2019 IEEE International Conference on Big Data (Big Data)

Finding Stable Clustering for Noisy Data via Structure-Aware Representation

Huiyuan Chen
Department of Computer and Data Sciences
Case Western Reserve University
hxc501@case.edu

Jing Li
Department of Computer and Data Sciences
Case Western Reserve University
jingli@cwru.edu

***Abstract*—Clustering is one of the most prominent topics in machine learning. A multitude of clustering methods have been proposed, among which the spectral clustering has attracted much attention. However, in practice, spectral clustering is highly sensitive to noise data and a post-processing step (e.g., k-means for eigenvectors) is often required to obtain clustering indicators, which may be not optimal. Also, it does not scale well to large-scale data due to its eigen-decomposition procedures.**

Here we propose a structure-aware clustering model to address those issues. To achieve our goal, a high-quality affinity matrix is extracted from the original noisy data by a sparse additive decomposition, which is used to approximate the ideal clustering structure. We then jointly learn the high-quality affinity matrix as well as the spectral embedding in a unified model— thus, being robust to noise and obtaining the optimal clustering indicators without any post-processing steps. We further improve the clustering stability by considering the Laplacian eigengap of the affinity matrix. We show that the larger the Laplacian eigengap, the more stable the clustering results. We introduce a speedup strategy to effectively compute eigenvectors of large matrices. Experimental results demonstrate that the proposed model outperforms existing approaches for noisy data.

***Index Terms*—Structure-Aware Clustering, Noisy Data, Eigengap, Doubly Stochastic Matrix, Graph Clustering**

I. INTRODUCTION

Clustering is an important research topics in machine learning, which tries to explore the inherent structure of data. Spectral clustering, exploiting the pairwise relationships between data instances in spectral domains, has shown great promise in many applications such as graph analysis [1], [2], bioinformatics [3], and computer vision. In addition, spectral clustering has a natural connection to normalized cut, matrix factorization, and random walk [1]. Easy implementation and superior performance make spectral clustering become the first choice for many clustering problems.

Spectral clustering embeds the graph-based similarity/affinity matrix into a low-rank dimensional space before performing k-means clustering [2]. The low-rank space is spanned by the first k eigenvectors corresponding to the k smallest eigenvalues of the (normalized) graph Laplacian matrix. Thus the performance of spectral clustering is largely depended on the input affinity matrix. Any noise or variations to the input affinity matrix may impair the final clustering performance [4], [5].

978-1-7281-0858-2/19/$31.00 ©2019 IEEE

To enhance clustering performance, several pre-processing strategies have been made to build a better affinity matrix as input from original data [6], [7], [8], [9]. One successful attempt is the doubly stochastic normalization of the affinity matrix, which achieves superior clustering performance [7]. However, the doubly stochastic matrix cannot handle noise in the data. The designed doubly stochastic matrix is expected to be very closer to original affinity matrix (e.g., using relative-entropy or Frobenius norm minimization), but the original affinity matrix can be corrupted by considerable noise, leading to a noisy doubly stochastic matrix. Moreover, even with the doubly stochastic matrix approximation, the clusters are still not obvious and post-processing like k-means is often required to get the final cluster indicators, which is sensitive to the initializations [1], [2], [9].

To address these challenges, we propose structure-aware SCAN, a **S**pectral **C**lustering algorithm for noisy data via **A**ugmenting eige**N**gap. The core idea is that the input affinity matrix can be decomposed into two matrices: a high-quality affinity matrix and an irrelevant noise matrix. Rather than directly operating on the original affinity matrix, we learn a better doubly stochastic matrix from the high-quality affinity matrix, which explicitly reflects the true data structure. Inspired by the rank minimization [10], [11], we impose a rank constraint on the Laplacian matrix of the desirable doubly stochastic matrix, thereby guaranteeing that the number of connected components in the data is exactly k, i.e., the number of clusters. Moreover, unlike traditional spectral clustering algorithm requiring two stages, we simultaneously learn the doubly stochastic matrix as well as spectral embedding in a unified model. We show that as long as the doubly stochastic matrix is optimal, the cluster indicators can be directly obtained without any post-processing steps. To be robust against noise, we further strengthen the clustering stability via augmenting the Laplacian eigengap. The underlying intuition is that a clustering structure is stable if small distortions/noises in the data do not affect its eigenspace that is spanned by the primary k eigenvectors [12], [2]. Mathematically, we turn our attention to regularized spectral learning [12], which indicates that the larger the Laplacian eigengap, the more stable the clustering results. We model the problem of robust spectral clustering as an optimization problem and introduce an efficient optimization algorithm to solve it. Our method

can heuristically find more stable clusters for noisy data. We also report an empirical study on both synthetic and benchmark datasets. The experimental results demonstrate the effectiveness of our proposed method. Our contributions are summarized as follows:

- We propose a structure-aware clustering model for noisy data. Specifically, we decompose the original affinity matrix into a high-quality affinity matrix and an irrelevant error matrix, in which high-quality affinity matrix should better reflect the true data structure.
- We learn a structured doubly stochastic matrix to approximate our high-quality affinity matrix, in which the clustering indicators can be obtained without post-processing steps. To find stable clusters, we further consider the Laplacian eigengap to refine the eigenspace.
- We formulate the clustering problem as a optimization problem and propose an effective algorithm to solve the optimization problem. Extensive experiments on both both synthetic and benchmark datasets demonstrate the effectiveness of our model.

The rest of the paper is organized as follows. Section II reviews some related work. Sections III defines the problem. IV and V propose our optimization problem with an efficient optimization algorithm to solve it. Section VI reports experimental results. Section VII concludes the paper.

II. RELATED WORK

Spectral clustering, which aims to partition data into several groups, has been studied extensively. The literatures [1], [2] provide a very detailed overview of spectral clustering. Here, we review several robust spectral learning methods according to different principles.

The only input to spectral clustering is the graph-based affinity matrix, which is thus largely responsible for the final clustering results. However, constructing pairwise similarities for spectral clustering is challenging because data can be noisy, incomplete, heterogeneous, and without any prior knowledge [1]. Researchers have thus attempted to build a better affinity matrix for spectral clustering [13], [14], [6], [15]. For example, Zelnik et al. [13] proposed a local similarity scaling approach to learn an adaptive scaling factor in the Gaussian kernel when computing the affinity between two data instances. It worked well on noise-free data but was susceptible to noisy inputs. To alleviate this issue, Correa et al. [15] introduced a new method to improve the robustness by using empty regions and a diffusion based local scaling approach. Similarly, Premachandran et al. [14] developed a k-NN neighborhood selection method to obtain strong local neighborhoods. They made use of the consensus information from multiple k-NNs to discard noisy edges so that the resulting affinity matrix is more robust against noisy data.

In addition to local scaling approaches, doubly stochastic normalization of affinity matrix is another promising strategy to enhance the clustering results [7], [9], [8], [16]. Ron et al. [7] first showed that the doubly stochastic normalization of an affinity matrix was intimately related to kernel k-means. By doing so, they obtained an improved affinity matrix providing superior clustering performance. Zhang et al. [16] presented a decomposition method for clustering, which surprisingly leaded to a doubly-stochastic approximating matrix. Such matrix had been shown to be desired for balanced graph cuts. Recently, Douik et al. [17] developed a Riemannian optimization framework for solving clustering problems on the set of symmetric doubly stochastic matrices, which produced satisfactory for graph clustering. Although the doubly stochastic matrix can improve clustering performance, the clustering structures in the doubly stochastic matrix are still unknown. The post-processing like k-means is often required to get the final cluster indicators.

To overcome this problem, Nie et al. [11] and Wang et al. [9] proposed two convex models to learn the structured doubly stochastic matrix by imposing low-rank constraints on the graph Laplacian matrix. Their structured doubly stochastic matrix could explicitly uncover the clustering structure without post-processing steps. Jiwoong et al. [8] further introduced a novel method to build a doubly stochastic affinity matrix by incorporating Davis-Kahan theorem of matrix perturbation theory such that every cluster was well retained in its eigenspace. Furthermore, considerable efforts have been made to find more stable clusters [12], [18]. For example, Meila et al. [12] first investigated the clustering stability by considering the Laplacian eigengap in the regularized spectral learning methods. They further proved that a larger Laplacian eigengap corresponded to better clustering stability. Inspired by this idea, Juhua et al. [18] developed a multi-clustering method to obtain a certain number of stable clusters by directly maximizing the Laplacian eigengap.

Other methods for robust spectral clustering include random forest-based affinity construction [6], Laplacian smoothing [19], and random binning features [20]. However, those method require post-process steps (e.g., k-means) to obtain the final clustering indicators, thus are sensitive to the initializations. Recently, several studies jointly learn the affinity matrix as well as the spectral embedding in one unified model [5], [11], [10], [21]. However, their eigenspaces are still sensitive to noise, resulting in suboptimal performance.

