

Regularized Non-negative Spectral Embedding for Clustering

Yifei Wang, Rui Liu, Yong Chen(Corresponding), Hui Zhang, Zhiwen Ye

School of Computer Science and Engineering, Beihang University, Beijing 100191, China

{yifei.wang,lr}@buaa.edu.cn, {chenyong,hzhang}@nlsde.buaa.edu.cn, yezhiwen@buaa.edu.cn

Abstract—Spectral Clustering is a popular technique to split data points into groups, especially for complex datasets. The algorithms in the Spectral Clustering family typically consist of multiple separate stages (such as similarity matrix construction, low-dimensional embedding, and K-Means clustering as post-processing), which may lead to sub-optimal results because of the possible mismatch between different stages. In this paper, we propose an end-to-end single-stage learning method to clustering called Regularized Non-negative Spectral Embedding (RNSE) which extends Spectral Clustering with the adaptive learning of similarity matrix and meanwhile utilizes non-negative constraints to facilitate one-step clustering (directly from data points to clustering labels). Two well-founded methods, *successive alternating projection* and *strategic multiplicative update*, are employed to work out the quite challenging optimization problems in RNSE. Extensive experiments on both synthetic and real-world datasets demonstrate RNSE’s superior clustering performance to some state-of-the-art competitors.

Index Terms—Spectral Clustering, One-stage Learning, Successive Alternating Projection, Strategic Multiplicative Update

I. INTRODUCTION

Clustering is an important unsupervised learning task which aims to group a set of data objects into clusters in such a way that objects in the same cluster are more similar to each other than those in different clusters. For complex datasets, Spectral Clustering [1] and its many variants [2]–[4] are particularly popular due to their ability of discovering highly non-convex clusters. Such algorithms make use of the spectrum of the similarity matrix of the data to perform dimensionality reduction before grouping objects in a low-dimensional space [1], [5]–[10]. Typically, the algorithms in the Spectral Clustering family consist of multiple separate stages as follows:

- (1) Construct a pairwise similarity matrix, e.g., according to the k -nearest-neighbors graph of the data;
- (2) Compute the corresponding Laplacian matrix and normalize it;
- (3) Represent the data objects in a low-dimensional space using a few eigenvectors of the Laplacian matrix corresponding to its smallest eigenvalues;
- (4) Re-normalize the embedded data vectors and then group them into C clusters, e.g., through the classic K-Means algorithm [11].

The above multi-stage approach may lead to sub-optimal clustering results due to the possible mismatch between different stages. Moreover, there is still much room for improvement

in the optimization methods. For example, normalizing the similarity matrix into a doubly stochastic matrix¹ has been found to be beneficial [6], and imposing some global priors (like Laplacian rank) could help to reveal the underlying clustering structure of the dataset [9], [12].

In light of the above analysis, we extend Spectral Clustering and propose an end-to-end single-stage learning framework for clustering named “Regularized Non-negative Spectral Embedding”, or RNSE in short. It does not rely on a predefined similarity matrix but learn the similarity matrix in a data-driven self-adaptive manner. Furthermore, it introduces two global priors, i.e., the doubly stochastic constraint (for normalizing the similarity matrix) and the non-negative low-rank constraint (for capturing the intrinsic clustering structure), to facilitate the optimization. The effectiveness of RNSE for clustering has been confirmed by extensive experiments on both synthetic and real-world datasets.

II. RELATED WORK

Our proposed clustering technique RNSE is closely connected to several research areas.

First of all, RNSE obviously has its roots in the Spectral Clustering [1] and also its many variants including Laplacian Eigenmap (LE) [5], Locality Preserving Projections (LPP) [13], and Spectral Regression (SR) [7]. Those techniques start from a predefined pairwise similarity matrix and perform clustering (and other tasks) via spectral decomposition. They typically consist of several separate stages. In contrast, our proposed RNSE technique carries out the whole process from the given data straight to the clustering result in just one stage, with the similarity matrix automatically learned.

Next, the formulation of RNSE imposes non-negative constraints, so it is related to the series of Non-negative Matrix Factorization (NMF) [14] techniques including Non-negative Matrix Tri-factorization (NM3F) [15] and Graph regularized NMF (GNMF) [16]. Those techniques decompose the data matrix into two or more low-rank non-negative matrices from which the clustering structures of the data could be read out. However, our proposed RNSE technique is different from them as it contains two sub-problems of optimization with non-negative constraints which are combined in a unified optimization framework. Thus RNSE could obtain the clustering

¹A doubly stochastic matrix is a square matrix that satisfies $\mathbf{S}\mathbf{1} = \mathbf{1}$, $\mathbf{S}^T\mathbf{1} = \mathbf{1}$, and $\mathbf{S} \geq 0$, where $\mathbf{1}$ is a column vector with all elements to be 1 and $\mathbf{S} \geq 0$ represents element-wise non-negativity.

results directly after solving the optimization problem, while those NMF-based methods need some post-processing such as using the K-Means algorithm [11] to get the final clustering results.

Last but not the least, RNSE utilizes structural regularization in its learning algorithm. Generally speaking, it is useful to incorporate appropriate priors into the learning process as the priors could help to find the intrinsic structure of data. Locality Preserving Projections (LPP) [13] tries to maintain the k -nearest-neighbors graph while performing linear dimensionality reduction of the data. Similarly, Graph Regularized Non-negative Matrix Factorization (GNMF) [16] adds a k -nearest-neighbors graph based regularizer term to the vanilla NMF algorithm. Those two techniques both use the local (manifold) structure of data for regularization. There also exist techniques with global structural regularization. For example, Doubly Stochastic Normalization (DSN) [6] enforces the doubly stochastic condition on the similarity matrix before carrying out Spectral Clustering. Besides, Structured Doubly Stochastic Matrix (SDS) [17], Clustering with Adaptive Neighbors (CAN) [18] and Constrained Laplacian Rank (CLR) [9] borrow the idea of C -connected components (cf. Theorem 1) from the spectral graph theory to form the regularization for their learning algorithms. Inspired by the above methods, our proposed RNSE technique utilizes both global structures (i.e., doubly stochastic matrix and non-negativity constrained C -connected components) as regularizers for clustering.

