Received 19 January 2024; revised 2 May 2024; accepted 2 May 2024. Date of publication 13 May 2024;
date of current version 16 July 2024. The review of this article was arranged by Associate Editor Shunqiao Sun.

Digital Object Identifier 10.1109/OJSP.2024.3400683

Robust Regularized Locality Preserving Indexing for Fiedler Vector Estimation

AYLIN TAŞTAN [1], MICHAEL MUMA [2] (Senior Member, IEEE), AND ABDELHAK M. ZOUBIR [3] (Fellow, IEEE)

[1] Pattern Recognition Group, University of Bern, 3012 Bern, Switzerland
[2] Robust Data Science Group, Technische Universität Darmstadt, 64283 Darmstadt, Germany
[3] Signal Processing Group, Technische Universität Darmstadt, 64283 Darmstadt, Germany

CORRESPONDING AUTHOR: AYLIN TAŞTAN (email: aylin.tastan@unibe.ch).

The work of Aylin Taştan was supported by the Republic of Turkey Ministry of National Education. The work of Michael Muma was supported by the LOEWE initiative (Hesse, Germany) within the emergenCITY centre.
This article has supplementary downloadable material available at https://doi.org/10.1109/OJSP.2024.3400683, provided by the authors.

ABSTRACT The Fiedler vector is the eigenvector associated with the algebraic connectivity of the graph Laplacian. It is central to graph analysis as it provides substantial information to learn the latent structure of a graph. In real-world applications, however, the data may be subject to heavy-tailed noise and outliers which deteriorate the structure of the Fiedler vector estimate and lead to a breakdown of popular methods. Thus, we propose a Robust Regularized Locality Preserving Indexing (RRLPI) Fiedler vector estimation method that approximates the nonlinear manifold structure of the Laplace Beltrami operator while minimizing the impact of outliers. To achieve this aim, an analysis of the effects of two fundamental outlier types on the eigen-decomposition of block affinity matrices is conducted. Then, an error model is formulated based on which the RRLPI method is developed. It includes an unsupervised regularization parameter selection algorithm that leverages the geometric structure of the projection space. The performance is benchmarked against existing methods in terms of detection probability, partitioning quality, image segmentation capability, robustness and computation time using a large variety of synthetic and real data experiments.

INDEX TERMS Dimension reduction, eigen-decomposition, eigenvectors, Fiedler vector, locality preserving indexing.

I. INTRODUCTION

The Fiedler vector of a connected graph is the eigenvector associated with the second smallest eigenvalue, the so called Fiedler value, of the graph Laplacian matrix. The Fiedler vector and the Fiedler value provide important information for estimating [1], [2], [3] and controlling [4], [5], [6] the algebraic connectivity of a graph, finding densely connected groups of vertices that are hidden in the graph structure [7], [8], [9], [10], and representing the implicit relationships between variables in a low-dimensional space [11], [12]. Due to its central role in graph analysis, the computation of the Fiedler vector has been a fundamental research area for decades [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30], [31], [32], [33], [34], [35], [36], [37], [38], [39], [40].

A popular method to embed data-points into a low-dimensional space is Laplacian eigenmaps (LE) [39] method for which the embedding is performed based on the eigen-decomposition of Laplacian matrix. It can therefore also be used for Fiedler vector computation. The LE performs a nonlinear dimensionality reduction while preserving the local neighborhood information in a certain sense, and it explicitly reveals the manifold structure [40]. An alternative approach is that of locality preserving indexing (LPI) [15], which transforms the nonlinear dimensionality reduction in the Laplace Beltrami operator into a linear system of equations. LPI requires a complete singular value decomposition (SVD), resulting in a considerable computational complexity, which is why computationally more attractive alternative approaches have been proposed in [17], [30]. However, the performance of [17] strongly depends on the penalty parameter selection.

In addition to challenges arising from computational complexity and the setting of the penalty parameter, in real-world scenarios, outliers and heavy-tailed noise may obscure the

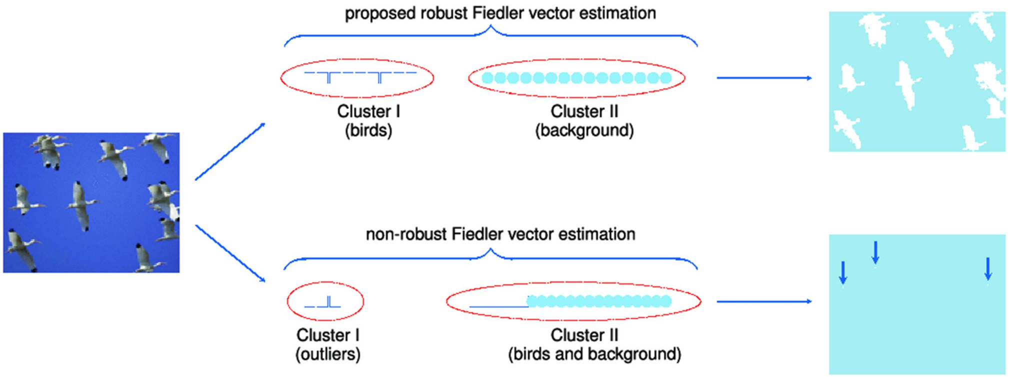


FIGURE 1. **Examplary image segmentation result comparing the popular Laplacian eigenmaps (LE) [39] and the proposed Robust Regularized Locality Preserving Indexing (RRLPI) methods for Fiedler vector estimation.**

