + \langle \mathbf{Y}_1^{(t)}, \mathbf{S} - \mathbf{S}\mathbf{Z}^{(t+1)} - \mathbf{E}\rangle + \frac{\mu^{(t)}}{2}\|\mathbf{S} - \mathbf{S}\mathbf{Z}^{(t+1)} - \mathbf{E})\|_{\mathrm{F}}^2 \\ &= \underset{\mathbf{E}}{\operatorname{argmin}}\, \frac{\lambda_2}{\mu^{(t)}}\|\mathbf{E}\|_{2,1} + \frac{1}{2}\left\|\mathbf{E} - \left(\mathbf{S} - \mathbf{S}\mathbf{Z}^{(t+1)} + \frac{\mathbf{Y}_1^{(t)}}{\mu^{(t)}}\right)\right\|_{\mathrm{F}}^2.\end{aligned} \tag{14}$$

Following the lemma provided in Liu et al. (2013), we may solve this subproblem column-wisely, where each column of $\mathbf{E}^{(t+1)}$ has a closed-form solution as

$$\forall i\ \mathbf{E}_i^{(t+1)} = \begin{cases} \frac{\|\mathbf{Q}_i\|_2 - \lambda_2/\mu^{(t)}}{\|\mathbf{Q}_i\|_2}, & \text{if } \|\mathbf{Q}_i\|_2 > \lambda_2/\mu^{(t)} \\ 0, & \text{otherwise} \end{cases}, \tag{15}$$

where $\|\mathbf{Q}_i\|_2$ denotes the ℓ_2 norm of a column vector, $1 \le i \le n$, and $\mathbf{Q} \equiv \mathbf{S} - \mathbf{S}\mathbf{Z}^{(t+1)} + \frac{\mathbf{Y}_1^{(t)}}{\mu^{(t)}}$.

Update $\mathbf{H}$. To solve the subproblem of $\mathbf{H}$, we first update $\mathbf{L}_z$ and $\mathbf{D}_z$ by using their definition in Equation (5) and Equation (6) with $\mathbf{Z}^{(t+1)}$ and $\mathbf{H}^{(t)}$. Then, $\mathbf{H}^{(t+1)}$ is obtained by

$$\begin{aligned}\mathbf{H}^{(t+1)} &= \underset{\mathbf{H}}{\operatorname{argmin}}\, \operatorname{tr}(\mathbf{H}^{\mathrm{T}}\mathbf{L}_z\mathbf{H}), \\ \mathbf{L}_z &= \mathbf{I} – \mathbf{D}_z^{-1/2}((\mathbf{Z}^{(t+1)} + \mathbf{Z}^{(t+1)\mathrm{T}})/2 + \mathbf{H}^{(t)}\mathbf{H}^{(t)\mathrm{T}})\mathbf{D}_z^{-1/2}.\end{aligned} \tag{16}$$

Since $\mathbf{H}^{(t)}$ is a constant as the consensus partition obtained by last iteration, Equation (16) is still quadratic w.r.t. $\mathbf{H}$. Actually, solving the subproblem of $\mathbf{H}$ is equivalent to the normalized spectral clustering on $\mathbf{L}_z$. Thus, a well-known solution for Equation (16) is to set $\mathbf{H}^{(t+1)}$ as the smallest K eigenvectors of the $\mathbf{L}_z$ (Shi and Malik 2000; Ng et al. 2001; Luxburg 2007).

ALGORITHM 1: Robust Spectral Ensemble Clustering

```
Input: a set of basic partitions $\Pi = \{\pi_1, \pi_2, \ldots, \pi_m\}$, cluster number $K$, two parameters $\lambda_1, \lambda_2 > 0$.
Initial: $\mathbf{J}^0 = \mathbf{Z}^0 = \mathbf{E}^0 = \mathbf{0} \in \mathbb{R}^{n \times n}$, $\mathbf{Y}_1^0 = \mathbf{Y}_2^0 = \mathbf{0} \in \mathbb{R}^{n \times n}$, $\mathbf{H}^0 = \mathbf{0} \in \mathbb{R}^{n \times K}$, $\rho = 1.1, \mu_0 = 10^{-6}, \mu_{\max} = 10^{10}$,
    and $t = 0$.
  Derive S from $\Pi$ via Equation (1) and normalize it by $\mathbf{S} = \mathbf{S}/m$;
  while not converged do
      Update $\mathbf{J}^{(t+1)}$ via Equation (11);
      Update $\mathbf{Z}^{(t+1)}$ via Equation (13);
      Update $\mathbf{E}^{(t+1)}$ via Equation (15);
      Compute $\mathbf{D}_z$ and $\mathbf{L}_z$ based on $\mathbf{Z}^{(t+1)}$ and $\mathbf{H}^{(t)}$ via Equation (5–6);
      Set $\mathbf{H}^{(t+1)}$ as the smallest $K$ eigenvectors of $\mathbf{L}_z$;
      Update the Lagrangian multipliers via Equation (17);
      $\mu^{(t+1)} = \min(\rho\mu^{(t)}, \mu_{\max})$, $t = t + 1$;
  end while
  Run $K$-means on H or spectral clustering on Z to obtain the final partition $\pi$.
Output: the final partition $\pi$.
```

Remark 3.2. Co-association matrix is in essence a pairwise similarity matrix upon the partition space, which transforms ensemble clustering into a graph partitioning problem such as Equation (16). Hence, $(\mathbf{Z}^{(t+1)} + \mathbf{Z}^{(t+1)\mathrm{T}})/2 + \mathbf{H}^{(t)}\mathbf{H}^{(t)\mathrm{T}}$ serves as a better affinity graph than the original $\mathbf{S}$: (1) the LRR $\mathbf{Z}^{(t+1)}$ alleviates the noise existing in $\mathbf{S}$, and thus highlight the cluster structure; (2) the high-quality consensus partition $\mathbf{H}^{(t)}$ is incorporated into the affinity graph to further enhance its block-diagonal structure.

Update Multipliers. Having $\mathbf{J}^{(t+1)}$, $\mathbf{Z}^{(t+1)}$, $\mathbf{E}^{(t+1)}$, and $\mathbf{H}^{(t+1)}$ fixed, we compute the Lagrange multipliers by performing a gradient ascent with the step size of $\mu^{(t)}$ as

$$\begin{aligned} \mathbf{Y}_1^{(t+1)} &= \mathbf{Y}_1^{(t)} + \mu^{(t)}(\mathbf{S} - \mathbf{S}\mathbf{Z}^{(t+1)} - \mathbf{E}^{(t+1)}), \\ \mathbf{Y}_2^{(t+1)} &= \mathbf{Y}_2^{(t)} + \mu^{(t)}(\mathbf{Z}^{(t+1)} - \mathbf{J}^{(t+1)}). \end{aligned} \tag{17}$$

We eventually obtain a robust representation $\mathbf{Z}$ and an optimized partition matrix $\mathbf{H}$ when the procedure of solving Equation (7) terminates. Note that, the final partition could be generated either by running K-means on $\mathbf{H}$ or conducting spectral clustering on $\mathbf{Z}$. Generally, RSEC employs $\mathbf{H}$ by default. However, as will be shown in the experiment, these two manners achieve the same performance for the most cases. Algorithm 1 provides a detailed summarization for RSEC.

3.4 Convergence and Complexity

Convergence Analysis. Generally, it is hard to guarantee global convergence for solving optimization problem with more than two variables, such as Equation (7). However, as suggested by previous works (Lin et al. 2010, 2011), ALM with Alternating Direction Minimizing (ADM) strategy could solve the proposed RSEC effectively, since each subproblem in our optimization process has a closed-form solution. Moreover, empirical evidence on real-world datasets indicates that our algorithm has a stable convergence performance, which shows Algorithm 1 works well in practice.

Complexity Analysis. As shown by Algorithm 1, the computing cost of the proposed RSEC mainly includes 4 parts: (1) For updating $\mathbf{J}$, the SVD computation in Equation (11) would cost $O(n^3)$, which would be computationally expensive when n is large. However, it could be accelerated as $O(rn^2)$ by using the skinny SVD decomposition (Lin et al. 2011), where $r \ll n$ is the rank of the matrix $\mathbf{J}$. (2) Several matrix multiplications and a matrix inverse operation are involved in Equation (13), thus it needs $O((l+1)n^3)$, where l is the number of the multiplications. (3) $\mathbf{E}$ can be updated with a thresholding solution in Equation (15), whose complexity is $O(n^2)$. (4) To find the partition $\mathbf{H}$, an eigenvalue decomposition is employed, which takes $O(n^3)$. Thus, the total cost of our method is $O(t((r+1)n^2 + (l+2)n^3)))$, where t is the iteration number of Algorithm 1. Our approach has a similar time and space complexity compared with previous low-rank based methods (Yi et al. 2012; Zhou et al. 2015), which is relatively high for large-scale datasets. Fortunately, it could be reduced by some off-the-shelf fast LRR methods (Xiao et al. 2015; Oh et al. 2015), and be further accelerated by a learnable predicting scheme (Li et al. 2016b, 2017b, 2017c). However, in this article, we are more interested in improving robustness and effectiveness for ensemble clustering, and thus we leave the scalability in our future work.

4 A NON-CONVEX ALTERNATIVE

In this section, we first explain how a non-convex penalty function can benefit the rank minimization problem, and then provide a non-convex RSEC model, along with an effective solution via the MM-ALM method.

4.1 Preliminary

Before giving our non-convex RSEC model, here, we first introduce how to use the Minmax Concave Plus (MCP) (Zhang 2010) function to obtain the matrix MCP norm and γ-norm, which can be used as non-convex penalty for sparsity modeling and rank minimization, respectively.