III. THE PROPOSED APPROACH

Given a data matrix $\mathbf{X} = [x_1, x_2, \dots, x_N] \in \mathbb{R}^{M \times N}$, where M is the dimension of a sample, N marks the number of the total samples, and $x_i \in \mathbb{R}^M$ denotes the i -th sample ($i = 1, 2, \dots, N$). Let $\mathbf{S} \in \mathbb{R}^{N \times N}$ be the similarity matrix and S_{ij} corresponds to the similarity between x_i and x_j . Besides, consider that $\Phi(\cdot) : x \in \mathbb{R}^M \mapsto \mathcal{H}$ be a feature mapping from x onto a reproducing kernel Hilbert space \mathcal{H} and there exists $\mathcal{K}(x, y) = \langle \Phi(x), \Phi(y) \rangle$. Classic clustering methods usually pre-compute the similarity matrix based on the Euclid distance between pairwise samples. However, we formulate it as a data-based learning problem:

$$\begin{aligned} \min_{\mathbf{S}} \mathcal{O}(\mathbf{S}) &= \frac{1}{2} \sum_{ij} S_{ij} \|\Phi(x_i) - \Phi(x_j)\|_2^2 \quad (1) \\ s.t. \mathbf{S} &\geq 0, \mathbf{S} = \mathbf{S}^T, \mathbf{S}\mathbf{1} = \mathbf{1}, \end{aligned}$$

where \mathbf{S} is constrained to become a doubly stochastic matrix for better clustering [6], [19]. Note that we utilize the distance between the kernel mappings instead of the original vectors in Eq. (1) because $\|\cdot\|_2$ measures the Euclid space and kernel mappings in the Hilbert space would match such characteristic. Based on the similarity matrix \mathbf{S} , the classic spectral embedding methods [1], [5], [6], [9], [10] could be formulated as below:

$$\min_{\mathbf{P}} \frac{1}{2} \sum_{ij} S_{ij} \|\mathbf{P}_i - \mathbf{P}_j\|_2^2, \quad s.t. \mathbf{P}\mathbf{P}^T = \mathbf{I}_E, \quad (2)$$

where \mathbf{P} is the spectral embedding for the data points and E is the dimension of the embedding vectors. Thereafter, K-Means is adopted to cluster the data samples as a popular post-processing technique. However, we further put the non-negativity on \mathbf{P} , set E to C (the number of clusters), and finally arrive at:

$$\begin{aligned} \min_{\mathbf{P}} \mathcal{O}(\mathbf{P}) &= \frac{1}{2} \sum_{ij} S_{ij} \|\mathbf{P}_i - \mathbf{P}_j\|_2^2, \quad (3) \\ s.t. \mathbf{P} &\geq 0, \mathbf{P}\mathbf{P}^T = \mathbf{I}_C. \end{aligned}$$

Noticeably, each column of \mathbf{P} will be one-hot vector; in other words, \mathbf{P} could be treated as an indicator matrix for clustering. Besides, from the following Theorem 1 and Theorem 2, we can conclude that the optimization (3) actually captures the intrinsic structures, i.e., C -collected clusters.

Theorem 1 (C -connected clusters [20]). “The multiplicity C of the eigenvalue 0 of the Laplacian matrix \mathbf{L}_S is equal to the number of connected clusters/components in the graph associated with \mathbf{S} ”, which implies:

$$\text{rank}(\mathbf{L}_S) = N - C \Leftrightarrow \sum_{i=1}^C \lambda_i = 0, \quad (4)$$

where $\mathbf{L}_S = \text{diag}(\mathbf{S}\mathbf{1})^2 - \mathbf{S}$, and $\{\lambda_1, \lambda_2, \dots, \lambda_N\}$ are the eigenvalues of \mathbf{L}_S in an ascending order.

Theorem 2 (Ky Fan’s Theorem [21]). Given a matrix $\mathbf{P} \in \mathbb{R}^{C \times N}$, the following optimization problem:

$$\begin{aligned} \min_{\mathbf{P}} \frac{1}{2} \sum_{ij} S_{ij} \|\mathbf{P}_i - \mathbf{P}_j\|_2^2 &= \text{tr}(\mathbf{P} \cdot \mathbf{L}_S \cdot \mathbf{P}^T) \quad (5) \\ s.t. \mathbf{P}\mathbf{P}^T &= \mathbf{I}_C \end{aligned}$$

is equivalent to $\sum_{i=1}^C \lambda_i \rightarrow 0$.

Generally, learning with multi-stages, e.g., Spectral Clustering, would usually lead to sub-optimal solutions for clustering. Therefore, we build a marriage between (1) and (3) into a joint learning approach:

$$\min_{\mathbf{S}, \mathbf{P}} \mathcal{O}(\mathbf{S}, \mathbf{P}), \quad s.t. \mathbf{S} \in \mathbf{S}_{dsm}, \mathbf{P} \in \mathbf{P}_{nlr}, \quad (6)$$

with

$$\mathcal{O}(\mathbf{S}, \mathbf{P}) = \mathcal{O}(\mathbf{S}) + \alpha \|\mathbf{S}\|_F^2 + \beta \mathcal{O}(\mathbf{P}), \quad (7)$$

$$\mathbf{S}_{dsm} = \{\mathbf{S} \geq 0 | \mathbf{S} = \mathbf{S}^T, \mathbf{S}\mathbf{1} = \mathbf{1}\}, \quad (8)$$

$$\mathbf{P}_{nlr} = \{\mathbf{P} \geq 0 | \mathbf{P} \cdot \mathbf{P}^T = \mathbf{I}_C\}. \quad (9)$$

Obviously, \mathbf{S}_{dsm} is a set of doubly stochastic matrices, \mathbf{P}_{nlr} is a set of non-negative low-rank matrices, and α and β are two positive hyper-parameters. The philosophy of the optimization (6) is an end-to-end single-stage learning for clustering based on non-negative spectral embedding. Therefore, we call our method “Regularized Non-negative Spectral Embedding (RNSE)” for Clustering.

² $\text{diag}(\cdot)$ is a diagonal matrix spanned by the vector parameter. For instance, $\text{diag}(\mathbf{1})$ is actually an Identity matrix \mathbf{I} . However, if the parameter is a square matrix, then it will return a column vector with the diagonal elements, e.g., $\text{diag}(\mathbf{I}) = \mathbf{1}$.

Algorithm 1: The learning process: RNSE

Input: Dataset $\{x_i\}_{i=1}^N$; hyper-parameters α, β .**Output:** Similarity matrix \mathbf{S} and indicator matrix \mathbf{P} .

```

1 begin
2   Randomly initialize indicator matrix  $\mathbf{P}$  and similarity
   matrix  $\mathbf{S}$ ;
3   repeat
4     Optimize problem (6) w.r.t.  $\mathbf{S}$  while keeping  $\mathbf{P}$ 
       fixed, i.e., Algorithm 2;
5     Optimize problem (6) w.r.t.  $\mathbf{P}$  while keeping  $\mathbf{S}$ 
       fixed, i.e., Algorithm 3;
6   until convergence;

```

IV. OPTIMIZATION METHODS

Regarding the objective function $\mathcal{O}(\mathbf{S}, \mathbf{P})$, there are two coupled variables to be learned which indicates that it's a non-convex optimization problem. Thus, we adopt the classic strategies to address such optimization problem with alternative iterations [14], [16], [22], i.e., updating \mathbf{P} while keeping \mathbf{S} fixed and vice versa, until a local minima is achieved. The learning process is narrated in Algorithm 1. Subsequently, we depict the detailed ideas for solving the two subproblems.