III. PROBLEM FORMULATION

Notations: Throughout this paper, matrices are denoted as uppercase letters (e.g., $\mathbf{A}$). $\lambda(\mathbf{A})$ is a vector containing all eigenvalues of $\mathbf{A}$ in decreasing order (e.g., $\lambda_1(\mathbf{A}) \geq \lambda_2(\mathbf{A}) \geq \cdots$) and $\lambda_i(\mathbf{A})$ is the i-th largest eigenvalue of A; $\sigma(\mathbf{A})$ denotes all singular values of $\mathbf{A}$ in a similar way. $\mathbb{1}$ is a vector whose elements are all one.

Problem Setup: Given a set of n data points $[\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n] = \mathbf{X}^{n\times d}$, with $\boldsymbol{x}_i \in \mathbb{R}^d$. The spectral clustering algorithm typically constructs a graph affinity matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$, where $\mathbf{A}_{ij}$ represents the similarity between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. A common way to construct such affinity matrix is the Gaussian kernel:

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{2\sigma^2}\right) \tag{1}$$

where σ is the bandwidth parameter. The goal of spectral clustering is to use $\mathbf{A}$ to partition data points into k different clusters. Therefore, the graph affinity matrix is largely responsible for the performance.

Doubly stochastic normalization of affinity matrix often induces a significant performance boost in practice [7], [9], [17]. For the doubly stochastic normalization problem, a new affinity matrix $\mathbf{S}$ is learned from the graph affinity matrix $\mathbf{A}$ by minimizing the following objective function:

$$\min_{\mathbf{S}} \quad \|\mathbf{S}-\mathbf{A}\|_F^2 \qquad \text{s.t.} \quad \mathbf{S}\geq 0,\ \mathbf{S}=\mathbf{S}^T,\ \mathbf{S}\mathbb{1}=\mathbb{1} \tag{2}$$

where $\|\cdot\|_F$ denotes the Frobenius norm; $\mathbf{S}\in\mathbb{R}^{n\times n}$ is a doubly stochastic matrix with the following constraints: non-negativity, symmetry, and row/column-wise sum being one. However, there are two drawbacks with this approach: (i) the doubly stochastic matrix is highly sensitive to noisy. The doubly stochastic matrix $\mathbf{S}$ is expected to get closer to matrix $\mathbf{A}$, but the original affinity matrix $\mathbf{A}$ may be corrupted by noise, leading to a noisy matrix $\mathbf{S}$. (ii) the cluster structures are not obvious from the matrix $\mathbf{S}$ and a post-processing step (e.g., k-means) is needed to uncover the clustering indicators.

IV. STRUCTURE-AWARE SPECTRAL CLUSTERING

To address these two challenges, we aim to learn a new doubly stochastic matrix that can handle noise and maintain the data clusters at the same time.

A. Noise-free Affinity Matrix

We assume that the graph affinity matrix $\mathbf{A}$ is not perfect but may contain certain level of noise. Inspired by the robust principal component analysis [22], we naturally decompose the original graph affinity matrix $\mathbf{A}$ into two parts: a high-quality affinity matrix $\mathbf{A^h}$ and an error matrix $\mathbf{A^e}$:

$$\mathbf{A}=\mathbf{A^h}+\mathbf{A^e} \tag{3}$$

The high-quality affinity matrix $\mathbf{A^h}$ should perfectly reveal the true memberships of the data clusters. Meanwhile, the irrelevant noise matrix $\mathbf{A^e}$ is expected to be relatively sparse. If not, the noise will dominant the affinity matrix and a reasonable clustering cannot be detected effectively.

Based on these assumptions, instead of approximating the original affinity matrix $\mathbf{A}$, we learn the doubly stochastic matrix $\mathbf{S}$ from the high-quality affinity matrix $\mathbf{A^h}$. The Eq. (2) can then be improved as follows:

$$\begin{aligned}\min_{\mathbf{S},\mathbf{A^h},\mathbf{A^e}} \quad & \|\mathbf{S}-\mathbf{A^h}\|_F^2+\alpha\|\mathbf{A^e}\|_0 \\ \text{s.t.} \quad & \mathbf{S}\geq 0,\ \mathbf{S}=\mathbf{S}^T,\ \mathbf{S}\mathbb{1}=\mathbb{1},\ \mathbf{A}=\mathbf{A^h}+\mathbf{A^e}\end{aligned} \tag{4}$$

where α controls the sparsity of error matrix $\mathbf{A^e}$ by using l_0-norm.

B. Structure-aware Representation

In the ideal case, the doubly stochastic matrix $\mathbf{S}$ in Eq. (4) should be a *block-diagonal* structured matrix with proper permutation [10], [11], [9]. The reason is that the ideal clustering structure of the data have exactly k (the number of the clusters) connected components and only data instances from the same cluster are connected to each other in the graph affinity matrix $\mathbf{S}$. Given the graph affinity matrix $\mathbf{S}$ and its corresponding Laplacian matrix $\mathbf{L}_S=\mathbf{D}_S-\mathbf{S}$, where $\mathbf{D}_S$ is a diagonal matrix whose i-th diagonal element is $\sum_j \mathbf{S}_{ij}$, we have the following Lemma 1 from spectral graph theory [23]:

Lemma 1. *The multiplicity k of the eigenvalue zero of the Laplacian matrix $\mathbf{L}_S$ is equal to the number of connected components in the graph.*

According to Lemma 1, if the matrix $\mathbf{S}$ has exactly k blocks along with diagonal, then the first smallest k eigenvalues of its Laplacian matrix $\mathbf{L}_S$ are zeros. Suppose $\lambda_i(\mathbf{L}_S)$ is the i-th largest eigenvalue of $\mathbf{L}_S$, we have $\lambda_i(\mathbf{L}_S)\geq 0$ since $\mathbf{L}_S$ is positive semi-definite. The Eq. (4) can be rewritten as:

$$\begin{aligned}\min_{\mathbf{S},\mathbf{A^h},\mathbf{A^e}} \quad & \|\mathbf{S}-\mathbf{A^h}\|_F^2+\alpha\|\mathbf{A^e}\|_0+\beta\sum_{i=n-k+1}^{n}\lambda_i(\mathbf{L}_S) \\ \text{s.t.} \quad & \mathbf{S}\geq 0,\ \mathbf{S}=\mathbf{S}^T,\ \mathbf{S}\mathbb{1}=\mathbb{1},\ \mathbf{A}=\mathbf{A^h}+\mathbf{A^e}\end{aligned} \tag{5}$$

when $\beta\to+\infty$, the optimal solution $\mathbf{S}$ to the Problem (5) will make term $\sum_{i=n-k+1}^{n}\lambda_i(\mathbf{L}_S)=0$, in which a k-block diagonal matrix $\mathbf{S}$ will be satisfied. Under this condition, we can directly partition the data into k clusters based on the structured $\mathbf{S}$ without post-processing steps [11]. According to the Ky Fan's Theorem [24], we further have

$$\sum_{i=n-k+1}^{n}\lambda_i(\mathbf{L}_S)=\min_{\mathbf{F}\in\mathbb{R}^{n\times k},\mathbf{F}^T\mathbf{F}=\mathbf{I}}\mathrm{Tr}(\mathbf{F}^T\mathbf{L}_S\mathbf{F})$$

Therefore, Eq. (5) can be converted to a much easier objective function with auxiliary variable $\mathbf{F}$ (as shown in Eq. (6)).

In addition, in the ideal case the probability of a certain node to correlate with nodes in the same cluster should be the same, which can be regarded as a prior knowledge of structured matrix $\mathbf{S}$. We thus add another regularized term $\gamma\|\mathbf{S}\|_F^2$ to the Eq. (5), where a large parameter γ forces the elements in each block of matrix $\mathbf{S}$ to be the same [11].

C. Clustering Stability

Beyond the structured property, we also hope to recover an optimal affinity matrix $\mathbf{S}$ that is stable for clustering tasks. A clustering is said to be stable if small distortions/noises on data do not affect its eigenspace of $\mathbf{L}_S$ [12], [2]. Fortunately, the eigenspace is only spanned by the first k eigenvectors of the Laplacian matrix $\mathbf{L}_S$ corresponding to the k smallest eigenvalues. We next show how to find a stable clustering based on the matrix perturbation theory.

Note that $\mathbf{L}_S=\mathbf{I}-\mathbf{S}$[1], it is easy to verify that the first k eigenvectors of matrix $\mathbf{S}$ corresponding to the k largest eigenvalues are identical to the first k eigenvectors of $\mathbf{L}_S$ corresponding to the k smallest eigenvalues. Now we have a nice property of stability on our doubly stochastic matrix $\mathbf{S}$.

[1]We have $\mathbf{D}_S=I$ since $\mathbf{S}$ is a doubly stochastic matrix.