Given a matrix $\mathbf{X} \in \mathbb{R}^{d \times n}$ and $\lambda > 0$, $\gamma > 1$, the matrix MCP norm is defined as

$$\mathcal{M}_{\lambda,\gamma}(\mathbf{X}) = \sum_{i,j} \phi_{\lambda,\gamma}(\mathbf{X}_{i,j}), \tag{18}$$

of which $\phi_{\lambda,\gamma}(\cdot)$ is given by

$$\phi_{\lambda,\gamma}(t) = \lambda \int_0^t \left(1 - \frac{u}{\gamma\lambda}\right)_+ du = \begin{cases} \gamma\lambda^2/2, & \text{if } |t| \geq \gamma\lambda \\ \lambda|t| - t^2/2\gamma, & \text{otherwise} \end{cases},$$

where $(z)_+ = \max(z, 0)$. The definition given by Equation (18) is also applicable to a vector, in which case $d = 1$. In the following, we set $\lambda = 1$ as the default, and denote $\mathcal{M}_\gamma(\mathbf{X}) = \mathcal{M}_{1,\gamma}(\mathbf{X})$ for conciseness. It is worth noting that, the matrix MCP norm offers a better way, which jointly enjoys sparsity, continuity and unbiasedness (Zhang 2010), to approximate the ℓ_0 norm than its convex proxies (e.g., ℓ_1 norm and $\ell_{2,1}$ norm). Based on MCP, the matrix γ-norm is defined by (Wang et al. 2013) as

$$\|\mathbf{X}\|_\gamma = \sum_{i=1}^{r} \int_0^{\sigma_i(\mathbf{X})} \left(1 - \frac{u}{r}\right)_+ du = \sum_{i=1}^{r} \phi_{1,\gamma}(\sigma_i(\mathbf{X})) = \mathcal{M}_\gamma(\sigma(\mathbf{X})), \tag{19}$$

where $\gamma > 1$, $r = \min(d, n)$, $\sigma(\mathbf{X}) = (\sigma_1(\mathbf{X}), \ldots, \sigma_r(\mathbf{X}))^{\mathrm{T}}$ denotes a function from $\mathcal{R}^{m \times n}$ to $\mathcal{R}_+^r$ and $\sigma_i(\mathbf{X})$ is the ith largest singular value of $\mathbf{X}$, $1 \leq i \leq r$. Considering the matrix rank is equivalent to $\|\sigma(\mathbf{X})\|_0$, γ-norm naturally inherits the advantage of MCP function, and thus provides an unbiased approximation to the matrix rank. $\mathcal{M}_\gamma(\mathbf{X})$ and $\|\mathbf{X}\|_\gamma$ are both non-convex w.r.t. $\mathbf{X}$.

4.2 Motivation and Formulation

Recall the RSEC model in Equation (7), we use nuclear norm to approximate the matrix rank and transform RSEC as a convex problem with respect to each variable. However, since the nuclear norm simply sums all the singular values of a matrix, it may over-penalize large singular values (Wang et al. 2013; Peng et al. 2015), resulting in a biased rank estimation. This problem can degrade the performance of RSEC on some imbalanced datasets. Therefore, motivated by the robustness of non-convex rank minimization, we propose to reformulate our RSEC model by using the MCP matrix norm and γ-norm. The objective function of RSEC in Equation (4) is remodeled as

$$\min_{\mathbf{H},\mathbf{Z},\mathbf{E}} \operatorname{tr}(\mathbf{H}^{\mathrm{T}}\mathbf{L}_z\mathbf{H}) + \lambda_1\|\mathbf{Z}\|_{\gamma_1} + \lambda_2\mathcal{M}_{\gamma_2}(\mathbf{E}) \quad \text{s.t. } \mathbf{H}^{\mathrm{T}}\mathbf{H} = \mathbf{I}, \mathbf{S} = \mathbf{S}\mathbf{Z} + \mathbf{E}, \tag{20}$$

with $\mathbf{L}_z$ being defined by Equation (5). In Equation (20), we employ $\|\mathbf{Z}\|_{\gamma_1}$ and $\mathcal{M}_{\gamma_2}(\mathbf{E})$ as the non-convex relaxation for rank($\mathbf{Z}$) and $\|\mathbf{E}\|_0$, respectively. According to the definition in Equation (18) and Equation (19), when $\gamma \to \infty$, we have $\mathcal{M}_\gamma(\cdot) \to \|\cdot\|_1$ and hence $\|\cdot\|_\gamma \to \|\cdot\|_*$; when $\gamma \to 1$, $\mathcal{M}_\gamma(\cdot)$ works as a hard thresholding operator correspond to the ℓ_0 norm, which may induce poor local convergence result. Thus, $\mathcal{M}_\gamma$ bridges the gap between the ℓ_1 and ℓ_0 norm, and we always keep $\gamma > 1$ to achieve a good relaxation to ℓ_0 penalty. As suggested by [Wang et al. 2013], both $\mathcal{M}_{\gamma_1}(\cdot)$ and $\|\cdot\|_{\gamma_2}$ are quite insensitive to γ_1 and γ_2 as long as they stay in small real values strictly greater than 1. We fix $\gamma_1 = \gamma_2 = 2$ in the experiment.

4.3 Optimization

The Majorization–Minimization (MM) method is a common way for solving the non-convex optimization problem (Wang et al. 2013; Li and Fu 2015). It proceeds by iteratively solving a convex problem that locally approximates the original non-convex problem in each step. In this section, we develop a MM-ALM method to solve the problem of Equation (20), consisting of two loops: the *outer loop* replaces our non-convex RSEC by its locally linear approximation (LLA) to form a weighter convex problem; while the *inner loop* performs an inexact ALM algorithm. Similar to Equation (8), we first introduce an auxiliary variable $\mathbf{J}$ and relax the $\mathbf{H}^{\mathrm{T}}\mathbf{H} = \mathbf{I}$ to facilitate the optimizing process, and then our non-convex RSEC can be formulated as

$$\min_{\mathbf{H},\mathbf{Z},\mathbf{E},\mathbf{J}} \operatorname{tr}(\mathbf{H}^{\mathrm{T}}\mathbf{L}_z\mathbf{H}) + \lambda_1\|\mathbf{J}\|_{\gamma_1} + \lambda_2\mathcal{M}_{\gamma_2}(\mathbf{E}) \quad \text{s.t. } \mathbf{S} = \mathbf{S}\mathbf{Z} + \mathbf{E}, \mathbf{Z} = \mathbf{J}. \tag{21}$$

In the *outer loop*, since Equation (21) is concave with respect to $(\sigma(\mathbf{J}), |\mathbf{E}|)$, we approximate $\|\mathbf{J}\|_{\gamma_1}$ and $\mathcal{M}_{\gamma_2}(\mathbf{E})$ by its LLA at $(\sigma(\mathbf{J})^{old}, |\mathbf{E}|^{old})$, and transform Equation (21) into a convex problem:

$$\begin{aligned}&\min_{\mathbf{H},\mathbf{Z},\mathbf{E},\mathbf{J}} \operatorname{tr}(\mathbf{H}^{\mathrm{T}}\mathbf{L}_z\mathbf{H}) + \lambda_1\mathcal{Q}_{\gamma_1}(\sigma(\mathbf{J})|\sigma(\mathbf{J})^{old}) + \lambda_2\mathcal{Q}_{\gamma_2}(\mathbf{E}|\mathbf{E}^{old})\\ &\text{s.t. } \mathbf{S} = \mathbf{S}\mathbf{Z} + \mathbf{E}, \mathbf{Z} = \mathbf{J},\end{aligned} \tag{22}$$

where

$$\mathcal{Q}_\gamma(\mathbf{A}|\mathbf{A}^{old}) = \mathcal{M}_\gamma(\mathbf{A}) + \sum_{i,j}(1 - |\mathbf{A}_{ij}^{old}|/\gamma)_+(|\mathbf{A}_{ij}| - |\mathbf{A}_{ij}^{old}|),$$

is the LLA of $M_\gamma(\mathbf{A})$ given $\mathbf{A}^{old}$.

In the *inner loop*, similar to Section 3.3, the inexact ALM algorithm is used to solve Equation (22) by dividing it into several subproblems, which updates one variable while keeping others fixed. The augmented Lagrange functions of Equation (22) is written as

$$\begin{aligned}\mathcal{L} = {} & \operatorname{tr}(\mathbf{H}^{\mathrm{T}}\mathbf{L}_z\mathbf{H}) + \lambda_1\mathcal{Q}_{\gamma_1}(\sigma(\mathbf{J})|\sigma(\mathbf{J})^{old}) + \lambda_2\mathcal{Q}_{\gamma_2}(\mathbf{E}|\mathbf{E}^{old})\\ & + \langle \mathbf{Y}_1, \mathbf{S} - \mathbf{S}\mathbf{Z} - \mathbf{E}\rangle + \langle \mathbf{Y}_2, \mathbf{Z} - \mathbf{J}\rangle + \frac{\mu}{2}(\|\mathbf{S} - \mathbf{S}\mathbf{Z} - \mathbf{E}\|_{\mathrm{F}}^2 + \|\mathbf{Z} - \mathbf{J}\|_{\mathrm{F}}^2),\end{aligned} \tag{23}$$

where $\mathbf{Y}_1, \mathbf{Y}_2$ are two Lagrange multipliers, and $\mu > 0$ is a penalty parameter. One may note that, since the subproblems of $\mathbf{Z}$ and $\mathbf{H}$ are similar to that of Equations (12) and (16), their solutions keep unchanged. Thus, here, we only give the solutions to the subproblems of $\mathbf{J}$ and $\mathbf{E}$. For the $(t+1)$-th iteration of the inner loop, based on the theorem provided by Wang et al. (2013), we can obtain $\mathbf{J}^{(t+1)}$ and $\mathbf{E}^{(t+1)}$ by