A. Optimizing \mathbf{S} while keeping \mathbf{P} fixed

When clustering indicator matrix \mathbf{P} is fixed, the subproblem for optimizing similarity matrix \mathbf{S} can be written as:

$$\min_{\mathbf{S}} \mathcal{O}(\mathbf{S}, \mathbf{P}) \quad \text{s.t. } \mathbf{S} \in \mathbf{S}_{dsm}. \quad (10)$$

Set $\mathbf{A} = [\Phi(x_1), \Phi(x_2), \dots, \Phi(x_N)]$, then we can transform (10) into:

$$\min_{\mathbf{S}} \alpha \|\mathbf{S}\|_F^2 + \text{tr}(\mathbf{A} \cdot \mathbf{L}_S \cdot \mathbf{A}^T) + \beta \cdot \text{tr}(\mathbf{P} \cdot \mathbf{L}_S \cdot \mathbf{P}^T) \quad (11)$$

$$\text{s.t. } \mathbf{S} \in \mathbf{S}_{dsm},$$

which is further equivalent to the simplified formalization:

$$\min_{\mathbf{S}} \|\mathbf{S} - \mathbf{T}\|_F^2, \quad \text{s.t. } \mathbf{S} \in \mathbf{S}_{dsm}, \quad (12)$$

with $\mathbf{T} = \frac{1}{2\alpha} [(\mathbf{A}^T \mathbf{A} + \beta \mathbf{P}^T \mathbf{P}) - \text{diag}(\mathbf{A}^T \mathbf{A} + \beta \mathbf{P}^T \mathbf{P}) \cdot \mathbf{1}^T]$. Essentially, this is an optimization problem to find a doubly stochastic matrix \mathbf{S} nearest to the given matrix \mathbf{T} , which could be converted into the “metric projection optimizations” and solved with alternating projection methods.

Definition 1 (Metric Projection). *Given a set $\mathcal{S} \subseteq \mathcal{H}$ and a point $x \in \mathcal{H}$, the metric projection (if exists) of x onto \mathcal{S} is a point $p \in \mathcal{S}$ such that:*

$$\|p - x\| = d(x, \mathcal{S}) := \inf_{s \in \mathcal{S}} \|x - s\|. \quad (13)$$

Additionally, if for any $x \in \mathcal{H}$, there exists such a unique p , then the metric projection onto \mathcal{S} is rewritten as the following operator:

$$P_{\mathcal{S}} : \mathcal{H} \mapsto \mathcal{S}, \quad \text{i.e., } P_{\mathcal{S}}(x) = p. \quad (14)$$

Algorithm 2: Optimizing \mathbf{S} while keeping \mathbf{P} fixed

Input: Dataset $\{x_i\}_{i=1}^N$, indicator matrix \mathbf{P} , hyper-parameters α, β .**Output:** Similarity matrix \mathbf{S} .

```

1 begin
2   Initialize  $x_1^0 := \mathbf{T}$ ,  $I_0^1 := 0$ ,  $I_0^2 := 0$ ,  $n = 1$ ;
3   repeat
4     Calculate  $x_n^1 := P_{C_1}(x_n^0)$  and  $x_n^2 := P_{C_2}(x_n^1)$ ;
5     Calculate  $I_n^1 := x_n^1 - (x_n^0 - I_{n-1}^1)$  and
        $I_n^2 := x_n^2 - (x_n^1 - I_{n-1}^2)$ ;
6     Iterative index  $n = n + 1$ ;
7      $x_n^0 := x_{n-1}^2$ ;
8   until convergence;

```

Theorem 3 (Projection Theorem). *Set $\mathcal{C} \subseteq \mathcal{H}$ be a closed convex set. For any $x \in \mathcal{H}$, there exists a unique $p \in \mathcal{C}$ such that $\|x - p\| \leq \|x - c\|$ for all $c \in \mathcal{C}$, which is formally denoted as:*

$$P_{\mathcal{C}}(x) = p. \quad (15)$$

Proof. See Ref. [23]. \square

Theorem 4 (Dykstra's Method). *Let $\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_r \subseteq \mathcal{H}$ be closed convex sets and $\mathcal{C} := \bigcap_{i=1}^r \mathcal{C}_i$. If $\mathcal{C} \neq \emptyset$, then given $x \in \mathcal{H}$ iterated by:*

$$\begin{cases} x_n^i := P_{\mathcal{C}_i}(x_n^{i-1} - I_{n-1}^i) \\ I_n^i := x_n^i - (x_n^{i-1} - I_{n-1}^i) \\ x_n^0 := x_{n-1}^r \end{cases}, \quad (16)$$

with initial values $x_1^0 := x$, $I_0^i := 0$, there holds:

$$\lim_{n \rightarrow \infty} x_n = P_{\mathcal{C}}(x). \quad (17)$$

Proof. See Refs. [24], [25] and [26]. \square

Theorem 5. *Let $\mathcal{H} = \mathbb{R}^{N \times N}$ and $\mathcal{S} = \{\mathbf{M} \in \mathcal{H} | \mathbf{M} = \mathbf{M}^T, \mathbf{M}\mathbf{1} = \mathbf{1}, \mathbf{M} \geq 0\}$, then \mathcal{S} is a closed convex set.*

Proof. This can be easily verified. \square

Now, let's go back to subproblem (12). Firstly, Theorem 3 and Theorem 5 together tell us that there must be one global and unique \mathbf{S} for the optimization problem (12). Then, inspired by the Theorem 4, we could turn (12) into:

$$x_n^1 := P_{C_1}(x_n^0), \quad (18)$$

and

$$x_n^2 := P_{C_2}(x_n^1), \quad (19)$$

where $\mathcal{C}_1 = \{\mathbf{S} | \mathbf{S} = \mathbf{S}^T, \mathbf{S}\mathbf{1} = \mathbf{1}\}$, $\mathcal{C}_2 = \{\mathbf{S} | \mathbf{S} \geq 0\}$, $x_1^0 := \mathbf{T}$, $x_n^0 := x_{n-1}^2$, and $n = 1, 2, \dots$ denotes the iterative index. The solutions to the two sub-problems (18) and (19) could be achieved by Theorem 6 and Theorem 7, respectively.

Theorem 6. *Given any point \mathbf{T}_v , the global optimal solution to $P_{C_1}(\mathbf{T}_v)$ is*

$$\mathbf{K} + \frac{N + \mathbf{1}^T \mathbf{K} \mathbf{1}}{N^2} \mathbf{1} \mathbf{1}^T - \frac{1}{N} \mathbf{K} \mathbf{1} \mathbf{1}^T - \frac{1}{N} \mathbf{1} \mathbf{1}^T \mathbf{K}, \quad (20)$$

where $\mathbf{K} = \frac{\mathbf{T}_v + \mathbf{T}_v^T}{2}$.

Proof. See Ref. [17]. \square

Theorem 7. Given any point \mathbf{T}_v , the global optimal solution to $P_{C_2}(\mathbf{T}_v)$ is

$$[\mathbf{T}_v]_+, \quad (21)$$

where $[\cdot]_+$ is an element-wise non-negative operation.

Proof. This can be easily verified. \square

Up to present, we could achieve the optimal solution for problem (12) through Dykstra's method, which is concluded in Algorithm 2.

B. Optimizing \mathbf{P} while keeping \mathbf{S} fixed

It is straightforward to see that, when similarity matrix \mathbf{S} is fixed, the optimization problem (6) could be reduced as:

$$\min_{\mathbf{P}} \mathcal{O}(\mathbf{P}), \quad s.t. \mathbf{P} \in \mathbf{P}_{nlr}. \quad (22)$$

Regarding that \mathbf{P} is constrained to be both orthogonal and non-negative in (22), it seems quite challenging to deal with such problem. Subtly, since the similarity matrix \mathbf{S} is a doubly stochastic matrix, then we could draw the following Theorem 8.