Lemma 2. ***(Stability** [18]): Given a doubly stochastic affinity matrix* $\mathbf{S}$*, if the eigengap* $\Delta = \lambda_k(\mathbf{S}) - \lambda_{k+1}(\mathbf{S})$ *is large enough, then the top* k *eigenvectors of* $\mathbf{S}_{perb} = \mathbf{S} + \epsilon$ *corresponding to the* k *largest eigenvalues are the sames as those of* $\mathbf{S}$*, where* ϵ *a symmetric perturbation matrix of small spectral norm* $\|\epsilon\|_2$.

We know that the top k largest eigenvalues of $\mathbf{S}$ are all one since the first k smallest eigenvalues of $\mathbf{L}_S$ are all zero. According to Lemma 2, the assumption that eigengap Δ being large is exactly the assumption that λ_{k+1} be bounded away from 1. To obtain stable clustering, we can thus add one further constraint $\lambda_{k+1}(\mathbf{S}) \leq 1 - \delta$ in our Eq. (5), where δ is the eigengap within $(0, 1)$.

D. The SCAN Model

To better uncover the data clusters, we propose SCAN model for spectral clustering with full benefit of the noise-free, structured and stable affinity matrix. Formally, we obtain a unified objective function as following:

$$\begin{aligned} \min_{\mathbf{S},\mathbf{F},\mathbf{A^h},\mathbf{A^e}} \quad & \|\mathbf{S}-\mathbf{A^h}\|_F^2+\alpha\|\mathbf{A^e}\|_0+\beta\,\mathrm{Tr}(\mathbf{F}^T\mathbf{L}_S\mathbf{F})+\gamma\|\mathbf{S}\|_F^2 \\ \text{s.t.} \quad & \mathbf{S}\geq 0,\ \mathbf{S}=\mathbf{S}^T,\ \mathbf{S}\mathbb{1}=\mathbb{1},\ \mathbf{F}^T\mathbf{F}=\mathbf{I} \\ & \mathbf{L}_S=\mathbf{I}-\mathbf{S},\ \mathbf{A}=\mathbf{A^h}+\mathbf{A^e},\ \lambda_{k+1}(\mathbf{S})\leq 1-\delta \end{aligned} \tag{6}$$

We next propose an efficient algorithm to solve our model.

V. OPTIMIZATION ALGORITHM

Problem (6) is challenging to solve due to the non-convex l_0-norm. We replace with l_1-norm on $\mathbf{A^e}$ since l_1-norm is the convex envelope of l_0-norm. Furthermore, the singular values of $\mathbf{S}$ are equivalent to its eigenvalues i.e., $\sigma(\mathbf{S}) = \lambda(\mathbf{S})$, since matrix $\mathbf{S}$ is symmetric and positive semi-definite. We thus focus on the singular value of $\mathbf{S}$ due to its easy implementation.

A. Learning Algorithm

We solve Eq. (6) via Augmented Lagrange Multiplier (ALM) method [25]. First, we introduce the auxiliary variables $\mathbf{L}$ and $\mathbf{M}$ such that $\mathbf{L} = \mathbf{I} - \mathbf{S}$ and $\mathbf{S} = \mathbf{M}$, the Eq. (6) becomes:

$$\begin{aligned} \min_{\mathbf{S},\mathbf{F},\mathbf{L},\mathbf{A^h},\mathbf{A^e}} \quad & \|\mathbf{S}-\mathbf{A^h}\|_F^2+\alpha\|\mathbf{A^e}\|_1+\beta\,\mathrm{Tr}(\mathbf{F}^T\mathbf{L}\mathbf{F})+\gamma\|\mathbf{S}\|_F^2 \\ \text{s.t.} \quad & \mathbf{S}\geq 0,\ \mathbf{S}=\mathbf{S}^T,\ \mathbf{S}\mathbb{1}=\mathbb{1},\ \mathbf{F}^T\mathbf{F}=\mathbf{I},\ \mathbf{L}=\mathbf{I}-\mathbf{S} \\ & \mathbf{A}=\mathbf{A^h}+\mathbf{A^e},\ \sigma_{k+1}(\mathbf{M})\leq 1-\delta,\ \mathbf{S}=\mathbf{M} \end{aligned} \tag{7}$$

The objective of Eq. (7) is much easier to solve since the term $\beta\,\mathrm{Tr}(\mathbf{F}^T\mathbf{L}\mathbf{F})$ is now independent of $\mathbf{S}$, and the eigenvalue constraint only involves matrix $\mathbf{M}$. By using augmented Lagrangian function, the above problem is equivalent to:

$$\begin{aligned} \min_{\mathbf{S},\mathbf{F},\mathbf{L}\mathbf{A^h},\mathbf{A^e}} & \|\mathbf{S}-\mathbf{A^h}\|_F^2+\alpha\|\mathbf{A^e}\|_1+\beta\,\mathrm{Tr}(\mathbf{F}^T\mathbf{L}\mathbf{F})+\gamma\|\mathbf{S}\|_F^2+ \\ & \langle\mathbf{\Lambda},\mathbf{L}-\mathbf{I}+\mathbf{S}\rangle+\langle\mathbf{\Theta},\mathbf{A}-\mathbf{A^h}-\mathbf{A^e}\rangle+\langle\mathbf{\Sigma},\mathbf{S}-\mathbf{M}\rangle+ \\ & \frac{\mu}{2}(\|\mathbf{L}-\mathbf{I}+\mathbf{S}\|_F^2+\|\mathbf{A}-\mathbf{A^h}-\mathbf{A^e}\|_F^2+\|\mathbf{S}-\mathbf{M}\|_F^2) \\ \text{s.t.} \quad & \mathbf{S}\geq 0,\ \mathbf{S}=\mathbf{S}^T,\ \mathbf{S}\mathbb{1}=\mathbb{1},\ \mathbf{F}^T\mathbf{F}=\mathbf{I},\ \sigma_{k+1}(\mathbf{M})\leq 1-\delta \end{aligned} \tag{8}$$

where $\mathbf{\Lambda}, \mathbf{\Theta}, \mathbf{\Sigma} \in \mathbb{R}^{n\times n}$ are the Lagrange multipliers and μ is the penalty parameter for Eq. (8). By using ALM, we optimize one variable while fixing others and this procedure repeats until convergence.

1) *Optimize* $\mathbf{S}$*:* The Eq. (8) with respect to $\mathbf{S}$ is defined as:

$$\begin{aligned} \min_{\mathbf{S}} \quad & \|\mathbf{S}-\mathbf{A^h}\|_F^2+\gamma\|\mathbf{S}\|_F^2+\frac{\mu}{2}\|\mathbf{L}-\mathbf{I}+\mathbf{S}+\frac{1}{\mu}\mathbf{\Lambda}\|_F^2+ \\ & \frac{\mu}{2}\|\mathbf{S}-\mathbf{M}+\frac{1}{\mu}\mathbf{\Sigma}\|_F^2 \\ \text{s.t.} \quad & \mathbf{S}\geq 0,\ \mathbf{S}=\mathbf{S}^T,\ \mathbf{S}\mathbb{1}=\mathbb{1} \end{aligned} \tag{9}$$

Let $\widehat{\mathbf{T}} = \frac{1}{\mu+\gamma+1}(\mathbf{A^h} + \frac{\mu}{2}(\mathbf{I} - \mathbf{L} - \frac{1}{\mu}\mathbf{\Lambda} + \mathbf{M} - \frac{1}{\mu}\mathbf{\Sigma}))$, then the Eq. (9) can be simplified to:

$$\min_{\mathbf{S}} \quad \|\mathbf{S}-\widehat{\mathbf{T}}\|_F^2 \quad \text{s.t.} \quad \mathbf{S}\geq 0,\ \mathbf{S}=\mathbf{S}^T,\ \mathbf{S}\mathbb{1}=\mathbb{1} \tag{10}$$

Above objective function is similar to Eq. (2) and can be effectively solved by Von Neumann's successive projection learning algorithm [7].

2) *Optimize* $\mathbf{L}$*,* $\mathbf{A^h}$ *and* $\mathbf{A^e}$*:* Similarly, we can update the variables $\mathbf{L}$, $\mathbf{A^h}$ and $\mathbf{A^e}$ using following closed-form solution in each iteration:

$$\mathbf{L} = \mathbf{I} - \mathbf{S} - \frac{1}{\mu}\mathbf{\Lambda} - \frac{\beta}{\mu}\mathbf{F}\mathbf{F}^T \tag{11}$$

$$\mathbf{A^h} = \frac{1}{\mu+2}(2\mathbf{S} + \mu\mathbf{A} + \mathbf{\Theta} - \mu\mathbf{A^e}) \tag{12}$$

$$\mathbf{A^e} = \mathcal{S}_{\alpha\mu^{-1}}(\mathbf{A} - \mathbf{A^h} + \mu^{-1}\mathbf{\Theta}) \tag{13}$$

where $\mathcal{S}_\tau$: $\mathbb{R} \to \mathbb{R}$ denote the shrinkage operator, which is defined as $\mathcal{S}_\tau[x] = \mathrm{sgn}(x)\max(|x|-\tau, 0)$, where $\mathrm{sgn}(\cdot)$ is the Sign function. It is extended to a matrix by applying the function to each element [25].