$$\begin{aligned}\mathbf{J}^{(t+1)} &= \operatorname*{argmin}_{\mathbf{J}} \frac{1}{2}\left\|\mathbf{J} - \left(\mathbf{Z}^{(t)} + \frac{\mathbf{Y}_2^{(t)}}{\mu^{(t)}}\right)\right\|_{\mathrm{F}}^2 + \frac{\lambda_1}{\mu^{(t)}}\mathcal{Q}_{\gamma_1}(\sigma(\mathbf{J}|\sigma(\mathbf{J})^{old})\\ &= \Theta_{\frac{\lambda_1}{\mu^{(t)}},\Lambda}\left(\mathbf{Z}^{(t)} + \frac{\mathbf{Y}_2^{(t)}}{\mu^{(t)}}\right),\end{aligned} \tag{24}$$

$$\begin{aligned}\mathbf{E}^{(t+1)} &= \operatorname*{argmin}_{\mathbf{E}} \frac{1}{2}\left\|\mathbf{E} - \left(\mathbf{S} - \mathbf{S}\mathbf{Z}^{(t+1)} + \frac{\mathbf{Y}_1^{(t)}}{\mu^{(t)}}\right)\right\|_{\mathrm{F}}^2 + \frac{\lambda_2}{\mu}\mathcal{Q}_{\gamma_2}(\mathbf{E}|\mathbf{E}^{old})\\ &= \mathcal{D}_{\frac{\lambda_2}{\mu^{(t)}},\mathbf{W}}\left(\mathbf{S} - \mathbf{S}\mathbf{Z}^{(t+1)} + \frac{\mathbf{Y}_1^{(t)}}{\mu^{(t)}}\right),\end{aligned} \tag{25}$$

where $\Lambda = (\mathbf{I} - \Sigma_{\mathbf{J}^{old}}/\gamma_1)_+$ and $\mathbf{W} = (\mathbf{1}_n\mathbf{1}_n^{\mathrm{T}} - |\mathbf{E}^{old}|)_+$ are two elementwise non-negative matrices obtained by the *outer loop*. $\Theta_{\tau,\Lambda}$ and $\mathcal{D}_{\tau,W}$ are defined as the generalized singular value shrinkage operator and generalized shrinkage operator (Wang et al. 2013) as the following:

$$\begin{aligned}\Theta_{\tau,\Lambda}(\mathbf{X}) &= \mathbf{U}_{\mathbf{X}}\mathcal{D}_{\tau,\Lambda}(\Sigma_{\mathbf{X}})\mathbf{V}_{\mathbf{X}}^{\mathrm{T}},\\ [\mathcal{D}_{\tau,\mathbf{W}}(\mathbf{A})]_{ij} &= \operatorname{sgn}(\mathbf{A}_{ij})(|\mathbf{A}_{ij}| - \tau\mathbf{W}_{ij})_+.\end{aligned} \tag{26}$$

ALGORITHM 2: Non-Convex RSEC by MM-ALM

```
Input: a set of basic partitions $\Pi = \{\pi_1, \pi_2, \ldots, \pi_m\}$, cluster number $K$, two parameters $\lambda_1, \lambda_2 > 0$.
Initial: $\mathbf{J}^0 = \mathbf{Z}^0 = \mathbf{E}^0 = \mathbf{0} \in \mathbb{R}^{n\times n}$, $\mathbf{H}^0 = \mathbf{0} \in \mathbb{R}^{n\times K}$, $\rho = 1.3$, $\mu_{\max} = 10^{10}$ and $\epsilon = 10^{-7}$.
    Derive S from $\Pi$ via Equation (1) and normalize it by $\mathbf{S} = \mathbf{S}/m$;
    while not converged do
        Initialize $\mathbf{Y}_1^0 = \mathbf{Y}_2^0 = \mathbf{0} \in \mathbb{R}^{n\times n}$, $\mu_0 = 1.5/\|\mathbf{S}\|_2$, and $t = 0$;
        $\Lambda = (\mathbf{I} - \Sigma_{\mathbf{J}^{old}}/\gamma_1)_+$;
        $\mathbf{W} = (\mathbf{1}_n\mathbf{1}_n^{\mathrm{T}} - |\mathbf{E}^{old}|)_+$;
        while not converged do
            Update $\mathbf{J}^{(t+1)}$ via Equation (24);
            Update $\mathbf{Z}^{(t+1)}$ via Equation (13);
            Update $\mathbf{E}^{(t+1)}$ via Equation (25);
            Compute $\mathbf{D}_z$ and $\mathbf{L}_z$ based on $\mathbf{Z}^{(t+1)}$ and $\mathbf{H}^{(t)}$ via Equation (5–6);
            Set $\mathbf{H}^{(t+1)}$ as the smallest $K$ eigenvectors of $\mathbf{L}_z$;
            Update the Lagrangian multipliers $\mathbf{Y}_1^{(t+1)}$ and $\mathbf{Y}_2^{(t+1)}$ via Equation (17);
            Check the convergence condition
                $\|\mathbf{S} - \mathbf{S}\mathbf{Z}^{(t+1)} - \mathbf{E}^{(t+1)}\|_\infty < \epsilon$ and $\|\mathbf{J}^{(t+1)} - \mathbf{Z}^{(t+1)}\|_\infty < \epsilon$;
            $\mu^{(t+1)} = \min(\rho\mu^{(t)}, \mu_{\max})$, $t = t + 1$;
        end while
        $\mathbf{J}^{old} \leftarrow \mathbf{J}^{(t+1)}$, $\mathbf{E}^{old} \leftarrow \mathbf{E}^{(t+1)}$;
    end while
    Run $K$-means on H or spectral clustering on Z to obtain the final partition $\pi$.
Output: the final partition $\pi$.
```

4.4 Convergence and Complexity

Convergence Analysis. The Algorithm 2 has two parts including *outer loop* and *inner loop*. Let $f(\mathbf{H}, \mathbf{Z}, \mathbf{E}, \mathbf{J})$ be the objective function in Equation (21). For the *outer loop*, by keeping the other variables fixed, $f(\mathbf{J}, \mathbf{E})$ obeys the following property (Wang et al. 2013):

$$\begin{aligned} f(\mathbf{J}, \mathbf{E}) &\leq \lambda_1 Q_{\gamma_1}(\sigma(\mathbf{J})|\sigma(\mathbf{J})^{old}) + \lambda_2 Q_{\gamma_2}(\mathbf{E}|\mathbf{E}^{old}) \\ &\leq \lambda_1 Q_{\gamma_1}(\sigma(\mathbf{J})^{old}|\sigma(\mathbf{J})^{old}) + \lambda_2 Q_{\gamma_2}(\mathbf{E}^{old}|\mathbf{E}^{old}) \\ &= f(\mathbf{J}^{old}, \mathbf{E}^{old}), \end{aligned}$$

which shows Algorithm 2 is non-increasing, and hence could find a local optimal solution. For the *inner loop*, it has the similar convergence property to the Algorithm 1, which has been provided in Section 3.4 before. The convergence behavior of Algorithm 2 is also demonstrated by the empirical evidence in the experiment.

Complexity Analysis. The MM-ALM method inevitably burdens the computing cost of our non-convex RSEC model, since it may conduct the LLA for $(\sigma(\mathbf{J}), |\mathbf{E}|)$ several times. To alleviate such defect, we adopt a one-step LLA strategy, which only proceeds the *outer loop* once. As suggested by Wang et al. (2013), this has little effect to the final performance. The *inner loop* in Algorithm 2 is also solved by the ALM method with a similar procedure to the Algorithm 1, thus these two algorithms actually have the same time complexity.

5 EXPERIMENT

In this section, we report the clustering results of our convex model in Algorithm 1 (denoted as RSEC), as well as its non-convex relaxation provided in Algorithm 2 (denoted as NRSEC). We compare our algorithms with several state-of-the-art ensemble clustering (EC) methods on

Table 1. Dataset Details

Dataset	#Instance	#Feature	#Class	#minClass	#maxClass	CV	Density	Source
breast_w	699	9	2	241	458	0.4390	0.9975	UCI
BreastTissue	106	9	6	14	22	0.1849	0.9927	UCI
Glass	214	9	6	9	76	0.8339	0.7965	UCI
iris	150	4	3	50	50	0.0000	1.0000	UCI
ionosphere	351	35	2	126	225	0.3989	0.6002	UCI
pendigits	10992	16	10	1055	1144	0.0422	0.8717	UCI
wine	178	13	3	48	71	0.1939	1.0000	UCI
fbis	2463	2000	17	38	506	0.9614	0.0799	CLUTO
k1b	2340	21839	6	60	1389	1.3162	0.0068	CLUTO
re0	1504	2886	13	11	608	1.5023	0.0179	CLUTO
tr12	313	5804	8	9	93	0.6381	0.0471	CLUTO
tr11	414	6429	9	6	132	0.8817	0.0438	CLUTO
tr23	204	5832	6	6	91	0.9345	0.0661	CLUTO
wap	1560	8460	20	5	341	1.0403	0.0167	CLUTO
COIL20	1440	1024	20	72	72	0.0000	0.6561	others
ImageNet	7341	4096	5	910	2126	0.3072	0.1623	others
MNIST4K	4000	784	10	359	454	0.0818	0.1854	others
USPS	9298	256	10	708	1553	0.2903	0.2456	others

18 real-world datasets by using two widely-used validation criteria. Moreover, extensive discussions are given to the factors related to the clustering performance of our methods.

5.1 Experimental Setup

Datasets Details. Eighteen real-world datasets are used in the experiment for evaluating the proposed method. To achieve a comprehensive evaluation, we employ datasets of various types from different sources. Specifically, we select seven datasets from the UCI Machine Learning Repository[1]; seven text-type datasets from CLUTO[2]; and four image-type datasets from other sources, such as *COIL20* (Nene et al. 1996), *ImageNet*,[3] *MNIST4K*,[4] and *USPS*.[4] UCI is the most widely-used dataset collection for the machine learning community, and CLUTO provides a testbed for the document clustering task. The *MNIST4K* provided by Cai et al. (2011) is a subset of the *MNIST* dataset (LeCun et al. 1998). More details are shown in Table 1, where "#minClass" and "#maxClass" denote as the instance number of the smallest cluster and the largest one, respectively. Besides, CV indicates the Coefficient of Variation statistic which characterizes the imbalance degree of clusters, and Density is used to measure the feature sparseness.