graph structure that represents the clean data. Consequently, the computed Fiedler vectors are corrupted, and embeddings based on these vectors no longer provide useful information about the majority of the data set, as they are dominated by outliers. Therefore, robust Fiedler vector estimation methods are needed. One popular strategy is to mitigate the effects of outliers in the *representation space* by restructuring the affinity matrix based on prior information, such as, the number of clusters [41], [42] and the level of sparsity [21], [41], [42], [43], [44] that plays a crucial role in the structure of the eigenvectors. However, it becomes very challenging to determine such prior information, especially in the presence of outliers and heavy-tailed noise that may completely obscure the underlying structure. Alternatively, outliers can be suppressed in the *projection space*, such as in [19], [20], [21], [32], [45]. However, most of these approaches, again, require prior knowledge, e.g., the label information of a data set [19], [20] or data dependent parameter tuning to determine the descriptive features [21]. Moreover, the robust projection operation in [45], uses the ℓ_1 norm that creates a different eigenbasis and requires prior information about the data, i.e. representative samples. The robust locality preserving feature mapping (RLPFM) approach in [32] preserves the ℓ_2 norm and builds upon M-estimation to suppress the outliers. However, it performs M-estimation of the eigenvectors by iteratively reweighting the residuals of Laplacian eigenmaps-based prediction which results in a large computation cost.

Contributions: To address these issues of the above discussed methods, we propose a new Robust Regularized Locality Preserving Indexing (RRLPI) method for Fiedler vector estimation. The key idea is to provide robustness in the *embedding space* by transforming the Fiedler vector estimation problem into a linear system of equations that reveals the hidden group structure in a given graph without assuming any prior knowledge or available training data. Motivated by the importance of the sparsity level in the Fiedler vector structure, we begin by distinguishing fundamental outlier types and investigate how their occurrence depends on the determined sparsity level. This analysis of the effects of outliers on the eigen-decomposition enables us to understand how to best integrate robustness into the Fiedler vector computation. Based on our analysis, we show that the weighted node degree of a vertex is a valuable information to identify an outlier. Therefore, an error model is formulated based on the typical weighted node degree of a graph. Unlike other embedding approaches whose performance strongly depends on correctly setting parameters with the help of prior knowledge, e.g., [17], [19], [20], [21], our penalty parameter determination is formulated as part of the optimization (similar to [32]) based on Δ-separated sets [46] which are defined based on geometric analysis of *well-separated* ℓ_2^2 *representations*. However, in contrast to RLPFM [32], RRLPI robustly estimates the Fiedler vector based on the typical weighted node degree of the graph, which incorporates the weighting operation into a single step and makes the proposed method computationally efficient in comparison to [32].

Illustrating example: An image segmentation application illustrating the need for robust Fiedler vector estimation is provided in Fig. 1. Starting from an original image including birds and background (sky), the aim clearly is to assign birds and the background into different segments. To this end, the image is represented as graph and the Fiedler vectors are computed using RRLPI (top) and LE (bottom). The resulting Fiedler vectors are then clustered into two groups. As can be seen, the LE based Fiedler vector computation results in assigning outlying entries of the Fiedler vector as one small cluster while merging birds and background in a second big cluster. By contrast, the robustly estimated Fiedler vector using the proposed RRLPI method provides the correct structure in the Fiedler vector estimate to enable the desired segmentation into birds and background. Robustness is obtained by assigning weights to all data-points based on the typical weighted node degree in the associated graph. Consistent with the ideas of M-estimation in robust statistics, a large degree of outlyingness of a data-point corresponds to a small weight.

The paper is organized as follows. Section II introduces the basic concepts and briefly discusses Fiedler vector estimation using LPI. The ideas underlying the proposed algorithm and the problem formulation are the subject of Section III.