3) *Optimize* $\mathbf{F}$*:* The Eq. (8) with respect to $\mathbf{F}$ becomes

$$\min_{\mathbf{F}\in\mathbb{R}^{n\times k},\mathbf{F}^T\mathbf{F}=\mathbf{I}} \mathrm{Tr}(\mathbf{F}^T\mathbf{L}\mathbf{F}) \tag{14}$$

The optimal solution can be obtained by the k eigenvectors of $\mathbf{L}$ corresponding to the k smallest eigenvalues.

4) *Optimize* $\mathbf{M}$*:* The optimal $\mathbf{M}$ can be obtained by solving:

$$\min_{\mathbf{M}} \quad \|\mathbf{M}-\mathbf{S}-\frac{1}{\mu}\mathbf{\Sigma}\|_F^2 \quad \text{s.t.} \quad \sigma_{k+1}(\mathbf{M}) \leq 1-\delta \tag{15}$$

Let $\widetilde{\mathbf{S}} = \mathbf{S} + \frac{1}{\mu}\mathbf{\Sigma}$. According to Theorem 1 in Appendix, the optimal solution is

$$\mathbf{M} = \mathbf{U}\,\mathrm{Diag}(\sigma(\mathbf{M}))\mathbf{V}^T \tag{16}$$

$$\sigma_i(\mathbf{M}) = \begin{cases} 1 & \text{if } 1 \leq i \leq k \\ \min(1-\delta, \sigma_i(\widetilde{\mathbf{S}})) & \text{if } k+1 \leq i \leq n \end{cases}$$

where $\mathbf{U}$ and $\mathbf{V}$ are the left and right orthonormal matrices in the SVD of $\widetilde{\mathbf{S}}$.

Our algorithm to solve objective function in Eq. (6) is outlined in Algorithm 1. We obtain the structured doubly stochastic matrix $\mathbf{S}$, and optimal matrix $\mathbf{F}$ from Algorithm 1.

With the block diagonal structure of $\mathbf{S}$, we can immediately obtain clustering indicators without any post-processing steps. In other words, the optimal solution $\mathbf{F}$ is formed by the first k eigenvectors of $\mathbf{L}$ (or $\mathbf{L}_S$) in Eq. (14). We can thus directly use the optimal $\mathbf{F}$ to get the final clustering results without post-processing, which is normally required by traditional spectral clustering algorithms [11].

Algorithm 1: SCAN

Input: affinity matrix $\mathbf{A}$, regularized parameters α, β and γ, constant δ, the number of clusters k.

Initialize $\mathbf{S} = \mathbf{A}$, $\mathbf{L} = \mathbf{I} - \mathbf{A}$. Randomly initialize matrix $\mathbf{F}$, $\mathbf{A^h}$, $\mathbf{A^e}$ and $\mathbf{M}$. Set the Lagrange multiplier $\mathbf{\Lambda} = \mathbf{\Theta} = \mathbf{\Sigma} = \mathbf{0}$, $\mu_{max} = 10^8$, $\rho = 1.15$ and $\mu = 10^{-4}$.
repeat
  Update $\mathbf{S}$ by solving the Problem (10)
  Update $\mathbf{L}$, $\mathbf{A^h}$ and $\mathbf{A^e}$ by Eq. (11) to Eq. (13)
  Update $\mathbf{F}$ by solving the Problem (14)
  Update $\mathbf{M}$ by Eq. (16)
  Update $\mathbf{\Lambda}$ by $\mathbf{\Lambda} \leftarrow \mathbf{\Lambda} + \mu(\mathbf{L} - \mathbf{I} + \mathbf{S})$
  Update $\mathbf{\Theta}$ by $\mathbf{\Theta} \leftarrow \mathbf{\Theta} + \mu(\mathbf{A} - \mathbf{A^h} - \mathbf{A^e})$
  Update $\mathbf{\Sigma}$ by $\mathbf{\Sigma} \leftarrow \mathbf{\Sigma} + \mu(\mathbf{S} - \mathbf{M})$
  Update μ by $\mu \leftarrow \min(\rho\mu, \mu_{max})$
until *convergence*
return $\mathbf{S}$ *and* $\mathbf{F}$

B. Speedup Strategy

Although the optimization problem in Eq. (6) can be solved by the algorithm proposed in the earlier section, the computational cost is high ($\mathcal{O}(n^3)$) because of the eigen-decomposition for updating $\mathbf{F}$ and the SVD for updating $\mathbf{M}$. It is thus imperative to develop an efficient algorithm that can scale well for large datasets. We further propose pSCAN, a provable SCAN, that is efficient for large datasets. The gain of pSCAN mainly comes from two aspects. First, we propose to use a more sparse affinity matrix for the noisy data, which saves both space and running time for large matrix computation. Second, we adopt some most recent advances [26] in solving eigen-decomposition and SVD [27] to improve the overall efficiency.

1) Random Binning Features: To address the first challenge, we effectively compute the affinity matrix by adopting the *random binning features*, which are successfully scaling up for large-scale kernels [28], [29], [20]. The random binning features consider a feature map:

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \int_{\boldsymbol{\omega}} p(\boldsymbol{\omega}) \boldsymbol{\phi}_{B_{\boldsymbol{\omega}}}(\boldsymbol{x}_i)^T \boldsymbol{\phi}_{B_{\boldsymbol{\omega}}}(\boldsymbol{x}_j) d\boldsymbol{\omega} \tag{17}$$

where a set of bins $B_{\boldsymbol{\omega}}$ is a random grid parameterized by $\boldsymbol{\omega} = (\omega_1, u_1, \ldots, \omega_d, u_d)$ drawn from a distribution $p(\boldsymbol{\omega})$, and a pair (ω_i, u_i) denotes *width* and *bias* in i-th dimension of a grid. For any bin $\boldsymbol{b} \in B_{\boldsymbol{\omega}}$, its feature vector $\boldsymbol{\phi}_{B_{\boldsymbol{\omega}}}(\boldsymbol{x})$ has

$$\boldsymbol{\phi}_{\boldsymbol{b}}(\boldsymbol{x}) = 1, \quad \text{if} \;\; \boldsymbol{b} = (\lfloor \frac{\boldsymbol{x}(1) - u_1}{\omega_1} \rfloor, \ldots, \lfloor \frac{\boldsymbol{x}(d) - u_d}{\omega_d} \rfloor) \tag{18}$$

and $\boldsymbol{\phi}_b(\boldsymbol{x}) = 0$ otherwise.

Following the procedures in [28], [20], [29], given R grids $\{B_{\omega_r}\}_{r=1}^R$ and the data points: $[\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n] = \mathbf{X}^{n \times d}$, we first draw ω_r from $p_r(\omega)$ (e.g., uniform distribution). The random binning feature for point $\boldsymbol{x}_i$: $\boldsymbol{z}_r(\boldsymbol{x}_i)$ is the indicator vector of bin index $(\lfloor \frac{\boldsymbol{x}_i(1) - u_1}{\omega_1} \rfloor, \ldots, \lfloor \frac{\boldsymbol{x}_i(d) - u_d}{\omega_d} \rfloor)$. In order to obtain good features, a simple Monte Carlo method is used to average over R grids of random binning features. The result feature matrix $\mathbf{Z} \in \mathbb{R}^{n \times D}$ is constructed for the original data matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$, where D is determined jointly by the number of grids and kernel width (more details can be found in [29]). Moreover, $\mathbf{Z}$ is a binary matrix and the number of nonzero elements in each row $nnz(\mathbf{Z}(i.:)) = R$, which is thus very sparse. A more sparse affinity matrix $\mathbf{A}$ can be computed by the inner product of the random binning features: $\mathbf{A} = \mathbf{Z}\mathbf{Z}^T$ in linear complexity [28], [20], [29].

Since the input $\mathbf{A}$ is sparse, its high-quality affinity $\mathbf{A^h}$ is naturally sparse with proper initialization. The sparse property also hold for $\mathbf{S}$ and $\mathbf{L}$. We next show that operations on those sparse matrices can improve the time complexity.