Compared Methods. We compare our approaches (RSEC and NRSEC) with six state-of-the-art ensemble clustering methods, including Graph-based Consensus Clustering (GCC) (Strehl and Ghosh 2003), Hierarchical Clustering on Co-association matrix (HCC) (Fred and Jain 2005), Ensemble Clustering by Matrix Completion (ECMC) (Yi et al. 2012), *K*-means based Consensus Clustering (KCC) (Wu et al. 2015), Spectral Ensemble Clustering (SEC) (Liu et al. 2015a, 2017), and Robust Clustering Ensemble (RCE) (Zhou et al. 2015). These six methods are selected in terms of

[1] https://archive.ics.uci.edu/ml/datasets.html.
[2] http://glaros.dtc.umn.edu/gkhome/cluto/cluto/download.
[3] http://www.cs.dartmouth.edu/~chenfang/.
[4] http://www.cad.zju.edu.cn/home/dengcai/Data/MLData.html.

(1) classic and representative methods (GCC and HCC); (2) recent effective work (KCC and SEC); and (3) robust EC methods based on low-rank constraint (ECMC and RCE). Besides, two traditional clustering algorithms, i.e., K-means and spectral clustering, are also employed as baseline methods in the experiment.

Validation Criteria. We use two well-known clustering validation criteria for the quantitative analysis, which are *Average Clustering Accuracy* (*ACC*) and *Normalized Mutual Information* (*NMI*). *ACC* and *NMI* are both positive measurements ranged from 0 to 1, where a higher value indicates better performance. A brief introduction to these two metrics are given as follows.

ACC is the fraction of resulted labels given by a clustering method that match with ground-truth labels (Yan et al. 2009). Given a dataset $\mathcal{X}$ of n instances with K clusters, *ACC* is formulated as

$$\max_{g} \frac{1}{n} \sum_{j=1}^{n} \delta(y_j, g(\pi(x_j))), \tag{27}$$

where $x_j \in \mathcal{X}, y_j \in [1, K]$ is the ground-truth cluster label of x_j, $1 \leq j \leq n$, π is the clustering result that maps $\mathcal{X}$ into a label set $\{\pi(x_1), \dots, \pi(x_n)\}$, $\pi(x_j) \in [1, K]$, and $g(\cdot)$ works as a permutation operation. Since clustering is an unsupervised algorithm, thus we need to search $g(\cdot)$ from all the permutations of K cluster labels to maximize the final *ACC*.

NMI computes the mutual information entropy between cluster labels and the ground-truth (Shao et al. 2015), which is defined as

$$NMI = \frac{\sum_h \sum_l n_{h,l} \log(\frac{n n_{h,l}}{n_h n_l})}{\sqrt{(\sum_h n_h \log \frac{n_h}{n})(\sum_l n_l \log \frac{n_l}{n})}}, \tag{28}$$

where n_h and n_l denote the number of instances in cluster C_h found by a partition result and C_l given by the ground-truth, respectively, and $n_{h,l}$ is the number of instances in both C_h and C_l. *NMI* will be inclined to 0 if the data are randomly partitioned. More details about clustering criteria could be referred by Wu et al. (2009).

Clustering Tools. Following Wu et al. (2015), we generate $r = 100$ BPs (denoted as Π) with the RPS strategy (Fred and Jain 2005; Wu et al. 2015) on each dataset, and use Π as the default input for all the EC methods. In details, each BP of Π is obtained by running K-means with the cluster number varying from K to $\sqrt{n}$, where K is the true cluster number and n is the size of dataset.

We performed each EC method by running the code provided by the authors, and tuned the parameters as suggested in their papers. For traditional methods, we directly used the MATLAB function K-means for the K-means algorithm, and implemented spectral clustering according to (Ng et al. 2001). The cluster number was set to be K for testing all the methods. Moreover, we tested KCC, SEC, K-means and spectral clustering twenty times and reported the average result, as they all include random initialization. For ECMC and RCE, their model has two parts, i.e., first, recovering a low-rank co-association matrix and, second, performing spectral clustering on the recovered matrix for the final partition result. Thus, we also ran the second part of these two methods 20 times for a fair comparison. Similarly, for our RSEC (NRSEC), we repeated the last step in Algorithm 1(2) twenty times. The *standard deviation* (*std*) of clustering results were reported for all the methods except for HCC, as HCC is a deterministic algorithm. We set $\lambda_1 = 0.1$, $\lambda_2 = 0.01$ for RSEC and $\lambda_1 = 1$, $\lambda_2 = 0.01$ for NRSEC as the default setting.

All the experiments were conducted by MATLAB on a 64-bit Ubuntu 14.04 platform with two Intel Core i7 3.4GHz CPUs and 32GB RAM. Note that, since RCE suffers from a high space complexity of $O(rn^2)$, we cannot run it on some datasets due to the memory limitation, labeled by "N/A" in Tables 2 and 3.

Table 2. Clustering Performance on 18 Real-world Datasets by *ACC* (%)

Datasets	Our methods		Ensemble clustering methods						Baseline methods	
	RSEC	NRSEC	GCC	HCC	ECMC	KCC	SEC	RCE	*K*-means	spectral
breast_w	**97.14±0.00**	**97.14±0.00**	90.56±0.00	94.99	95.85±0.00	66.35±8.83	95.81±0.00	**97.14±0.00**	95.85±0.00	96.28±0.00
BreastTissue	37.00±1.30	**43.40±0.00**	37.74±0.00	40.57	*42.45±0.00*	41.32±1.88	39.62±0.00	41.23±0.44	33.96±0.00	34.43±1.28
Glass	47.20±0.00	**54.16±0.55**	51.47±0.78	52.34	52.14±2.76	53.24±3.57	*54.08±2.99*	52.34±0.00	52.10±3.12	51.79±1.11
iris	**97.33±0.00**	**97.33±0.00**	**97.33±0.00**	89.33	88.67±0.00	88.92±2.47	96.00±0.00	90.00±0.00	89.27±0.20	88.67±0.00
ionosphere	67.52±0.00	**68.38±0.00**	67.18±0.00	**68.38**	56.01±1.16	63.82±0.00	63.53±0.00	67.52±0.00	71.23±0.00	70.37±0.00
pendigits	**86.44±0.00**	**86.44±0.00**	73.06±0.00	74.24	78.30±0.00	63.97±5.12	74.61±4.20	N/A	74.51±4.76	70.75±3.05
wine	52.25±0.00	**53.76±2.38**	51.69±0.00	50.00	*52.92±2.01*	50.49±0.33	51.12±0.00	50.00±0.00	50.00±0.00	51.12±0.00
fbis	*54.39±2.29*	**58.15±2.15**	45.49±0.02	54.28	45.98±3.65	48.91±2.81	48.37±1.79	53.47±0.00	28.62±3.00	19.33±0.20
k1b	77.62±1.36	*80.09±1.04*	51.06±0.01	**82.26**	68.80±8.18	49.17±5.78	49.38±1.66	74.79±0.00	72.59±8.42	58.91±1.53
re0	*42.08±1.96*	**43.93±0.04**	35.51±0.00	39.56	39.17±2.72	36.27±2.60	34.91±1.46	35.44±0.00	37.20±3.16	29.11±2.22
tr11	63.38±2.48	63.35±1.80	**65.70±0.00**	63.29	50.19±2.86	60.53±2.83	53.57±2.50	*64.49±0.00*	31.27±1.59	31.63±0.61
tr12	**69.33±0.36**	**69.33±0.00**	55.59±0.00	45.05	47.60±5.10	56.03±3.90	58.85±2.07	50.16±0.00	28.20±1.32	28.85±0.30
tr23	*41.18±0.00*	**42.67±2.97**	*41.18±0.00*	37.75	38.73±0.00	41.32±3.75	39.46±1.61	40.20±0.00	40.74±1.98	38.56±1.27
wap	*48.96±1.55*	**51.86±0.61**	42.04±0.13	41.47	42.42±2.41	42.08±2.99	38.76±2.67	40.04±1.04	37.02±2.97	38.73±2.40
COIL20	**64.96±2.36**	*62.27±3.09*	53.68±0.00	30.49	51.72±2.92	61.18±3.84	61.03±3.14	40.83±0.29	63.09±4.41	64.04±3.27
ImageNet	*79.62±1.22*	**83.72±0.00**	72.40±0.01	79.70	79.92±5.06	74.40±4.59	78.86±5.78	N/A	73.48±2.98	76.04±0.02
MNIST4K	*61.50±0.66*	61.43±1.22	**66.08±0.00**	58.20	51.55±2.94	55.75±4.05	57.04±2.91	58.96±0.03	54.31±2.29	52.88±1.83
USPS	68.13±2.51	**73.15±1.30**	59.59±0.01	*69.49*	59.30±3.91	57.52±5.66	68.61±3.75	N/A	67.15±0.42	68.76±3.62
score	*17.17*	**17.79**	15.80	15.93	15.55	15.34	15.84	13.29	14.80	14.30

The top EC performer is highlighted by bold font with red color, while the second best by italic with blue.