Theorem 8 (P-Transformer). Given \mathbf{S} is a doubly stochastic matrix, the subproblem (22) could be converted to the following optimization problem:

$$\min_{\mathbf{P}} \|\mathbf{S} - \mathbf{P}^T \mathbf{P}\|_F^2, \quad s.t. \mathbf{P} \in \mathbf{P}_{nlr}. \quad (23)$$

Proof. Since $\mathbf{S} \in \mathbf{S}_{dsm}$, then there is $\mathbf{L}_S = \mathbf{I}_C - \mathbf{S}$, which is followed by:

$$\text{tr}(\mathbf{P} \cdot \mathbf{L}_S \cdot \mathbf{P}^T) = C - \text{tr}(\mathbf{P} \cdot \mathbf{S} \cdot \mathbf{P}^T).$$

Then the objective function in (22) could be further transformed into:

$$\begin{aligned} & \min_{\mathbf{P}} \text{tr}(\mathbf{P} \cdot \mathbf{L}_S \cdot \mathbf{P}^T) \\ & \propto \max_{\mathbf{P}} \text{tr}(\mathbf{P} \cdot \mathbf{S} \cdot \mathbf{P}^T) \\ & \propto \min_{\mathbf{P}} \{ \text{tr}(\mathbf{S} \cdot \mathbf{S}^T) - 2\text{tr}(\mathbf{S} \cdot \mathbf{P}^T \mathbf{P}) + \text{tr}(\mathbf{P}^T \mathbf{P} \mathbf{P}^T \mathbf{P}) \} \\ & \propto \min_{\mathbf{P}} \|\mathbf{S} - \mathbf{P}^T \mathbf{P}\|_F^2. \end{aligned} \quad (24)$$

Consequently, the two optimization problems (22) and (23) are equivalent. \square

In light of the optimization problem (23), its augmented Lagrange function is displayed as:

$$\mathcal{L}(\mathbf{P}, \Lambda) = \|\mathbf{S} - \mathbf{P}^T \mathbf{P}\|_F^2 + \text{tr}\{\Lambda(\mathbf{P} \mathbf{P}^T - \mathbf{I}_C)\}, \quad (25)$$

where Λ denotes the Lagrange multiplier for the constraint $\mathbf{P} \mathbf{P}^T = \mathbf{I}_C$. With respect to the non-negative constraints, they could be ignored in the augmented Lagrange function when the multiplicative update philosophy [22], [27], [28] is adopted, because the non-negativity for variable \mathbf{P} is naturally maintained during iterative updatings. More specifically, the

Algorithm 3: Optimizing \mathbf{P} while keeping \mathbf{S} fixed

Input: Similarity matrix \mathbf{S} and hyper-parameters λ, μ
Output: Clustering indicator matrix \mathbf{P} .

```

1 begin
2   Randomly initialize clustering indicator matrix  $\mathbf{P}$ ;
3   repeat
4     Update  $\mathbf{P}$  according to the strategic multiplicative
       update rule (34);
5   until convergence;
```

derivative of $\mathcal{L}(\mathbf{P}, \Lambda)$ w.r.t. \mathbf{P} can be formulated into two parts, i.e.,

$$\begin{aligned} \frac{\partial \mathcal{L}(\mathbf{P}, \Lambda)}{\partial \mathbf{P}} &= 4\mathbf{P} \mathbf{P}^T \mathbf{P} - 2\mathbf{P}(\mathbf{S} + \mathbf{S}^T) + (\Lambda + \Lambda^T) \mathbf{P} \\ &\triangleq (\mathcal{A})_+ - (\mathcal{B})_+, \end{aligned} \quad (26)$$

where $(\mathcal{A})_+$ and $(\mathcal{B})_+$ both represent element-wise non-negative matrices. Then the iterative formula for updating \mathbf{P} could be written as:

$$\mathbf{P} \leftarrow \mathbf{P} \otimes (\mathcal{B})_+ \oslash (\mathcal{A})_+, \quad (27)$$

where \otimes and \oslash corresponds to the element-wise multiplication and element-wise division respectively [29]. Obviously, if the factors in Equation (27) are all non-negative, then the result will also hold non-negativity. In order to figure out the detailed formula for Equation (27), we have to derive Λ .

Taking the partial derivatives of $\mathcal{L}(\mathbf{P}, \Lambda)$ w.r.t. \mathbf{P} and Λ respectively, and setting them to zero by the Karush-Kuhn-Tucker conditions, we arrive at:

$$\frac{\partial \mathcal{L}(\mathbf{P}, \Lambda)}{\partial \mathbf{P}} = 4\mathbf{P} \mathbf{P}^T \mathbf{P} - 2\mathbf{P}(\mathbf{S} + \mathbf{S}^T) + (\Lambda + \Lambda^T) \mathbf{P} = 0, \quad (28)$$

and

$$\frac{\partial \mathcal{L}(\mathbf{P}, \Lambda)}{\partial \Lambda} = \mathbf{P} \mathbf{P}^T - \mathbf{I}_C = 0. \quad (29)$$

Multiplying factor \mathbf{P}^T on both sides of Equation (28) and then taking the Equation (29) into it, there comes:

$$\Lambda + \Lambda^T = 2\mathbf{P}(\mathbf{S} + \mathbf{S}^T) \mathbf{P}^T - 4\mathbf{I}_C. \quad (30)$$

Combining Equation (30) with Equation (26), we could easily obtain:

$$\begin{aligned} (\mathcal{A})_+ &= 4\mathbf{P} \mathbf{P}^T \mathbf{P} + 2\mathbf{P}(\mathbf{S} + \mathbf{S}^T) \mathbf{P}^T \mathbf{P}; \\ (\mathcal{B})_+ &= 2\mathbf{P}(\mathbf{S} + \mathbf{S}^T) + 4\mathbf{P}. \end{aligned} \quad (31)$$

Then the specific formula for updating \mathbf{P} is expressed in the following:

$$\begin{aligned} \mathbf{P} &\leftarrow \mathbf{P} \otimes \{ \mathbf{P}(\mathbf{S} + \mathbf{S}^T) + 2\mathbf{P} \} \\ &\oslash \{ 2\mathbf{P} \mathbf{P}^T \mathbf{P} + \mathbf{P}(\mathbf{S} + \mathbf{S}^T) \mathbf{P}^T \mathbf{P} \}, \end{aligned} \quad (32)$$

or in an element-wise version:

$$\mathbf{P}_{ij} \leftarrow \mathbf{P}_{ij} \frac{[\mathbf{P}(\mathbf{S} + \mathbf{S}^T) + 2\mathbf{P}]_{ij}}{[2\mathbf{P} \mathbf{P}^T \mathbf{P} + \mathbf{P}(\mathbf{S} + \mathbf{S}^T) \mathbf{P}^T \mathbf{P}]_{ij}}, \quad (33)$$

where $i = 1, 2, \dots, C$ and $j = 1, 2, \dots, N$. Inspired by Ref. [30], using Equation (32 or 33) to update \mathbf{P} directly would yield unstable performance, then we adopt the following “strategic multiplicative updates”:

$$\mathbf{P}_{ij} \leftarrow \mathbf{P}_{ij} \{(1 - \lambda) + \lambda \mathbf{Q}\}^\mu \quad (34)$$

where $\mathbf{Q} = \frac{[\mathbf{P}(\mathbf{S} + \mathbf{S}^T) + 2\mathbf{P}]_{ij}}{[2\mathbf{P}\mathbf{P}^T\mathbf{P} + \mathbf{P}(\mathbf{S} + \mathbf{S}^T)\mathbf{P}^T\mathbf{P}]_{ij}}$, λ and μ are set to 0.5 and 0.9 respectively in our experimental parts. Here we call the iterative rule (34) “Strategic Multiplicative Update”. Thus the procedure for optimizing \mathbf{P} is summarized in Algorithm 3.