2) Fast eigen-decomposition and k-SVD: To deal with the orthogonality constraint in Eq. (14), we use a gradient descent procedure with curvilinear search [26] to update $\mathbf{F}$. In each iteration, given the current feasible point $\mathbf{F}$, and its corresponding gradient $\mathbf{G} = \nabla \operatorname{Tr}(\mathbf{F}^T\mathbf{L}\mathbf{F}) = 2\mathbf{L}\mathbf{F}$, we define the skew-symmetric matrices: $\mathbf{P} = \mathbf{G}\mathbf{F}^T - \mathbf{F}\mathbf{G}^T$. The next feasible point $\mathbf{F}$ can be searched along the smooth curve:

$$\mathbf{Q}(\tau) = (\mathbf{I} + \frac{\tau}{2}\mathbf{P})^{-1}(\mathbf{I} - \frac{\tau}{2}\mathbf{P})\mathbf{F} \tag{19}$$

It is easy to verify that $\mathbf{Q}(\tau)$ is staying on the Stiefel manifold: $\mathcal{M} = \{\mathbf{X} : \mathbf{X}^T\mathbf{X} = \mathbf{I}\}$. The derivative with respect to τ is given $\frac{d\mathbf{Q}(\tau)}{d\tau} = -(\mathbf{I} + \frac{\tau}{2}\mathbf{P})^{-1}\mathbf{P}(\mathbf{F} + \mathbf{Q}(\tau))/2$. Because $\frac{d\mathbf{Q}(\tau)}{d\tau}|_{\tau=0} = -\mathbf{P}\mathbf{F}$ is the same as the projection of $-\mathbf{P}$ onto the tangent space of $\mathcal{M}$ at current point $\mathbf{F} = \mathbf{Q}(0)$, the curve $\mathbf{Q}(\tau)$ is a descent curve along $\tau > 0$. We use the classical nonmonotone line search with the Barzilai-Borwein algorithm to accelerate the gradient at each iteration:

$$\tau^{(t)} = \frac{\operatorname{Tr}[(\mathbf{F}^{(t)} - \mathbf{F}^{(t-1)})^T(\mathbf{F}^{(t)} - \mathbf{F}^{(t-1)})]}{|\operatorname{Tr}[2(\mathbf{F}^{(t)} - \mathbf{F}^{(t-1)})^T(\mathbf{L}\mathbf{F}^{(t)} - \mathbf{L}\mathbf{F}^{(t-1)})]|}\eta^h$$

here h is the smallest integer that satisfies:

$$\mathcal{F}[\mathbf{Q}^{(t)}(\tau^{(t)})] \leq r^{(t)} + \nu\tau^{(t)} \cdot \frac{d}{d\tau}\mathcal{F}[\mathbf{Q}^{(t)}(\tau^{(t)})]|_{\tau=0}$$

where $\mathcal{F}[\mathbf{Q}^{(t)}(\tau^{(t)})] = \operatorname{Tr}[\mathbf{Q}^{(t)}(\tau^{(t)})^T\mathbf{L}\mathbf{Q}^{(t)}(\tau^{(t)})]$; η, ν, ζ are all pre-define positive constants. $r^{(0)} = \operatorname{Tr}(\mathbf{F}^{(0)^T}\mathbf{L}\mathbf{F}^{(0)})$, $r^{(t)} = [\zeta s^{(t-1)} r^{(t-1)} + \operatorname{Tr}(\mathbf{F}^{(t)^T}\mathbf{L}\mathbf{F}^{(t)})]/s^{(t)}$, in which $s^{(0)} = 1$, $s^{(t)} = \zeta s^{(t-1)} + 1$. The theoretical convergence of curvilinear search algorithm can be found in the study [26].

In addtion, computing $\mathbf{M} = \mathbf{U}\operatorname{Diag}(\sigma(\mathbf{M}))\mathbf{V}^T$ in the original algorithm (Eq. (16)) requires SVD decomposition of $\mathbf{S}$, which is also not scalable to large data. Here, we will use a low-rank approximation of $\mathbf{M}$. From previous analysis, the top-$(k+1)$ singular values of $\mathbf{M}$ are more informative, we use rank-$(k+1)$ approximation of $\mathbf{M}$ as :

$$\mathbf{M}_{k+1} = \mathbf{U}_{k+1}\Delta_{k+1}\mathbf{V}_{k+1}$$

where $\mathbf{U}_{k+1} = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_{k+1}]$, $\mathbf{V}_{k+1} = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_{k+1}]$, and $\Delta_{k+1} = [\mathbb{1}_k^T, 1-\delta]$. With the low-rank approximation, the error between $\mathbf{M}$ and $\mathbf{M}_{k+1}$ is bounded as:

$$\|\mathbf{M} - \mathbf{M}_{k+1}\|_F^2 = \| \sum_{i=k+2}^{n} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^T \|^2 = \sum_{i=k+2}^{n} {\sigma_i}^2 < (n-k-1)$$

To this end, we only need the first $k+1$ singular vectors of the matrix $\widetilde{\mathbf{S}}$, which leads to a k-SVD problem. Recently, a few breakthroughs have been discovered for k-SVD, such as block Krylov method [27], variance-reduction stochastic method [30], and LazySVD [31]. In this work, we use the block Krylov method due to its gap-free convergence results.

C. Convergence and Time Complexity

The convergence of ALM has been proved and discussed in previous studies [25]. The complexity of pSCAN can be derived as follows. The computational complexity for computing random binning feature matrix $\mathbf{Z}$ takes $\mathcal{O}(ndR)$. Computing $\mathbf{Q}(\tau)$ requires $\mathcal{O}(nk^2 + k^3)$ by using Sherman-Morrison formula [26]. Evaluating objective function of Eq. (14) and its derivatives takes $\mathcal{O}(nnz(L)k^2)$. In addition, the block Krylow method for k-SVD takes $\mathcal{O}(nnz(S)k + nk^2 + k^3)$. Therefore, the total time complexity of pSCAN $\mathcal{O}(nnz(L)k^2 + nnz(S)k + nk^2 + k^3 + ndR)$. In practice, matrices L and S are sparse due to the random binning features. Furthermore, the number of clusters k, the number of grids R, and the number of features d are all much smaller than n. pSCAN is thus more effective than most of spectral clustering algorithms with time complexity of $\mathcal{O}(n^3)$.

VI. EXPERIMENTS

In this section, we assess the performance of the proposed methods on both synthetic and real benchmark datasets and compare it with several popular spectral clustering methods.

A. Experimental Settings

We compare the proposed SCAN and pSCAN with the following baseline methods:

- NCut [1]: the traditional normalized cut algorithm.
- SNMF [32]: a symmetric nonnegative matrix factorization method to decompose affinity matrix.
- DSN [7], a doubly stochastic normalization for affinity matrix, and using Ncut to obtain the clusters.
- SDS [9], a structured doubly stochastic approximation model for graph-based clustering.
- CSC [33], a scalable constrained spectral clustering method by using sparse coding.
- CSS [34], a spectral clustering algorithm with additional sparsity constraint.
- SCN [35], a fast spectral clustering method based on Nyström approximation method.

We use two commonly used clustering metrics for cluster quality evaluation: Accuracy (ACC) and Normalized Mutual Information (NMI) [36]. Accuracy denotes the percentage of the correctly assigned labels and is defined as:

$$ACC = \frac{\sum_{i=1}^{n} \delta(y_i, map(c_i))}{n}$$

where n is the total number of instances; y_i is the ground truth class label; c_i is the label assigned by the algorithm, which is mapped to a true class label through the mapping function $map(c_i)$ [36]; $\delta(x, y)$ is the delta function that equals one if $x = y$ and equals zero otherwise.

NMI is defined as:

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

where C and C' denote the real and the predicted label vectors, $MI(C, C')$ is the mutual information based on C and C'; $H(\cdot)$ is the entropy of clusters. The detailed definition of $MI(C, C')$ and $H(C)$ can be found in [36].

In the experiments, we set the number of clusters to be the same as ground truth in each dataset [7]. For all compared methods, their parameters are tuned for optimal performance in the experiments. For our model, the eigengap δ is tuned within $(0, 1)$. The regularized parameters α, β and γ vary in the range of $\{10^{-2}, 10^{-1}, 10^0, 10^1, 10^2\}$, and the number of grids for random binning features is suggested as $R \leq D \leq nR$ [29]. All parameters can be tuned for each dataset and the optimal results are reported. The impact of parameters will be studied later. For each method, the experiments are running repeat 20 times independently and the average results are reported.

B. Synthetic Dataset

We first conduct experiments on a synthetic dataset to test robustness of SCAN and pSCAN with noisy data. The synthetic dataset is a 100×100 matrix with four 25×25 block sub-matrices along the diagonal [9]. The data within same block indicates that they are in the same cluster and should be connected, while the data outside all blocks denotes noise (which should be zeros in the ideal clustering structure). The affinity values within each block are randomly generated within $(0, 1)$; while the off-diagonal noise data is randomly generated in the range of $(0, c)$, where c is set to be 0.55, 0.65 and 0.75, respectively, to reflect different levels of noise. Note that the random binning feature technique is not used for pSCAN in the synthetic dataset, since we directly generate the noise affinity matrix.