5.2 Clustering Performance

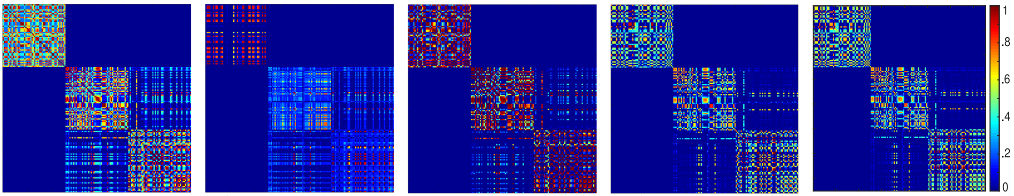
Overall. Tables 2 and 3 summarize the clustering performance of our approaches (RSEC and NRSEC) and compared methods in terms of *ACC* and *NMI*, respectively. As can be seen, our approaches outperform all the other EC methods in most cases. In detail, our approach is the top performer on 15 out of 18 datasets and the second best on two out of the remainder by *ACC* in Table 2. For another metric, we achieve the highest *NMI* on 16 out of 18 datasets and perform second best on the others in Table 3. Moreover, we get the first position by both *ACC* and *NMI* on 14 datasets. The superiority of our approach to the other methods can be clearly observed on *pendigits* and *tr12*, where we improve clustering performance over 8% on *pendigits* and 10% on *tr12*. To further evaluate the performance of all the methods, we compute a measurement score by following (Wu et al. 2015): $\text{score}(A_i) = \sum_j \frac{R(A_i, D_j)}{\max_i R(A_i, D_j)}$, where $R(A_i, D_j)$ indicates the *ACC* or *NMI* value of A_i method on the D_j dataset. This score gives an overall evaluation on all the datasets. As shown by Table 2(3), our approach outperforms other methods significantly.

Compared to Baseline Methods. From Tables 2 and 3, we may have several important observations. First, EC methods substantially boost the clustering performance over the baseline methods in general, which shows ensemble clustering as an effective clustering manner. Second, we observe that *K*-means and spectral clustering can exceed EC methods on some datasets, such as *ionosphere* and *COIL20*. We conjecture this is mainly due to the limitation of RPS strategy. RPS ensures the diversity of multiple BPs, while it may undermine the cluster structure of the dataset on some cases. Finally, one may note that, all the EC methods perform unsatisfactorily on *wine* and *ionosphere*. Nevertheless, the performance could be improved by using an alternative BP

Table 3. Clustering Performance on 18 Real-world Datasets by *NMI* (%)

Datasets	Our methods		Ensemble clustering methods						Baseline methods	
	RSEC	NRSEC	GCC	HCC	ECMC	KCC	SEC	RCE	*K*-means	spectral
breast_w	**82.38±0.00**	**82.38±0.00**	61.44±0.00	69.99	73.61±0.00	28.16±13.72	73.61±0.00	80.64±0.00	73.61±0.00	75.63±0.00
BreastTissue	25.67±0.88	**32.38±0.00**	27.44±0.00	30.46	29.12±0.00	*30.89±1.45*	29.08±0.00	28.98±0.20	36.52±0.00	35.11±2.06
Glass	*43.94±0.61*	**48.24±0.00**	41.80±0.54	40.83	37.72±3.88	39.56±1.77	40.95±0.83	35.42±0.00	39.69±3.36	43.37±2.88
iris	**90.11±0.00**	**90.11±0.00**	**90.11±0.00**	79.08	74.19±0.00	78.30±4.50	87.05±0.00	78.69±0.00	75.66±0.49	74.19±0.00
ionosphere	8.01 ±0.00	*8.57 ±0.00*	7.19 ±0.00	*8.57*	7.88 ±3.92	**8.81 ±0.00**	7.70 ±0.00	7.35±0.00	13.49±0.00	12.64±0.00
pendigits	**82.64±0.00**	**82.64±0.00**	70.58±0.00	77.29	74.19±0.00	68.36±3.43	73.87±1.67	N/A	68.93±0.56	65.94±0.76
wine	**28.89±0.00**	**28.89±0.00**	16.32±0.00	18.67	21.98±4.22	16.51±0.39	17.60±0.00	16.34±0.00	13.38±0.00	13.36±0.00
fbis	55.95±0.79	**57.07±0.11**	54.23±0.00	*56.21*	51.79±2.46	55.11±1.21	54.87±1.09	54.17±0.00	24.75±3.81	5.02 ±0.29
k1b	55.20±2.35	*60.40±0.23*	56.29±0.01	**63.18**	52.47±10.2	47.58±8.02	50.57±2.08	57.32±0.02	44.46±13.37	48.71±0.43
re0	*39.50±0.94*	**41.66±0.03**	38.02±0.03	37.71	35.65±3.81	39.04±1.16	38.24±1.17	32.64±0.01	23.24±1.24	28.39±0.79
tr11	*65.65±1.29*	**67.06±0.91**	65.33±0.00	65.58	54.21±3.54	64.35±1.81	59.73±2.28	64.80±0.00	10.06±1.32	7.39 ±0.59
tr12	*61.31±0.49*	**62.62±0.27**	49.03±0.00	41.89	34.88±4.79	51.21±3.32	51.39±2.07	45.72±0.00	8.90±1.68	7.35±0.67
tr23	32.54±0.00	**34.67±1.07**	*33.43±0.00*	28.97	29.50±0.00	31.91±3.24	29.24±1.29	29.44±0.00	12.98±2.31	7.19 ±1.17
wap	*57.29±1.09*	**58.46±0.39**	49.81±0.10	56.59	43.37±4.13	56.40±1.32	54.10±1.20	54.20±0.02	42.55±2.24	49.10±1.31
COIL20	**77.70±1.38**	*76.48±1.66*	71.41±0.00	61.24	69.14±1.64	75.63±1.40	75.00±1.22	62.01±0.45	76.70±1.82	76.98±1.59
ImageNet	*59.61±1.22*	**61.72±0.01**	49.35±0.01	54.19	56.33±2.16	52.50±3.89	57.84±2.62	N/A	45.22±2.37	45.96±0.03
MINIST4K	**65.25±0.66**	*64.92±0.24*	59.98±0.00	62.27	51.39±2.56	56.47±3.25	56.52±1.76	60.70±0.01	45.37±1.34	45.56±0.85
USPS	**76.77±0.45**	*75.49±0.41*	62.84±0.01	73.92	64.52±1.86	63.14±3.66	65.87±1.44	N/A	61.35±0.27	64.83±1.74
score	*16.85*	**17.44**	15.10	15.55	14.52	14.76	15.37	12.33	12.16	11.89

The top EC performer is highlighted by bold font with red color, while the second best by italic with blue.



(a) Input **S** of *iris* (b) ECMC (0.7419) (c) RCE (0.7869) (d) RSEC (0.9011) (e) NRSEC (0.9011)

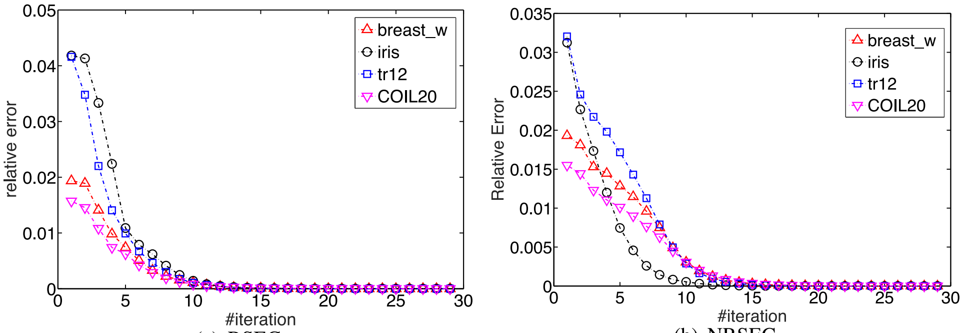
Fig. 2. An illustration of clustering structure in the learned co-association matrices from different methods on *iris*.

generation strategy, named as Random Feature Selection (RFS) (Wu et al. 2015). As will be shown in the next, our approach performs best with RFS on these two datasets.

Compared to Low-Rank Based Methods. The proposed method outperforms the other two low-rank based EC methods, i.e., ECMC and RCE, in most scenarios of Tables 2 and 3. This is mainly because our method can better uncover the cluster structure of a co-association matrix, which is demonstrated by Figure 2. Given the co-association matrix S of *iris*, ECMC alleviates the existing noises in S but undermines its cluster structure to some extend; on the other side, RCE holds the structure, yet still suffers from some gross noises. Different from these two methods, our approach learns a LRR for S, and employs the consensus partition to iteratively enhance the cluster structure of the learned representation. Thus, the proposed RSEC and NRSEC can reduce the noises in S meanwhile without weakening its structure. As shown by Figure (d) and (e), both RSEC and NRSEC obtain a clear $K = 3$ block-diagonal co-association matrix.