V. EXPERIMENTS

This section would empirically evaluate the effectiveness of RNSE for clustering on both *synthetic* and *real-world* datasets.

A. Experiments on Synthetic Datasets

The first synthetic dataset we constructed is a 1000×1000 matrix with four 250×250 block matrices diagonally arranged (Fig. 1). The data in each block denotes the affinity of any two points within one cluster while the data outside all blocks denotes noise. The affinity data within each block is randomly generated in the range from 0 to 1, while the noise is randomly generated in the range from 0 to c , which is set as 0.2, 0.4 and 0.8 respectively during the experiments.

Fig. 1 exhibits the original graphs and their corresponding clustering results under different noise (e.g. c) settings. We can see that RNSE overall presents good performances w.r.t. clustering task. Specifically, RNSE successfully learns a structured doubly stochastic matrix with explicit block structures, which divided the data samples into four clear clusters. As the noise increases, the block structure in the original graph blurs, but RNSE is still able to detect the intrinsic structures of the data, which indicates the robustness of the RNSE method for potential practical applications.

The second synthetic dataset is a randomly generated two-moon dataset. There are two clusters with each being a volume of 100 samples distributed in the moon shape (Fig. 2). Here, we tested K-Means [11], NCuts [1], [31] and RNSE on such dataset. Note that in this figure, the color of the two clusters are set to be red and blue, respectively; and the green lines denote the affinity of any two points. Obviously on the whole, Fig. 2 tells us the RNSE’s effectiveness.

More specifically, some analysis could be drawn as follows. First, there are some points split into the wrong cluster w.r.t K-Means (Fig. 2(b)). It’s easy to understand that K-Means mainly deals with ball-like distributed data points, which is obviously not fit for the manifold (e.g. two-moon) data points. Second, From Fig. 2(c), NCuts could well divide the data points into two separate clusters, but the green lines (affinities) between samples are mixed across different clusters, which indicates that the classic spectral clustering methods tends to mis-recognize neighbors. Third, RNSE (Fig. 2(d)), extended from the spectral family, could tell the differences between both classes and neighbors. This implies that RNSE potentially holds stronger abilities than the classic spectral methods (e.g. NCuts) in handling complex datasets.

TABLE I
STATISTICS OF DATASETS.

Datasets	#Samples	#Features	#Classes
diabetes	768	8	2
arcene	900	1000	2
mnist	1000	784	10
alpha_digit	1404	320	36
yeast_uni	1484	1470	10
PCMAC	1943	3289	2
waveform-21	2746	21	3
gisette	7000	5000	2

B. Experiments on Real-world Datasets

Datasets. Eight real-world datasets are selected in the clustering experiments. More specifically, the “diabetes”, “arcene”, “yeast_uni”, “waveform21” and “gisette” are 5 publicly available collections from website³; the “PCMAC” dataset is available from website⁴; while the “mnist” and “alpha-digit” datasets are collected from Sam Roweis’ page⁵. Note that we just select the top-100 samples of each digit (“0”~“9”) in “mnist” dataset for our experiments. The detailed statistics are summarized in Table I.

Competing Methods. To demonstrate the effectiveness of RNSE, we compare it with several popular clustering algorithms, i.e., (1) Canonical K-Means (K-Means) [11]⁶, (2) Principal Component Analysis (PCA) [32]⁷, (3) K-Means clustering in the Low-Rank subspaces (LRR) [33]⁸, (4) Non-negative Matrix Factorization (NMF) [14], [34]⁹, (5) Normalized Cut (NCuts) [1], [31]¹⁰, (6) Structured Doubly Stochastic Matrix (SDS) [17], (7) Clustering with Adaptive Neighbors (CAN) [18] and (8) Constrained Laplacian Rank Algorithm for Graph-based Clustering (CLR) [9]¹¹.

Among these algorithms, NCuts, SDS, CAN, CLR and RNSE are approaches that consider the graph-based structures. PCA and LRR seek the low-rank principal components for data compression. NMF is a model with non-negative constraints and thus could learn additive parts-based components. Note that K-Means directly splits the original high-dimensional data points into clusters; while our RNSE method could also cluster the datasets directly owing to the end-to-end single-stage learning for indicator matrix. However, other methods (excluding K-Means, CAN, CLR and RNSE) mainly conduct two-stage learning for data clustering, i.e., low-dimensional embedding and K-Means clustering.

³<https://archive.ics.uci.edu/ml/datasets.php>

⁴<http://featureselection.asu.edu/datasets.php>

⁵<https://cs.nyu.edu/home/index.html>

⁶<http://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans>

⁷<http://www.cad.zju.edu.cn/home/dengcai/Data/DimensionReduction.html>

html

⁸<https://sites.google.com/site/guangcanliu/>

⁹<http://scikit-learn.org/stable/modules/generated/sklearn.decomposition>

NMF

¹⁰<https://scikit-learn.org/stable/modules/generated/sklearn.cluster>

Spectral Clustering

¹¹SDS, CAN and CLR are available at <http://www.escience.cn/people/fpnjie/index.html>