Among the comparison methods, DSN [7] and SDS [9] are the most closely related approaches to our proposed models since all of them aim to learn a doubly stochastic affinity matrix. In the synthetic dataset experiments, we mainly compare SCAN and pSCAN against these two methods. Figure 1 shows the original affinity input matrix and corresponding approximation results from DSN, SDS, SCAN and pSCAN. We can observe that SCAN and pSCAN constantly perform better than DSN and SDS at all levels of noise. In particular, SDS seems to outperform DSN at low levels of noise (e.g.,

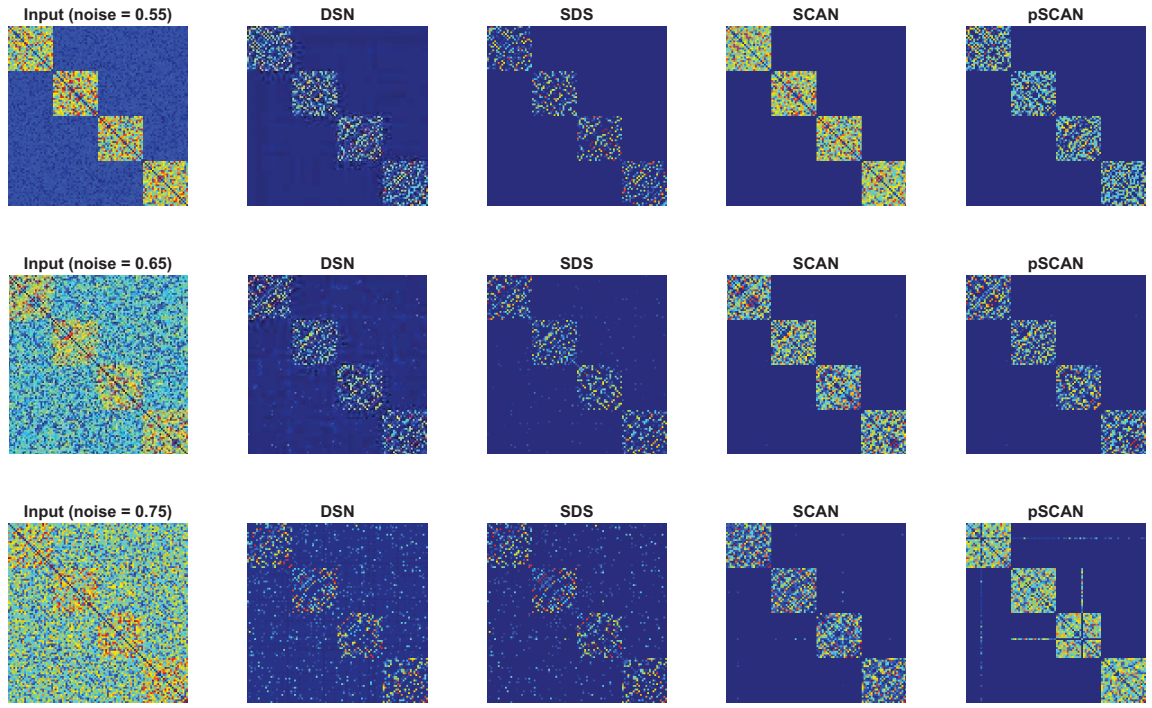

Fig. 1: Illustrations of clustering results on the block diagonal synthetic data with different levels of noise. The left column is the original graph affinity matrix in the experiment. The rest are the doubly stochastic affinity matrices obtained by DSN, SDS, SCAN and pSCAN respectively.

TABLE I: Statistics of the benchmark datasets.

Dataset	# instances	Dimensions	# Classes
COIL	1500	241	6
Yeast	1484	8	10
Statlog	946	18	4
ORL	400	1024	40
USPS	9,298	256	10
Pendigits	10,992	16	10
Letter	20,000	16	26
Mnist	70,000	780	10

$c = 0.55, 0.65$). As the noise increases (e.g., $c = 0.75$), SDS fails to detect the intrinsic cluster structure from the data. The proposed SCAN and pSCAN successfully learn a stable doubly stochastic matrix with explicit block structure even with high level of noise, which illustrate their robustness.

C. Benchmark Datasets

We further compare the performance of SCAN with other approaches on four small benchmark datasets: COIL[2], Yeast[3], Statlog[4] and ORL[5]. COIL is subset of the Columbia object image library (COIL-100), which contains a set of color images of 100 different objects taken from different angles. The subset contains 1500 images from 6 objects. Yeast contains 1484 sequences of proteins and each protein belongs to one of nine different cellular components. The original study is to utilize various descriptors to predict protein localizations in a cell. Statlog collects 946 vehicles, the features of which can be extracted from a 2D silhouette image. ORL contains a set of face images from AT&T lab. There are ten different images for each of 40 distinct subjects. We also choose another four large dataset from the Lib-SVM project[6], which are USPS, Pendigits, Letter and Mnist. USPS is an image database for handwritten text recognition research. Pendigits is also handwritten digit data set consisting of 250 samples from 44 users. Letter is a collection of images for 26 capital letters in the English alphabet. Minst is another popular collection of handwritten digit data set, in where each image is represented by a 780 dimensional vector. A summery of these datasets is given in Table I.

[2]http://olivier.chapelle.cc/ssl-book/benchmarks.html
[3]https://archive.ics.uci.edu/ml/datasets/Yeast
[4]https://archive.ics.uci.edu/ml/datasets/StatlogVehicle+(Vehicle+Silhouettes)
[5]https://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html
[6]https://www.csie.ntu.edu.tw/ cjlin/libsvmtools/datasets/

TABLE II: Experimental results on real benchmark datasets.

	Dataset	Ncut	SNMF	DSN	SDS	CSC	CSS	SCN	SCAN	pSCAN
ACC	COIL	0.219	0.206	0.286	0.302	0.276	0.298	0.293	**0.314**	0.312
	Yeast	0.401	0.411	0.454	0.483	0.463	0.479	0.449	**0.502**	0.498
	Statlog	0.462	0.458	0.477	0.479	0.453	0.473	-	**0.486**	0.481
	ORL	0.618	0.621	0.638	0.659	0.614	0.657	-	**0.667**	0.654
	USPS	0.663	0.654	0.668	0.679	0.621	**0.685**	0.601	0.683	0.673
	Pendigits	0.761	0.771	0.769	0.776	0.763	0.778	0.757	**0.821**	0.813
	Letter	0.306	0.304	0.311	0.317	0.309	0.312	0.301	**0.323**	0.311
	Mnist	0.659	0.712	0.671	-	0.698	-	0.553	-	**0.734**

	Dataset	Ncut	SNMF	DSN	SDS	CSC	CSS	SCN	SCAN	pSCAN
NMI	COIL	0.203	0.201	0.207	0.206	0.207	0.212	0.201	**0.218**	0.213
	Yeast	0.254	0.257	0.259	0.263	0.255	0.261	0.254	**0.276**	0.273
	Statlog	0.187	0.188	0.192	0.194	0.186	0.191	-	**0.212**	0.196
	ORL	0.782	0.796	0.806	0.816	0.791	0.818	-	**0.821**	0.813
	USPS	0.632	0.620	0.649	0.668	0.590	**0.674**	0.583	0.671	0.664
	Pendigits	0.713	0.738	0.735	0.739	0.724	0.733	0.702	**0.791**	0.786
	Letter	0.386	0.382	0.397	0.410	0.396	0.408	0.392	**0.412**	0.401
	Mnist	0.592	0.631	0.586	-	0.586	-	0.502	-	**0.694**

All methods require an affinity matrix as the input. For all methods except pSCAN, We adopt a widely used self-tune Gaussian kernel method to construct the input affinity matrix [13], in which the number of neighbors is set to be five and the value of σ is self-tuned. For pSCAN, we use the random binning features kernel to generate the affinity matrix that is more sparse than the matrix from Gaussian kernel. Among those methods, NCut, DSN, SDS, CSS and SCAN require full computation of SVD or eigen-decomposition and are thus slow to handle large datasets.

1) Experimental Results: For Mnist dataset, we skip the experiments for SDS, CSS, and SCAN due to their requiring full SVD and eigen-decomposition at each iteration. For Statlog and ORL datasets, we omit SCN because the sampling steps make little sense for small datasets. The results for other datasets are listed in Table II. Both SCAN and pSCAN perform well in almost all experiments. There are several interesting observations. First, for a majority of the benchmark datasets, the performance of DSN, SDS, pSCAN and SCAN are better than that of the Normalized cut, which shows the importance of doubly stochastic normalization of the original affinity matrix. Second, pSCAN, SCAN and DSN consistently perform better than SNMF. The primary reason is that SNMF assumes the affinity matrix to be low-rank. The block-diagonal affinity matrices of pSCAN, SCAN, and DSN characterize the data clusters more accurately, which are not only low-rank but doubly stochastic. Third, SCAN and pSCAN have better results than DSN with an average improvement of 5.6% in terms of ACC and 5.3% in terms of NMI, respectively. Forth, among the seven benchmark datasets, SCAN outperforms all the methods on all datasets with the only exception of the USPS dataset, on which CSS has the best performance, mainly due to its sparsity constraint. In comparison with SDS, SDS learns a structured doubly stochastic matrix with the rank constraint on its Laplacian matrix to maintain an explicit block structure. However, the doubly stochastic matrix of SDS may be biased since it is expected to be close to the original affinity matrix, which may be corrupted by noise. On the contrary, SCAN and pSCAN learn the doubly stochastic matrix from a more clean affinity matrix.