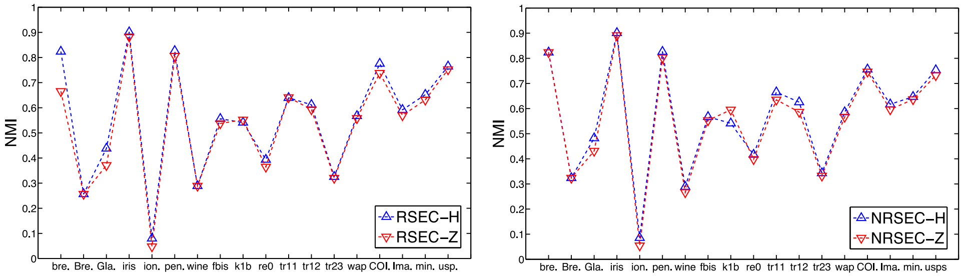
Table 4. Computational Time (*sec.*) of the Proposed RSEC and NRSEC on 18 Datasets

Datasets	*breast_w*	*BreastTissue*	*Glass*	*iris*	*ionosphere*	*pendigits*	*wine*	*fbis*	*k1b*
RSEC	18.33	0.29	1.51	0.67	2.37	31269.80	1.09	602.77	521.66
NRSEC	15.54	0.25	1.40	0.61	3.67	45815.79	0.92	652.40	798.35
Datasets	*re0*	*tr11*	*tr23*	*tr12*	*wap*	*COIL20*	*ImageNet*	*MINIST4K*	*USPS*
RSEC	93.62	6.76	3.52	1.83	136.83	67.15	10822.23	1949.05	21734.01
NRSEC	113.55	5.38	3.24	1.28	139.66	95.61	13565.59	2632.54	28079.31



(a) RSEC (b) NRSEC

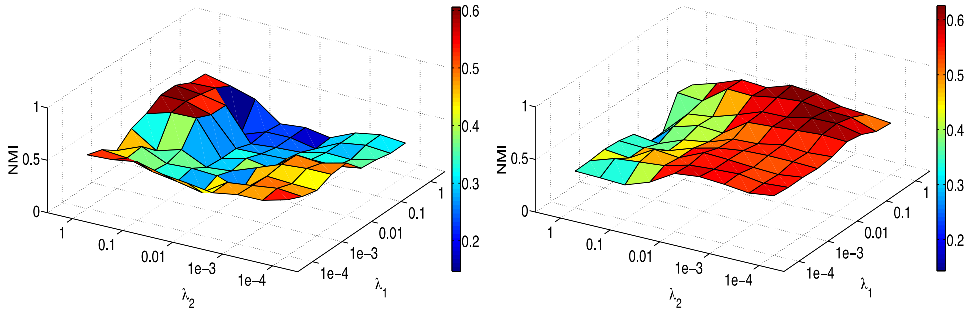
Fig. 3. Convergence curves of the Proposed RSEC and RSECN on four datasets.



(a) RSEC (b) NRSEC

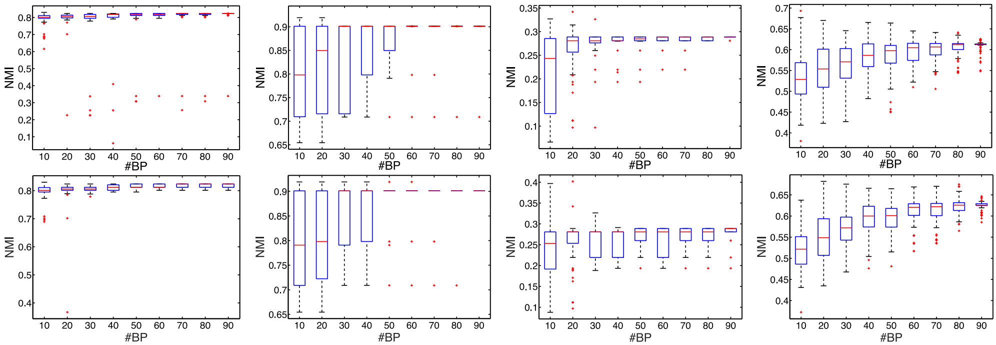
Fig. 4. A comparison of *NMI* between two different manners for obtaining the final clustering result on all the datasets, where RSEC-H (NRSEC-H) indicates running *K*-means on the optimal consensus partition **H**, and RSEC-Z (NRSEC-Z) denotes conducting spectral clustering on the learned low-rank representation **Z**.

RSEC vs. NRSEC. In this article, we propose a convex formulation (RSEC) and a non-convex relaxation (NRSEC) to our model, respectively. The main difference between these two solutions lies in which way we approximate the matrix rank, where RSEC employs the nuclear norm while NRSEC uses the matrix γ-norm. By comparing the results of RSEC and NRSEC in Table 2(3), we observe that NRSEC generally has a better performance than RSEC on some imbalance datasets, whose CV values are greater than 0.8 (see Table 1). Considering the co-association matrix of a imbalance dataset ideally has diagonal blocks with highly different sizes, our RSEC may overlook some small clusters duo to the nuclear norm is biased to large singular values. This can degrade the performance of RSEC on some cases, such as *Glass*, *fbis,* and *k1b*. Fortunately, NRSEC is able to address this issue by using the unbiased rank estimation, and thus effectively improves the robustness of our approach to the imbalance datasets. Moreover, we also compare NRSEC and



(a) RSEC (b) NRSEC

Fig. 5. Parameter analysis of λ_1 and λ_2 on *tr12* for RSEC and NRSEC.



(a) *breast_w* (b) *iris* (c) *wine* (d) *tr12*

Fig. 6. Impact of BPs number to the proposed RSEC (first row) and NRSEC (second row).

RSEC in terms of running time on all the datasets in Table 4, where NRSEC generally exhibits a similar computation complexity to RSEC. Thus, we may conclude that NRSEC boosts RSEC without degrading the computation efficiency.

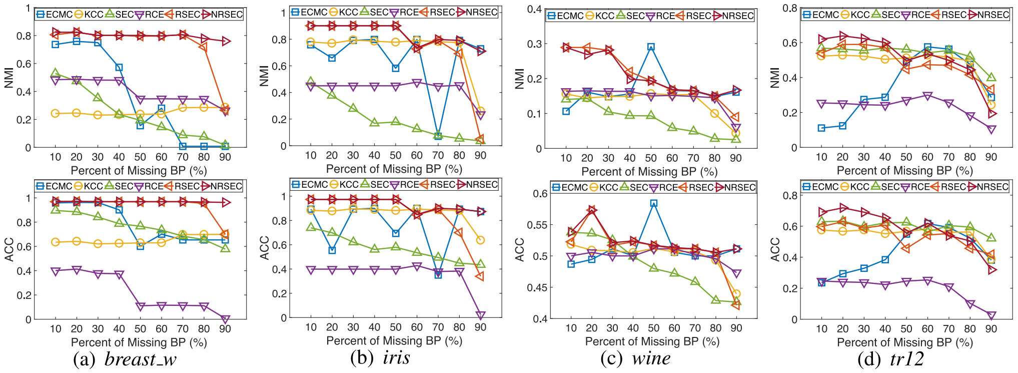
To sum up, Tables 2 and 3 demonstrate the effectiveness of our approach compared with the sate-of-the-art EC methods. Moreover, the proposed RSEC and NRSEC show promising clustering performance on the text-type and image-type datasets, which implies our approach can be further applied for many specific tasks, such as document clustering and image segmentation.

5.3 Discussion on RSEC

Here, we discuss the properties of RSEC and NRSEC from the following three aspects.

Convergence. We calculate the relative error as $\|\mathbf{S} - \mathbf{SZ} - \mathbf{E})\|_F / \|\mathbf{S}\|_F$ to evaluate the convergence property of our approach. As illustrated by Figure 3, the proposed RSEC and NRSEC both converges within 15 iterations on four datasets, which shows we have a fast and stable convergence performance.

Different Clustering Manners. Recall the Algorithm 1(2), RSEC (NRSEC) can obtain the final clustering result either by running *K*-means on the optimal consensus partition **H** (RSEC-H or NRSEC-H) or conducting spectral clustering on the learned representation **Z** (RSEC-Z or NRSEC-Z). Figure 4 shows the difference between these two manners by *NMI* for RSEC and NRSEC, respectively. As expected, RSEC-H and RSEC-Z have the similar performance in most cases, where the



(a) *breast_w* (b) *iris* (c) *wine* (d) *tr12*

Fig. 7. Impact of incomplete BPs to ensemble clustering methods. We test ECMC (Yi et al. 2012), KCC (Wu et al. 2015), SEC (Liu et al. 2015a), RCE (Zhou et al. 2015), and the proposed RSEC and NRSEC on four datasets by ranging missing ratio from 10% to 90% (*NMI* in the first row and *ACC* in the second).

same situation appears at NRSEC-H and NRSEC-Z. This indicates the efficacy of our optimization framework on jointly learning **H** and **Z**.

Parameter Analysis. In our objective function of Equation (4), λ_1 and λ_2 are two parameters to balance the rankness of the learned representation **Z** and sparseness of the noise matrix **E**, respectively. We take *tr12* as an example to show the effect of these two parameters to the clustering performance of our approach. Figure 5 depicts the *NMI* variation of RSEC and NRSEC, where λ_1 and λ_2 are both ranged from 10^{-4} to 1. As can be seen, RSEC generally keeps stable at the region of $\lambda_1 \in [0.01, 1]$ and $\lambda_2 \in [0.1, 1]$; NRSEC tends to be steady with the area of $\lambda_1 \in [10^{-4}, 1]$ and $\lambda_2 \in [10^{-4}, 0.01]$. It seems NRSEC is more robust to the parameters than RSEC.