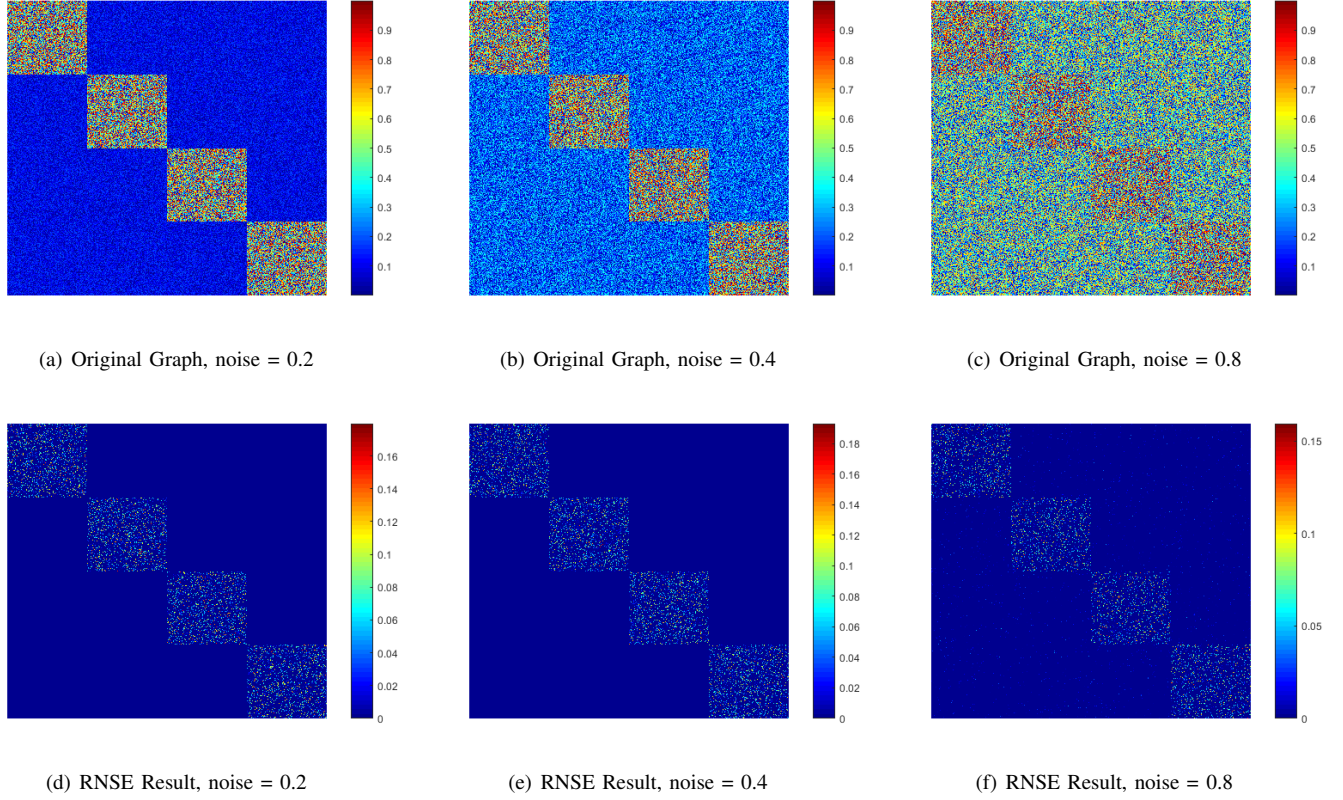


Fig. 1. Clustering results on the *block-diagonal* synthetic dataset.

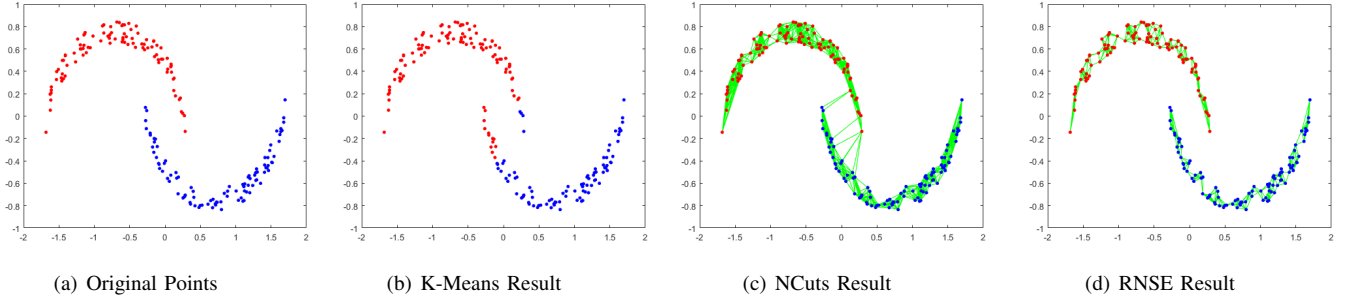


Fig. 2. Clustering results on the *two-moon* synthetic dataset.

Evaluation Metrics. Accuracy (ACC) and Purity are widely accepted and therefore employed here to assess the clustering performance. The higher the performance scores, the better the clustering results, based on the ground-truth class labels presented in the datasets. For more details, please refer to Refs. [35] and [36].

Experimental Settings. In light of the experimental settings, all the baseline methods adopt the best parameter configurations as suggested in their corresponding papers. We use the widely used self-tune Gaussian method [37] to construct the affinity matrix with kernel function $\mathcal{K}(x, y) = \exp\{-(x - y)^2/\gamma\}$ (the value of γ is self-tuned, for not only RNSE but also SDS, CLR, and CAN as suggested in their

corresponding papers).

As to our RNSE approach, we empirically achieve the competitive α and β according to the 6×6 grid searches with $10^{[-3:1:2]}$ and $10^{[-5:1:0]}$, respectively. Generally speaking, the best clustering performances on different datasets correspond to slightly different parameter settings. But around the parameters $\alpha = 1$ and $\beta = 1$, all have achieved competitive experimental results.

In addition, the maximum iterations for Algorithm 2 and Algorithm 3 are set to 20 and 20 respectively, and the convergence precision is configured as 10^{-9} for these two sub-algorithms. Based on such settings, the number of outer cycles for Algorithm 1 is set to 20 for convergence.

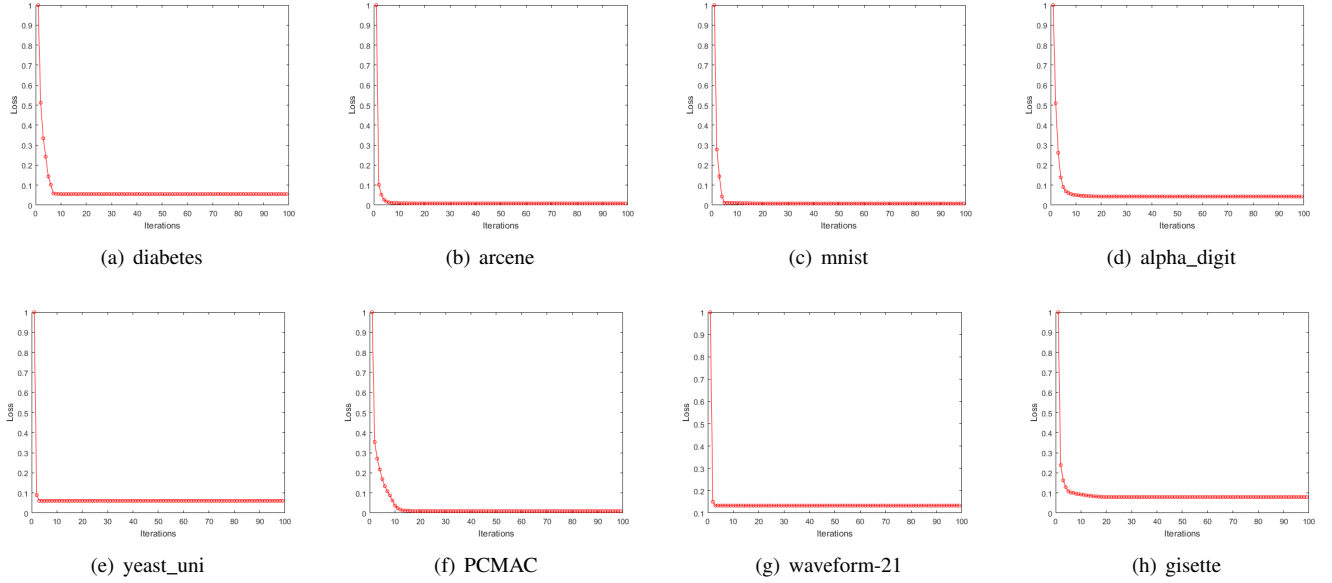


Fig. 3. Convergence curves of RNSE on different datasets: the y -axis is the normalized objective function value (loss) and x -axis is the iteration number.