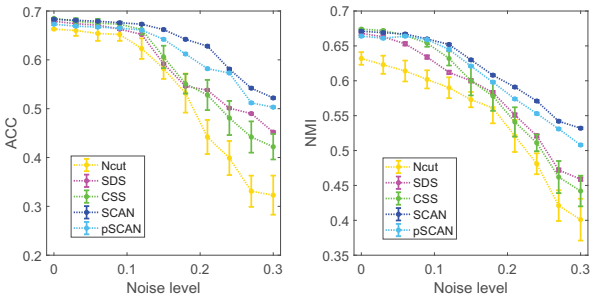

Fig. 2: Robustness to noise. Our SCAN and pSCAN clearly outperform other spectral clustering methods.

2) Robustness to Noise: We analyze the robustness of our proposed methods. To be specific, we study how an increasing degree of noisy data will affect the clustering performance. We use perturbed USPS data by adding Gaussian noise to its affinity matrix with variance from 0 to 0.3. For this analysis, we mainly compare SCAN and pSCAN with Ncut, SDS and CSS, based on their promising results in previous section.

Figure 2 shows the results of our methods and other spectral clustering algorithms. The lines represent the mean ACC and NMI values, while the error bars represent the variance for different runs. Clearly, SCAN and pSCAN are more robust to noise and outperform other approaches. The clustering quality of other spectral clustering (i.e., Ncut, SDS and CSS) decreases rapidly as the noise increases. The main reason is that SCAN and pSCAN learn an explicit error matrix to split the noise from the input data in Eq. (3). Furthermore, SCAN and pSCAN consider the Laplacian eigengap to refine the eigenspace of the top k eigenvectors, which make the clusters of SCAN and pSCAN more stable than others.

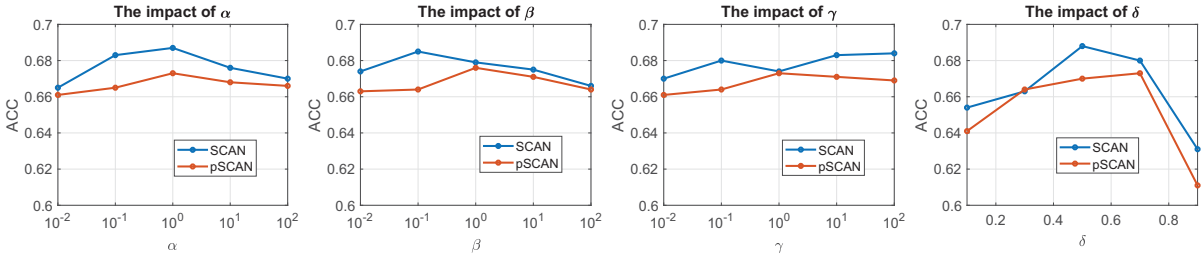

Fig. 3: Parameter studies of α, β, γ, and δ on the USPS dataset.

3) Parameter Studies: In our model (Eq. (6)), there are four major parameters α, β, γ, and δ, where α is used to control the sparsity of the error matrix in the graph; β is used to maintain the block diagonal structure of the learning matrix; γ forces the elements in each block of the matrix to be the same; and δ is a threshold for eigengap. One can adopt a grid-based search algorithm to find the set of best parameters, which requires significant computation time. Here, we perform a partial search by fixing the value of three parameters and study the impact of the last one on the inference results. For example, while studying the impact of parameter α, we fix eigengap $\delta = 0.6$ and the regularized parameters $\beta = \gamma = 0.1$. We then vary α in the range of $\{10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}\}$ in the experiments. Figure 3 show the clustering accuracy on the USPS dataset. Generally speaking, the regularized parameters α, β, and γ are relatively stable between $(0.1, 1)$. For eigengap, a relatively high accuracy can be achieved when δ is between 0.5 to 0.7. The experiments can be performed on other benchmark datasets, we omit the results due to space limitation.

4) Complexity and Convergence: In terms of efficiency, we mainly compare SCAN and pSCAN with CSS and SDS since they all involve full computation of SVD or eigen-decomposition. We evaluate the efficiency of different algorithm by using the Letter dataset. We randomly generate a subset of the balanced classes with different number of instances within $\{2,000, 4,000, \ldots, 20,000\}$. The experiments are conducted on a 2.40GHz machine with 48GB memory. Figure 4(a) shows the running time when varying the number of instances. Generally, the proposed pSCAN are much more efficient than the others, which demonstrate the effectiveness of our speedup strategy.

Figure 4(b) also shows the value of the objective function of Eq. (6) w.r.t. the number of iterations on Letter datasets with $10,000$ instances. Usually, less than 80 iterations are needed for convergence. Although in general SCAN requires less number of iterations for convergence comparing to pSCAN, pSCAN still runs faster because each of its iteration actually costs much less time comparing to SCAN.

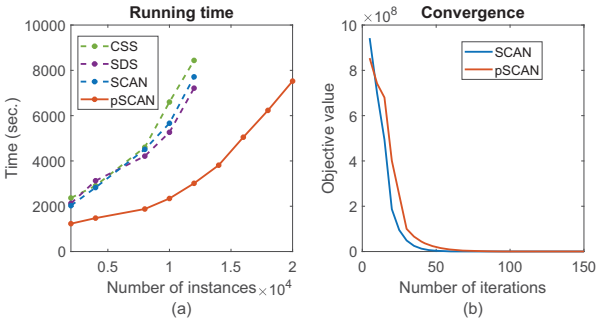

Fig. 4: (a) Running time of different methods. (b) Convergence of SCAN and pSCAN.

VII. CONCLUSION AND FUTURE WORK

In this paper, we propose a structure-aware clustering algorithm for noisy data. Our proposed models are able to learn a better doubly stochastic affinity matrix, from which we can immediately partition the data into k connected components. To achieve this goal, we learn a better doubly stochastic matrix from the high-quality affinity matrix, which explicitly reflects the true clusters. We also apply the idea of clustering stability based on Laplacian eigengap. The intuition is that a clustering is stable if small distortions on the data does not affect the discoverability of the data clustering structure. Mathematically, we augment the eigengap of our target affinity matrix. We model the clustering problem as a constrained optimization problem and derive an efficient algorithm to solve the problem. Experimental results on both synthetic and real-world datasets demonstrate the effectiveness and robustness of the proposed models for noisy data.

The stable doubly stochastic affinity matrix from our proposed algorithm can be directly applied to any machine learning tasks that require an affinity matrix as an input, such as semi-supervised learning. For example, many semi-supervised learning methods generally include two steps [37]: graph affinity matrix construction and label propagation on the graph. SCAN can learn a better graph affinity matrix in the first step. As future work, we are interested in extending our approach to semi-supervised learning.

Acknowledgments: This work has been supported in part by NSF CCF1815139 and by an allocation of computing time from the Ohio Supercomputer Center.

REFERENCES

[1] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and computing*, vol. 17, no. 4, pp. 395–416, 2007.

[2] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *NeurIPS*, 2002.