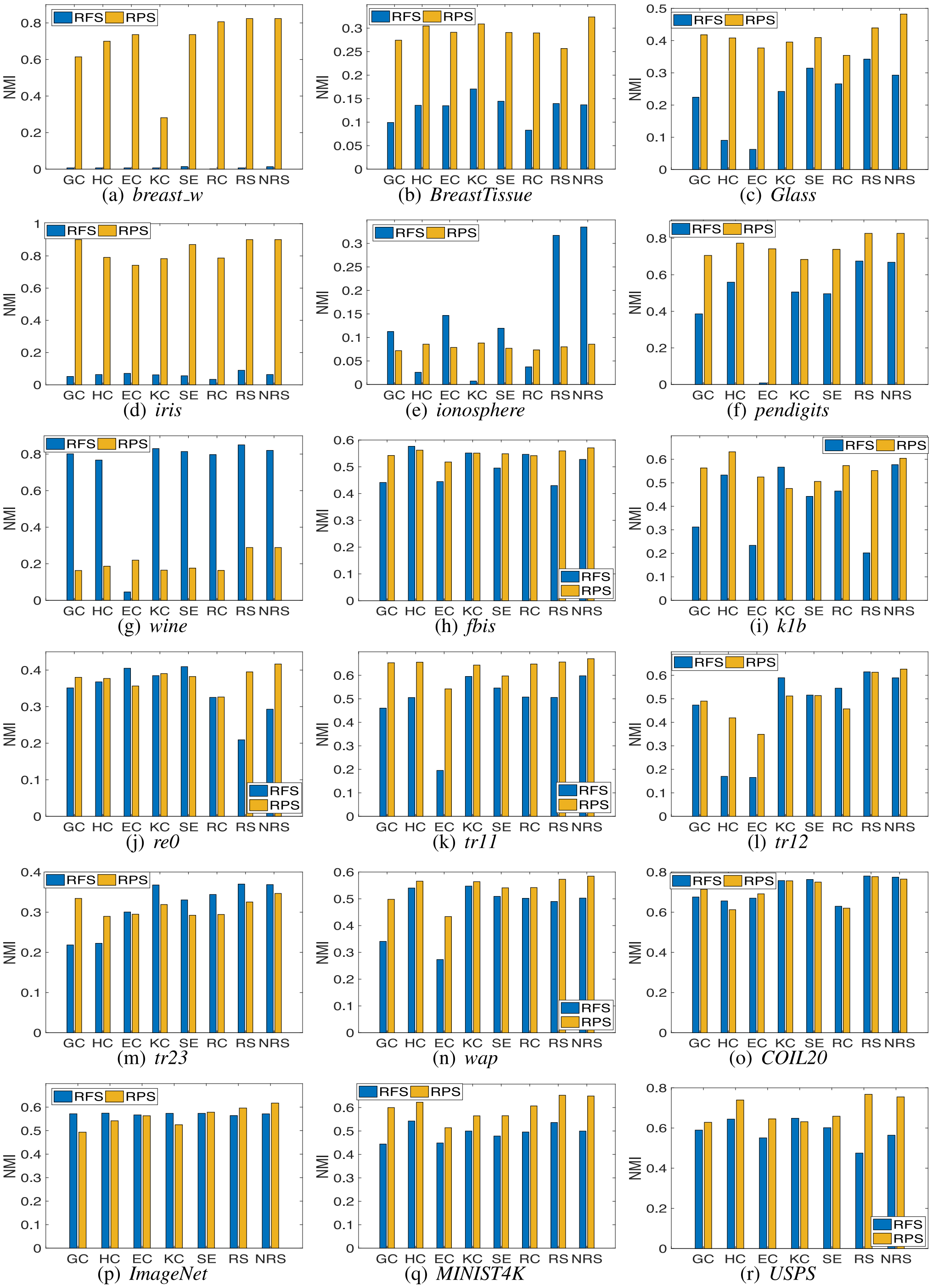
5.4 Exploration of Input BPs

In the following, we continue to explore some key factors of BPs for practical use, such as the number of BPs, the impact of missing data and another BPs generation strategy.

Impact of BPs Number. We vary the number of BPs to see the performance change of RSEC and NRSEC. Specifically, we randomly sample a certain number of BPs from Π and run RSEC and NRSEC for the consensus result. The above process is repeated 100 times for a fixed number of BPs. Figure 6 shows the boxplots of RSEC and NRSEC with the numbers of BPs increasing from 10 to 90 with 10 intervals on *breast_w*, *iris*, *wine* and *tr12*. It can be seen that the performance of RSEC and NRSEC goes up and the volatility becomes narrow with the increase of numbers of BPs. Moreover, when the number of BPs exceeds some threshold, the performance of RSEC and NRSEC converges. Therefore, we set the number of BPs to be 100 for a stable and robust solution for RSEC and NRSEC.

Impact of Incomplete BPs. In some real-world applications, BPs might be corrupted due to the transformation loss or device failure. These missing values lead to the incomplete BPs. One naive way to handle this scenario is to remove the instances with missing values, which is a kind of waste. In light of this, we extend our model to handle incomplete BPs by redefining the co-association matrix S as $\mathrm{S}(x_p, x_q) = \sum_{i=1}^{m} \Delta(\pi_i(x_p), \pi_i(x_q))$, and

$$\Delta(a,b) = \begin{cases} 1, & \text{if } a = b \text{ and } a, b \neq 0 \\ -1, & \text{if } a \neq b \text{ and } a, b \neq 0, \\ 0, & \text{if } a = 0 \text{ or } b = 0 \end{cases}$$



(a) *breast_w* (b) *BreastTissue* (c) *Glass*

(d) *iris* (e) *ionosphere* (f) *pendigits*

(g) *wine* (h) *fbis* (i) *k1b*

(j) *re0* (k) *tr11* (l) *tr12*

(m) *tr23* (n) *wap* (o) *COIL20*

(p) *ImageNet* (q) *MINIST4K* (r) *USPS*

Fig. 8. Comparison results between RFS and RPS on 18 datasets by *NMI*, where GC, HC, EC, KC, SE, RC, RE, and NRS denote GCC (Strehl and Ghosh 2003), HCC (Fred and Jain 2005), ECMC (Yi et al. 2012), KCC (Wu et al. 2015), SEC (Liu et al. 2015a), RCE (Zhou et al. 2015) and the proposed RSEC and NRSEC, respectively.

where $x_p, x_q \in \mathcal{X}$, and $\pi_i \in \Pi$. After a normalization by $\mathbf{S} = \mathbf{S}/m$, the entries of 0 value denote the most uncertainty in $\mathbf{S}$.

To mimic the data corruption, we randomly drop out some elements of each BP in Π according to a certain proportion τ, that is, to assign the missing elements in each BP as 0. To explore the impact of different corruption ratios, we vary τ from 10% to 90% with a fixed step of 10%, and sample incomplete BPs from Π for each proportion. Figure 7 shows the comparison result between our methods (RSEC and NRSEC) and several state-of-the-art EC methods with incomplete BPs on four datasets. As can be seen, RSEC and NRSEC outperform the competitors when $\tau \leq 40\%$ and achieve comparable results when corruption gets aggravated, which indicates our approach is effective to handle BPs with missing values.

Alternative BPs Generation Strategy. So far, we generate BPs with Random Parameter Section (RPS) strategy for all the experiments above. Here, we test EC methods with another common BPs generation strategy, called RFS (Wu et al. 2015), to further explore the impact of BPs to ensemble clustering performance. Different from RPS, RFS generates BPs by running K-means with randomly selected partial features. As following the empirical setting in (Liu et al. 2015b), we set the feature selection ration as 10% and generate 100 BPs for all the datasets.

Figure 8 shows the comparison results between RFS and RPS on 18 real-world datasets by *NMI*. One may note that, RFS significantly boosts the clustering performance over RPS on the *ionosphere* and *wine* dataset. We conjecture this is because these two datasets suffer from noisy features, which leads to low-quality BPs generated by RPS, while RFS alleviates this problem to some extent as it employs partial features. However, as shown by Figure 8, RPS still performs better than RFS in the majority cases. Hence, similar to previous works (Fred and Jain 2005; Wu et al. 2015; Liu et al. 2015a), we leverage RPS as the default BPs generation strategy and employ RFS as the alternative one.

6 CONCLUSION

In this article, we proposed a novel RSEC algorithm, which not only targeted at a denoising task for the co-association matrix for consensus clustering, but also reinforced the cluster structure. Generally speaking, we jointly learned a robust representation for the co-association matrix through low-rank constraint, and explored the cluster structure in a unified optimization framework. The consensus partition obtained from the learned representation was applied to further enhance its block-diagonal structure with an iterative manner. Moreover, we further proposed a non-convex model to improve the robustness of rank approximation, which exhibited higher performance on some imbalance datasets. Experimental results on 18 real-world datasets demonstrated that the new representation provided by RSEC enjoyed a much clearer structure than the ones learned from other methods and that our approaches outperformed several state-of-the-art methods in terms of *ACC* and *NMI*. Several impact factors of BPs generation were also explored extensively for practical use.

ACKNOWLEDGMENT

The authors would like to thank the associate editor and all the anonymous reviewers for their insightful comments on our article.

REFERENCES

H. Ayad and M. Kamel. 2008. Cumulative voting consensus method for partitions with variable number of clusters. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30, 1 (2008), 160–173.

D. Cai, X. He, J. Han, and T. S. Huang. 2011. Graph regularized nonnegative matrix factorization for data representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33, 8 (2011), 1548–1560.

J. Cai, E. J. Candès, and Z. Shen. 2010. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization* 20, 4 (2010), 1956–1982.

E. J. Candès, X. Li, Y. Ma, and J. Wright. 2011. Robust principal component analysis? *Journal of the ACM* 58, 3 (2011), 11:1–11:37.

X. Cao, Z. Tao, B. Zhang, H. Fu, and W. Feng. 2014. Self-adaptively weighted co-saliency detection via rank constraint. *IEEE Transactions on Image Processing* 23, 9 (2014), 4175–4186.

B. Cheng, G. Liu, J. Wang, Z. Huang, and S. Yan. 2011. Multi-task low-rank affinity pursuit for image segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*. 2439–2446.

I. S. Dhillon, Y. Guan, and B. Kulis. 2004. Kernel K-means: Spectral clustering and normalized cuts. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 551–556.

Z. Ding and Y. Fu. 2014. Low-rank common subspace for multi-view learning. In *Proceedings of the IEEE International Conference on Data Mining*. 110–119.

Z. Ding, M. Shao, and Y. Fu. 2015. Missing modality transfer learning via latent low-rank constraint. *IEEE Transactions on Image Processing* 24, 11 (2015), 4322–4334.

C. Domeniconi and M. Al-Razgan. 2009. Weighted cluster ensembles: Methods and analysis. *ACM Transactions on Knowledge Discovery from Data* 2, 4 (2009), 17:1–17:40.

X. Z. Fern and C. E. Brodley. 2004. Solving cluster ensemble problems by bipartite graph partitioning. In *Proceedings of International Conference on Machine Learning*.

A. L. N. Fred and A. K. Jain. 2005. Combining multiple clusterings using evidence accumulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27, 6 (2005), 835–850.