TABLE II
DIFFERENT COMPETITORS' CLUSTERING PERFORMANCES MEASURED BY ACC (%).

Dataset	K-Means	PCA	LRR	NMF	Ncuts	SDS	CAN	CLR	RNSE
diabetes	51.6 \pm 0.0	51.6 \pm 0.0	52.6 \pm 0.0	52.0 \pm 0.1	51.6 \pm 0.0	51.6 \pm 0.0	48.7 \pm 0.0	64.1 \pm 0.0	64.7 \pm 0.0
arcene	59.0 \pm 0.0	59.0 \pm 0.0	59.0 \pm 0.0	—	59.0 \pm 0.0	59.0 \pm 0.0	40.0 \pm 0.0	50.5 \pm 0.0	62.5 \pm 0.0
mnist	51.0 \pm 0.2	49.3 \pm 0.2	54.7 \pm 0.0	—	56.1 \pm 0.7	53.2 \pm 0.7	55.3 \pm 0.0	56.6 \pm 0.0	64.0 \pm 0.3
alpha_digit	43.1 \pm 1.9	45.6 \pm 1.2	37.8 \pm 0.8	38.9 \pm 1.0	39.9 \pm 1.3	46.4 \pm 1.0	19.8 \pm 0.0	28.9 \pm 0.0	48.1 \pm 0.8
yeast_uni	33.7 \pm 3.9	37.4 \pm 1.2	17.8 \pm 0.1	—	29.4 \pm 0.1	35.4 \pm 0.8	30.8 \pm 0.1	37.4 \pm 0.0	41.3 \pm 0.2
PCMAC	55.4 \pm 0.0	55.3 \pm 0.0	55.7 \pm 0.0	56.8 \pm 0.0	55.5 \pm 0.0	51.6 \pm 0.0	50.6 \pm 0.0	50.3 \pm 0.0	59.3 \pm 0.0
waveform-21	50.7 \pm 0.1	50.7 \pm 0.0	46.1 \pm 0.0	—	50.9 \pm 0.0	51.9 \pm 0.0	47.5 \pm 0.0	50.7 \pm 0.0	52.7 \pm 0.0
gisette	51.5 \pm 0.1	49.3 \pm 0.1	50.6 \pm 0.1	—	92.9 \pm 0.1	93.3 \pm 0.0	91.4 \pm 0.1	93.8 \pm 0.1	94.9 \pm 0.1

Note: “—” denotes that the mixed signed matrices for those datasets are not suitable for the MU algorithms employed by NMF.

TABLE III
DIFFERENT COMPETITORS' CLUSTERING PERFORMANCES MEASURED BY PURITY (%).

Dataset	K-Means	PCA	LRR	NMF	Ncuts	SDS	CAN	CLR	RNSE
diabetes	65.1 \pm 0.0	65.1 \pm 0.0	65.1 \pm 0.0	65.1 \pm 0.0	65.1 \pm 0.0	65.1 \pm 0.0	65.7 \pm 0.0	65.3 \pm 0.0	65.7 \pm 0.0
arcene	59.0 \pm 0.0	59.0 \pm 0.0	59.0 \pm 0.0	—	59.0 \pm 0.0	59.0 \pm 0.0	40.0 \pm 0.0	50.5 \pm 0.0	62.5 \pm 0.0
mnist	56.1 \pm 1.7	55.5 \pm 1.4	59.9 \pm 0.0	—	60.6 \pm 0.7	59.0 \pm 6.8	64.1 \pm 0.0	57.0 \pm 0.0	68.1 \pm 0.3
alpha_digit	46.5 \pm 1.6	48.9 \pm 0.9	41.2 \pm 0.4	41.9 \pm 1.0	42.8 \pm 1.3	49.7 \pm 0.7	19.8 \pm 0.0	33.2 \pm 0.0	50.5 \pm 0.3
yeast_uni	47.1 \pm 1.7	38.8 \pm 1.2	33.6 \pm 0.2	—	31.3 \pm 0.0	37.3 \pm 0.4	46.4 \pm 0.1	46.4 \pm 0.0	47.0 \pm 0.3
PCMAC	55.4 \pm 0.0	55.4 \pm 0.0	55.7 \pm 0.0	56.8 \pm 0.1	55.5 \pm 0.0	51.6 \pm 0.0	50.5 \pm 0.0	50.5 \pm 0.0	59.3 \pm 0.0
waveform-21	53.3 \pm 0.0	53.5 \pm 0.0	47.8 \pm 0.0	—	51.7 \pm 0.0	52.0 \pm 0.0	52.8 \pm 0.0	52.8 \pm 0.0	52.8 \pm 0.0
gisette	69.3 \pm 0.1	67.8 \pm 0.1	50.6 \pm 0.1	—	67.9 \pm 0.1	94.3 \pm 0.0	93.9 \pm 0.1	94.0 \pm 0.1	94.9 \pm 0.1

Clustering Results. In this part, we collect the average clustering results in terms of ACC and Purity for all the algorithms on these datasets and show them in Table II and Table III. Note that for all datasets, α and β in RNSE are both set to 1, we repeat the experiments for 10 times and average the metric values as the final results. Broadly speaking, different clustering approaches perform differently on various datasets. From Table II, we can easily figure out that LRR presents high ACC values on “PCMAC” dataset while performs quite poorly on “yeast_uni” and “alpha_digit” datasets. This phenomenon

also appears in Table III w.r.t. Purity. It’s probably due to the complex structures of “yeast_uni” (the cellular localization sites of proteins) and “alpha_digit” (containing both letters and numbers), which are not well match for the low-rank assumptions. In terms of NCuts, a spectral-based method, always performs quite well on various datasets, which is reasonable because spectral-based algorithms could capture the local structures by keeping the neighborhood similarities and therefore preserve the nonlinear manifolds on complex datasets. However, it’s usually inferior to the best performers,

and this is probably owing to the self-defined similarities and multi-stage learning. In light of other competitors (taking SDS for example), they also display the similar patterns, i.e., yielding high values on some datasets (“yeast_uni” or “waveform-21”), but meanwhile showing poor performance on some other datasets (“alpha_digit” or “PCMAC”). Nevertheless, as one can see clearly from all the experimental results in Table II and Table III, our RNSE method consistently achieves the best or at least comparative performances on all the datasets regarding ACC and Purity. This confirms that by designing an end-to-end single-stage learning paradigm with structured constraints, RNSE could better capture the hidden complex structures and thus learn a well-performed indicator matrix for clustering.

Convergence & Complexity Analysis. The updating rules in Algorithm 1 for minimizing the objective function in the optimization problem (6) are essentially iterative. Here, we investigate their convergence and fastness via experiments. Fig. 3 plots the loss curves of our RNSE on all the selected datasets. In each sub-graph, the y -axis denotes the normalized objective function value¹² and x -axis is the iteration number. Obviously, we could read that the RNSE algorithm converges quickly, usually within 20 iterations.