[3] H. Chen, S. K. Iyengar, and J. Li, "Large-scale analysis of drug combinations by integrating multiple heterogeneous information networks," in *BCB*, 2019.
[4] R. Xia, Y. Pan, L. Du, and J. Yin, "Robust multi-view spectral clustering via low-rank and sparse decomposition." in *AAAI*, 2014.
[5] A. Bojchevski, Y. Matkovic, and S. Günnemann, "Robust spectral clustering for noisy data: Modeling sparse corruptions improves latent embeddings," in *KDD*, 2017.
[6] X. Zhu, C. Change Loy, and S. Gong, "Constructing robust affinity graphs for spectral clustering," in *CVPR*, 2014.
[7] R. Zass and A. Shashua, "Doubly stochastic normalization for spectral clustering," in *NeurIPS*, 2007.
[8] J. Park and T. Kim, "Learning doubly stochastic affinity matrix via davis-kahan theorem," in *ICDM*, 2017.
[9] X. Wang, F. Nie, and H. Huang, "Structured doubly stochastic matrix for graph based clustering: Structured doubly stochastic matrix," in *KDD*, 2016.
[10] F. Nie, X. Wang, M. I. Jordan, and H. Huang, "The constrained laplacian rank algorithm for graph-based clustering." in *AAAI*, 2016.
[11] F. Nie, X. Wang, and H. Huang, "Clustering and projected clustering with adaptive neighbors," in *KDD*, 2014.
[12] M. Meila, S. M. Shortreed, and L. Xu, "Regularized spectral learning." in *AISTATS*, 2005.
[13] L. Zelnik-Manor and P. Perona, "Self-tuning spectral clustering," in *NeurIPS*, 2005.
[14] V. Premachandran and R. Kakarala, "Consensus of k-nns for robust neighborhood selection on graph-based manifolds," in *CVPR*, 2013.
[15] C. D. Correa and P. Lindstrom, "Locally-scaled spectral clustering using empty region graphs," in *KDD*, 2012.
[16] Z. Yang and E. Oja, "Clustering by low-rank doubly stochastic matrix decomposition," in *ICML*, 2012.
[17] A. Douik and B. Hassibi, "Low-rank riemannian optimization on positive semidefinite stochastic matrices with applications to graph clustering," in *ICML*, 2018.
[18] J. Hu, Q. Qian, J. Pei, R. Jin, and S. Zhu, "Finding multiple stable clusterings," *Knowledge and Information Systems*, vol. 51, no. 3, pp. 991–1021, 2017.
[19] H. Huang, S. Yoo, H. Qin, and D. Yu, "A robust clustering algorithm based on aggregated heat kernel mapping," in *ICDM*, 2011.
[20] L. Wu, P.-Y. Chen, I. E.-H. Yen, F. Xu, Y. Xia, and C. Aggarwal, "Scalable spectral clustering using random binning features," in *KDD*, 2018.
[21] H. Chen and J. Li, "Exploiting structural and temporal evolution in dynamic link prediction," in *CIKM*, 2018.
[22] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis," *Journal of the ACM (JACM)*, vol. 58, no. 3, p. 11, 2011.
[23] F. R. Chung and F. C. Graham, *Spectral graph theory*. American Mathematical Soc., 1997, no. 92.
[24] K. Fan, "On a theorem of weyl concerning eigenvalues of linear transformations i," *Proceedings of the National Academy of Sciences*, vol. 35, no. 11, pp. 652–655, 1949.
[25] Z. Lin, M. Chen, and Y. Ma, "The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices," *arXiv preprint arXiv:1009.5055*, 2010.
[26] Z. Wen and W. Yin, "A feasible method for optimization with orthogonality constraints," *Mathematical Programming*, vol. 142, no. 1-2, pp. 397–434, 2013.
[27] C. Musco and C. Musco, "Randomized block krylov methods for stronger and faster approximate singular value decomposition," in *NeurIPS*, 2015.
[28] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," in *NeurIPS*, 2008.
[29] L. Wu, I. E. Yen, J. Chen, and R. Yan, "Revisiting random binning features: Fast convergence and strong parallelizability," in *KDD*, 2016.
[30] O. Shamir, "Fast stochastic algorithms for svd and pca: Convergence properties and convexity," in *ICML*, 2016.
[31] Z. Allen-Zhu and Y. Li, "Lazysvd: Even faster svd decomposition yet without agonizing pain," in *NeurIPS*, 2016.
[32] C. Ding, X. He, and H. D. Simon, "On the equivalence of nonnegative matrix factorization and spectral clustering," in *SDM*, 2005.
[33] J. Li, Y. Xia, Z. Shan, and Y. Liu, "Scalable constrained spectral clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 2, pp. 589–593, 2015.
[34] C. Lu, S. Yan, and Z. Lin, "Convex sparse spectral clustering: Single-view to multi-view," *IEEE Transactions on Image Processing*, vol. 25, no. 6, pp. 2833–2843, 2016.
[35] C. Fowlkes, S. Belongie, F. Chung, and J. Malik, "Spectral grouping using the nystrom method," *IEEE transactions on pattern analysis and machine intelligence*, vol. 26, no. 2, pp. 214–225, 2004.
[36] W. Xu, X. Liu, and Y. Gong, "Document clustering based on non-negative matrix factorization," in *SIGIR*, 2003.
[37] K. Kamnitsas, D. Castro, L. Le Folgoc, I. Walker, R. Tanno, D. Rueckert, B. Glocker, A. Criminisi, and A. Nori, "Semi-supervised learning via compact latent space clustering," in *ICML*, 2018.
[38] L. Mirsky, "A trace inequality of john von neumann," *Monatshefte für mathematik*, vol. 79, no. 4, pp. 303–306, 1975.

APPENDIX

We first introduce the von Neumann's trace inequality [38] as:

Lemma 3. *(von Neumann's trace inequality) For any two matrices* $\mathbf{X}$ *and* $\mathbf{Y} \in \mathbb{R}^{m\times n}(m \le n)$. *Let* $\sigma_1(\mathbf{X}) \ge \sigma_2(\mathbf{X}) \ge \cdots \ge 0$ *and* $\sigma_1(\mathbf{Y}) \ge \sigma_2(\mathbf{Y}) \ge \cdots \ge 0$ *are the singular values of* $\mathbf{X}$ *and* $\mathbf{Y}$*, respectively. We have*

$$\mathrm{Tr}(\mathbf{X}^T\mathbf{Y}) \le \sum_{i=1}^{m} \sigma_i(\mathbf{X})\sigma_i(\mathbf{Y})$$

The equality holds if and only if there exist matrices $\mathbf{U}$ *and* $\mathbf{V}$ *such that* $\mathbf{X} = \mathbf{U}\,\mathrm{Diag}(\sigma(\mathbf{X}))\mathbf{V}^T$ *and* $\mathbf{Y} = \mathbf{U}\,\mathrm{Diag}(\sigma(\mathbf{Y}))\mathbf{V}^T$ *are the Singular Value Decomposition (SVD) of* $\mathbf{X}$ *and* $\mathbf{Y}$*, simultaneously.*

Theorem 1. *The following gives the global optimal solution to Eq. (15):*

$$\mathbf{M} = \mathbf{U}\,\mathrm{Diag}(\sigma(\mathbf{M}))\mathbf{V}^T$$

where

$$\sigma_i(\mathbf{M}) = \begin{cases} 1 & \text{if } 1 \le i \le k \\ \min(1-\delta, \sigma_i(\widetilde{\mathbf{S}})) & \text{if } k+1 \le i \le n \end{cases}$$

where $\mathbf{U}$ *and* $\mathbf{V}$ *are the left and right orthonormal matrices in the SVD of* $\widetilde{\mathbf{S}}$.

Proof. By Lemma 1, we have

$$\begin{aligned} \|\mathbf{M}-\widetilde{\mathbf{S}}\|_F^2 &= \mathrm{Tr}(\mathbf{M}^T\mathbf{M}) + \mathrm{Tr}(\widetilde{\mathbf{S}}^T\widetilde{\mathbf{S}}) - 2\,\mathrm{Tr}(\mathbf{M}^T\widetilde{\mathbf{S}}) \\ &= \sum_{i=1}^{n}\sigma_i^2(\mathbf{M}) + \sum_{i=1}^{n}\sigma_i^2(\widetilde{\mathbf{S}}) - 2\,\mathrm{Tr}(\mathbf{M}^T\widetilde{\mathbf{S}}) \\ &\ge \sum_{i=1}^{n}\sigma_i^2(\mathbf{M}) + \sum_{i=1}^{n}\sigma_i^2(\widetilde{\mathbf{S}}) - 2\sum_{i=1}^{n}\sigma_i(\mathbf{M})\sigma_i(\widetilde{\mathbf{S}}) \\ &= \sum_{i=1}^{n}(\sigma_i(\mathbf{M}) - \sigma_i(\widetilde{\mathbf{S}}))^2 \end{aligned} \tag{20}$$

From Eq. (20) we obtain the lower bound for Eq. (15). Note that the above equality holds when $\mathbf{M} = \mathbf{U}\,\mathrm{Diag}(\sigma(\mathbf{M}))\mathbf{V}^T$, where $\mathbf{U}$ and $\mathbf{V}$ are the left and right orthonormal matrix in the SVD of $\widetilde{\mathbf{S}}$. And Eq. (15) is simplified as:

$$\min_{\mathbf{M}} \quad \sum_{i=1}^{n}(\sigma_i(\mathbf{M}) - \sigma_i(\widetilde{\mathbf{S}}))^2 \quad \text{s.t.} \quad \sigma_{k+1}(\mathbf{M}) \le 1-\delta \tag{21}$$

Since Eq. (21) consists of individual quadratic function for every $\sigma_i(\mathbf{M})$ independently, it's easy to obtain every $\sigma_i(\mathbf{M})$ under the inequality constraint and the first k singular values are all one. The optimal solution is

$$\sigma_i(\mathbf{M}) = \begin{cases} 1 & \text{if } 1 \le i \le k \\ \min(1-\delta, \sigma_i(\widetilde{\mathbf{S}})) & \text{if } k+1 \le i \le n \end{cases}$$

□