A. Gionis, H. Mannila, and P. Tsaparas. 2007. Clustering aggregation. *ACM Transactions on Knowledge Discovery from Data* 1, 1 (2007), Article 4.

N. Iam-on, T. Boongoen, S. M. Garrett, and C. J. Price. 2011. A link-based approach to the cluster ensemble problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33, 12 (2011), 2396–2409.

A. K. Jain. 2010. Data clustering: 50 years beyond K-means. *Pattern Recognition Letter* 31, 8 (2010), 651–666.

L. I. Kuncheva and D. Vetrov. 2006. Evaluation of stability of k-means cluster ensembles with respect to random initialization. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28, 11 (2006), 1798–1808.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86, 11 (1998), 2278–2324.

J. Li, Y. Kong, H. Zhao, J. Yang, and Y. Fu. 2016b. Learning fast low-rank projection for image classification. *IEEE Transaction on Image Processing* 25, 10 (2016), 4803–4814.

J. Li, H. Liu, H. Zhao, and Y. Fu. 2017b. Projective low-rank subspace clustering via learning deep encoder. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. 2145–2151.

J. Li, H. Zhao, Z. Tao, and Y. Fu. 2017c. Large-scale subspace clustering by fast regression coding. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*. 2138–2144.

S. Li and Y. Fu. 2014. Robust subspace discovery through supervised low-rank constraints. In *Proceedings of SIAM International Conference on Data Mining*. 163–171.

S. Li and Y. Fu. 2015. Learning balanced and unbalanced graphs via low-rank coding. *IEEE Transactions on Knowledge and Data Engineering* 27, 5 (2015), 1274–1287.

S. Li and Y. Fu. 2016. Learning robust and discriminative subspace with low-rank constraints. *IEEE Transaction Neural Networks Learning System* 27, 11 (2016), 2160–2173.

S. Li, K. Li, and Y. Fu. 2018a. Self-taught low-rank coding for visual learning. *IEEE Transactions on Neural Networks and Learning Systems* 29, 3 (2018), 645–656.

S. Li, H. Liu, Z. Tao, and Y. Fu. 2017a. Multi-view graph learning with adaptive label propagation. In *Proceedings of the IEEE International Conference on Big Data (Big Data'17)*. IEEE, 110–115.

S. Li, M. Shao, and Y. Fu. 2018b. Multi-view low-rank analysis with applications to outlier detection. *ACM Transactions on Knowledge Discovery from Data* 12, 3 (2018), Article 32.

T. Li, B. Cheng, B. Ni, G. Liu, and S. Yan. 2016a. Multitask low-rank affinity graph for image segmentation and image annotation. *ACM Transactions on Intelligent Systems and Technology* 7, 4 (2016), Article 65.

T. Li, C. Ding, and M. I. Jordan. 2007. Solving consensus and semi-supervised clustering problems using nonnegative matrix factorization. In *Proceedings of the IEEE International Conference on Data Mining*. 577–582.

Z. Lin, M. Chen, and Y. Ma. 2010. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. arXiv:1009.5005.

Z Lin, R. Liu, and Z. Su. 2011. Linearized alternating direction method with adaptive penalty for low-rank representation. In *Proceedings of 24th International Conference on Neural Information Processing Systems*. 612–620.

G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma. 2013. Robust recovery of subspace structures by low-rank representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35, 1 (2013), 171–184.

G. Liu, Z. Lin, and Y. Yu. 2010. Robust subspace segmentation by low-rank representation. In *Proceedings of the 27th International Conference on Machine Learning*. 663–670.

G. Liu and S. Yan. 2011. Latent low-rank representation for subspace segmentation and feature extraction. In *Proceedings of the IEEE International Conference on Computer Vision*. 1615–1622.

H. Liu, T. Liu, J. Wu, D. Tao, and Y. Fu. 2015a. Spectral ensemble clustering. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 715–724.

H. Liu, M. Shao, S. Li, and Y. Fu. 2016. Infinite ensemble for image clustering. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1745–1754.

H. Liu, M. Shao, S. Li, and Y. Fu. 2018. Infinite ensemble clustering. *Data Mining and Knowledge Discovery* 32, 2 (2018), 385–416.

H. Liu, J. Wu, T. Liu, D. Tao, and Y. Fu. 2017. Spectral ensemble clustering via weighted K-means: Theoretical and practical evidence. *IEEE Transactions on Knowledge and Data Engineering* 29, 5 (2017), 1129–1143.

H. Liu, J. Wu, D. Tao, Y. Zhang, and Y. Fu. 2015b. Dias: A disassemble-assemble framework for highly sparse text clustering. In *Proceedings of SIAM International Conference on Data Mining*. 766–774.

C. Lu, J. Tang, S. Yan, and Z. Lin. 2014. Generalized nonconvex nonsmooth low-rank minimization. In *IEEE Conference on Computer Vision and Pattern Recognition*. 4130–4137.

Z. Lu, Y. Peng, and J. Xiao. 2008. From comparing clusterings to combining clusterings. In *Proceedings of AAAI Conference on Artificial Intelligence*. 665–670.

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

B. Minaei-Bidgoli, A. P. Topchy, and W. F. Punch. 2004. A comparison of resampling methods for clustering ensembles. In *Proceedings of the International Conference on Machine Learning; Models, Technologies and Applications*. 939–945.

B. Mirkin. 2001. Reinterpreting the category utility function. *Machine Learning* 45, 2 (2001), 219–228.

S. A. Nene, S. K. Nayar, and H. Murase. 1996. *Columbia Object Image Library (COIL-20)*. Technical Report CUCS-005-96.

A. Y. Ng, M. I. Jordan, and Y. Weiss. 2001. On spectral clustering: Analysis and an algorithm. In *Proceedings of Advances in Neural Information Processing Systems*. 849–856.

T. Hyun Oh, Y. Matsushita, Y. Tai, and I. Kweon. 2015. Fast randomized singular value thresholding for nuclear norm minimization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4484–4493.

C. Peng, Z. Kang, H. Li, and Q. Cheng. 2015. Subspace clustering using log-determinant rank approximation. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 925–934.

M. Shao, D. Kit, and Y. Fu. 2014. Generalized transfer subspace learning through low-rank constraint. *International Journal of Computer Vision* 109, 1–2 (2014), 74–93.

M. Shao, S. Li, Z. Ding, and Y. Fu. 2015. Deep linear coding for fast graph clustering. In *Proceedings of International Joint Conference on Artificial Intelligence*. 3798–3804.

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

A. Strehl and J. Ghosh. 2003. Cluster ensembles — A knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research* 3 (2003), 583–617.

Z. Tao, H. Liu, and Y. Fu. 2017a. Simultaneous clustering and ensemble. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*. 1546–1552.

Z. Tao, H. Liu, S. Li, Z. Ding, and Y. Fu. 2017b. From ensemble clustering to multi-view clustering. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. 2843–2849.

Z. Tao, H. Liu, S. Li, and Y. Fu. 2016. Robust spectral ensemble clustering. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. 367–376.

A. Topchy, A. K. Jain, and W. Punch. 2003. Combining multiple weak clusterings. In *Proceedings of IEEE International Conference on Data Mining*. 331–338.

A. Topchy, A Jain, and W. Punch. 2004. A mixture model for clustering ensembles. In *Proceedings of SIAM International Conference on Data Mining*. 379–390.

A. Topchy, A Jain, and W. Punch. 2005. Clustering ensembles: Models of consensus and weak partitions. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27, 12 (2005), 1866–1881.

S. Vega-Pons, J. Correa-Morris, and J. Ruiz-Shulcloper. 2010. Weighted partition consensus via kernels. *Pattern Recognition* 43, 8 (2010), 2712–2724.

S. Wang, D. Liu, and Z. Zhang. 2013. Nonconvex relaxation approaches to robust matrix recovery. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence*. 1764–1770.

J. Wright, A. Ganesh, S. Rao, Y. Peng, and Y. Ma. 2009. Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization. In *Proceedings of Advances in Neural Information Processing Systems*. 2080–2088.

J. Wu, H. Liu, H. Xiong, J. Cao, and J. Chen. 2015. K-means-based consensus clustering: A unified view. *IEEE Transactions on Knowledge and Data Engineering* 27, 1 (2015), 155–169.

J. Wu, H. Xiong, and J. Chen. 2009. Adapting the right measures for K-means clustering. In *Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 877–886.

S. Xiao, W. Li, D. Xu, and D. Tao. 2015. FaLRR: A fast low rank representation solver. In *IEEE Conference on Computer Vision and Pattern Recognition*. 4612–4620.

D. Yan, L. Huang, and M. I. Jordan. 2009. Fast approximate spectral clustering.. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 907–916.

J. Yi, T. Yang, R. Jin, A. K. Jain, and M. Mahdavi. 2012. Robust ensemble clustering by matrix completion. In *Proceedings of IEEE International Conference on Data Mining*. 1176–1181.

M. Yin, J. Gao, and Z. Lin. 2016. Laplacian regularized low-rank representation and its application. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38, 3 (2016), 504–517.

H. Yoon, S. Ahn, S. Lee, S. Cho, and J. Kim. 2006. Heterogeneous clustering ensemble method for combining different cluster results. *Data Mining for Biomedical Applications* 3916 (2006), 82–92.

C. Zhang. 2010. Nearly unbiased variable selection under minimax concave penalty. *Annals of Statistics* 38, 2 (2010), 894–942.

L. Zheng, T. Li, and C. Ding. 2014. A framework for hierarchical ensemble clustering. *ACM Transactions on Knowledge Discovery from Data* 9, 2 (2014), 9:1–9:23.

P. Zhou, L. Du, H. Wang, L. Shi, and Y. Shen. 2015. Learning a robust consensus matrix for clustering ensemble via Kullback–Leibler divergence minimization. In *Proceedings of International Joint Conference on Artificial Intelligence*. 4112–4118.

Received December 2016; revised February 2018; accepted September 2018