By the way, it’s also easy to analyze the computational complexities of the two subproblems (Algorithm 2 and 3) of RNSE, which corresponds to $O(MN^2 + CN^2)$ and CN^2 , respectively (usually $M, C \ll N$). Therefore, the RNSE’s overall complexity is square to the number of samples which is much faster than many existing competitors’ complexity (i.e., $O(N^3)$). However, for a larger dataset, say million-scale samples, RNSE is reasonably to loss efficiency which posits a big challenge for future work.

VI. CONCLUSION

To sum up, the main contributions of this paper are twofold: First, we prove the advantage of performing spectral-style clustering in an end-to-end single-stage fashion where the similarity matrix is not prefixed but learned adaptively from the data. Second, we show that the difficult optimization problem of our proposed RNSE technique can be decomposed into two subproblems (i.e., metric projection and orthogonal symmetric non-negative matrix factorization) and then solved by successive alternating projection and strategic multiplicative update respectively.

REFERENCES

- [1] A. Y. Ng, M. I. Jordan, and Y. Weiss, “On spectral clustering: Analysis and an algorithm,” *NeurIPS*, pp. 849–856, 2002.
- [2] Z. Zhang, Y. Zhang, S. Li, G. Liu, M. Wang, and S. Yan, “Robust unsupervised flexible auto-weighted local-coordinate concept factorization for image clustering,” *ICASSP*, pp. 2092–2096, 2019.
- [3] U. Shaham, K. Stanton, H. Li, B. Nadler, R. Basri, and Y. Kluger, “Spectralnet: Spectral clustering using deep neural networks,” *arXiv preprint arXiv:1801.01587*, 2018.
- [4] P. Ji, T. Zhang, H. Li, M. Salzmann, and I. Reid, “Deep subspace clustering networks,” *NeurIPS*, pp. 24–33, 2017.
- [5] M. Belkin and P. Niyogi, “Laplacian eigenmaps and spectral techniques for embedding and clustering,” *NeurIPS*, pp. 585–591, 2002.
- [6] R. Zass and A. Shashua, “Doubly stochastic normalization for spectral clustering,” *NeurIPS*, pp. 1569–1576, 2007.
- [7] D. Cai, X. He, and J. Han, “Spectral regression: A unified approach for sparse subspace learning,” *ICDM*, pp. 73–82, 2007.
- [8] X. Chen and D. Cai, “Large scale spectral clustering with landmark-based representation,” *AAAI*, pp. 313–318, 2011.
- [9] F. Nie, X. Wang, M. I. Jordan, and H. Huang, “The constrained laplacian rank algorithm for graph-based clustering,” *AAAI*, pp. 1969–1976, 2016.
- [10] F. Nie, W. Zhu, and X. Li, “Unsupervised large graph embedding,” *AAAI*, pp. 2422–2428, 2017.
- [11] S. P. Lloyd, “Least squares quantization in PCM,” *IEEE Trans. Information Theory*, pp. 129–136, 1982.
- [12] L. Zhang, Q. Zhang, B. Du, J. You, and D. Tao, “Adaptive manifold regularized matrix factorization for data clustering,” *IJCAI*, pp. 3399–3405, 2017.
- [13] X. He and P. Niyogi, “Locality preserving projections,” *NeurIPS*, pp. 153–160, 2004.
- [14] D. D. Lee and H. S. Seung, “Learning the parts of objects by non-negative matrix factorization,” *Nature*, pp. 788–791, 1999.
- [15] H. Ma, W. Zhao, Q. Tan, and Z. Shi, “Orthogonal nonnegative matrix tri-factorization for semi-supervised document co-clustering,” *PAKDD*, pp. 189–200, 2010.
- [16] D. Cai, X. He, J. Han, and T. S. Huang, “Graph regularized nonnegative matrix factorization for data representation,” *TPAMI*, pp. 1548–1560, 2011.
- [17] X. Wang, F. Nie, and H. Huang, “Structured doubly stochastic matrix for graph based clustering,” *KDD*, pp. 1245–1254, 2016.
- [18] F. Nie, X. Wang, and H. Huang, “Clustering and projected clustering with adaptive neighbors,” *KDD*, pp. 977–986, 2014.
- [19] F. Wang, P. Li, and A. C. Konig, “Learning a bi-stochastic data similarity matrix,” *ICDM*, pp. 551–560, 2010.
- [20] F. R. K. Chung, “Spectral graph theory,” *CBMS Regional Conference Series in Mathematics*, No. 92, American Mathematical Society, 1997.
- [21] K. Fan, “On a theorem of weyl concerning eigenvalues of linear transformations,” *National Academy of Sciences of the United States of America*, pp. 652–655, 1949.
- [22] D. D. Lee and H. S. Seung, “Algorithms for non-negative matrix factorization,” *NeurIPS*, pp. 556–562, 2001.
- [23] L. D., “Optimization by vector space methods,” *John Wiley and Sons, Inc., New York*, 1969.
- [24] J. P. Boyle and R. L. Dykstra, “A method for finding projections onto the intersection of convex sets in hilbert spaces,” *Lecture Notes in Statistics*, pp. 28–47, 1986.
- [25] M. Raydan and R. Escalante, “Alternating projection methods,” *Fundamentals of Algorithms Society for Industrial and Applied Mathematics*, 2011.
- [26] M. Tam, “The method of alternating projections,” *Doctoral dissertation, University of Newcastle, Australia*, 2012.
- [27] Z. Guan, L. Zhang, J. Peng, and J. Fan, “Multi-view concept learning for data representation,” *TKDE*, vol. 27, no. 11, pp. 3016–3028, 2015.
- [28] Y. Wang and Y. Zhang, “Nonnegative matrix factorization: A comprehensive review,” *TKDE*, vol. 25, no. 6, pp. 1336–1353, 2013.
- [29] J. Yoo and S. Choi, “Orthogonal nonnegative matrix factorization: Multiplicative updates on stiefel manifolds,” *IDEAL*, pp. 140–147, 2008.
- [30] Z. He, S. Xie, R. Zdunek, G. Zhou, and A. Cichocki, “Symmetric nonnegative matrix factorization: Algorithms and applications to probabilistic clustering,” *IEEE Trans. Neural Networks*, vol. 22, no. 12, pp. 2117–2131, 2011.
- [31] J. Shi and J. Malik, “Normalized cuts and image segmentation,” *TPAMI*, pp. 888–905, 2000.
- [32] S. Wold, K. Eebensen, and P. Geladt, “Principal component analysis,” *Chemometrics and Intelligent Laboratory Systems*, pp. 37–52, 1987.
- [33] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, “Robust recovery of subspace structures by low-rank representation,” *TPAMI*, pp. 171–184, 2013.
- [34] W. Xu, X. Liu, and Y. Gong, “Document clustering based on non-negative matrix factorization,” *SIGIR*, pp. 267–273, 2003.
- [35] J. Huang, F. Nie, H. Huang, and C. H. Q. Ding, “Robust manifold nonnegative matrix factorization,” *TKDD*, pp. 1–21, 2013.
- [36] F. Nie, D. Xu, and X. Li, “Initialization independent clustering with actively self-training method,” *IEEE Trans. Systems, Man, and Cybernetics*, pp. 17–27, 2012.
- [37] L. Z. Manor and P. Perona, “Self-tuning spectral clustering,” *NeurIPS*, pp. 1601–1608, 2004.

¹²Each iteration’s value is divided by the first iteration’s value.