Section IV is dedicated to the proposed robust regularized locality preserving indexing method. This section includes the theoretical analysis, penalty parameter selection, computational complexity analysis and possible applications. Section V demonstrates the performance of the proposed approach in comparison to popular competitors both in cluster enumeration and in image segmentation using real-world data. Finally, conclusions are drawn in Section VI. An implementation of RRLPI is available at: https://github/A-Tastan/RRLPI

II. PRELIMINARIES

A. SUMMARY OF NOTATIONS

Lower and upper-case bold letters denote vectors and matrices, respectively; $\mathbb{R}$ denotes the set of real numbers; $\mathbb{Z}^+$ denotes the set of positive integers; $|x|$ denotes the absolute value of x; $\|\mathbf{x}\|$ denotes the norm of vector $\mathbf{x}$, e.g. $\|\mathbf{x}\|_2$ is the ℓ_2 norm; $\text{med}(\mathbf{x})$ denotes the median of vector $\mathbf{x}$; $\text{sign}(x)$ denotes the sign function defined as $\text{sign}(x) = x/|x|$; $\text{diag}(x_1, \ldots, x_N)$ denotes a diagonal matrix of size $N \times N$ with $x_1, \ldots, x_N$ on its diagonal; $\mathbf{x}^\top$ denotes transpose of vector $\mathbf{x}$; λ_F denotes the Fiedler value; $\mathbf{y}_F$ denotes the Fiedler vector corresponding to λ_F; $\mathbf{I}$ denotes the identity matrix; $\mathbf{1}$ denotes the vector of ones; $\hat{\mathbf{x}}$ denotes the estimate of vector $\mathbf{x}$; $\tilde{\mathbf{W}}$ refers to a corrupted affinity matrix; d_i denotes the weighted node degree of the ith data-point for a weighted affinity matrix and the corresponding degree for an adjacency matrix; λ_i denotes the ith eigenvalue; $y_i(j)$ denotes the embedding result of the jth data-point in the eigenvector $\mathbf{y}_i$ associated with the ith eigenvalue λ_i.

B. PROBLEM STATEMENT

Given a set of data-points $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N] \in \mathbb{R}^{M\times N}$, the aim of this work is to estimate the Fiedler vector $\mathbf{y}_F \in \mathbb{R}^N$ such that it embeds each data-point on a real line, providing robustness at a reasonable computation cost. In the following sections, the applicability of existing projection strategies to Fiedler vector estimation is explained in terms of their theoretical adaptiveness, computational cost and robustness.

C. COMPUTING THE FIEDLER VECTOR

Suppose that a data set $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N] \in \mathbb{R}^{M\times N}$ with M denoting the data dimension and N being the number of data-points, can be represented as a graph $G = \{V, E, \mathbf{W}\}$, where V denotes the vertices, E represents the edges, and $\mathbf{W} \in \mathbb{R}^{N\times N}$ is the symmetric affinity matrix. The i, jth coefficient of the affinity matrix $w_{i,j} \in \mathbf{W}$ representing the edge weight between ith and jth vertex can be computed using a similarity measure, such as the cosine similarity, for which $w_{i,j} = \mathbf{x}_i^\top \mathbf{x}_j$, $i \neq j$ s.t. $\|\mathbf{x}_i\|_2 = 1$, $\|\mathbf{x}_j\|_2 = 1$. Let $\mathbf{L} \in \mathbb{R}^{N\times N}$ denote the discrete Laplacian operator (also known as the graph Laplacian matrix) that is nonnegative definite with associated eigenvalues $0 \leq \lambda_0 \leq \lambda_1 \leq \ldots \leq \lambda_{N-1}$ sorted in ascending order. Then, it follows that the Fiedler vector $\mathbf{y}_F \in \mathbb{R}^N$ is the eigenvector associated with the second smallest eigenvalue λ_1 of the eigen-problem

$$\mathbf{L}\mathbf{y}_i = \lambda_i \mathbf{y}_i, \tag{1}$$

or in a generalized eigenvalue problem form

$$\mathbf{L}\mathbf{y}_i = \lambda_i \mathbf{D}\mathbf{y}_i, \tag{2}$$

Here, the Laplacian matrix $\mathbf{L}$ is defined analogously to the Laplace Beltrami operator on the manifold by $\mathbf{L} = \mathbf{D} - \mathbf{W}$, where $\mathbf{D} \in \mathbb{R}^{N\times N}$ is a diagonal weight matrix with weighted node degrees $d_{i,i} = \sum_j w_{i,j}$ on the diagonal and $\mathbf{y}_i \in \mathbb{R}^N$ is the eigenvector associated with λ_i.

The LPI method determines linear approximations to the eigenfunctions of the Laplace Beltrami operator [15], [16] by representing the Fiedler vector as the response of a linear regression with input variables $\mathbf{X}$, i.e., $\mathbf{y}_F = \mathbf{X}^\top \boldsymbol{\beta}_F$. Hence, the LPI finds a transformation vector $\boldsymbol{\beta}_F \in \mathbb{R}^M$ that is the eigenvector associated with the second largest eigenvalue of the generalized eigen-problem

$$\mathbf{X}\mathbf{W}\mathbf{X}^\top \boldsymbol{\beta}_j = \lambda_j \mathbf{X}\mathbf{D}\mathbf{X}^\top \boldsymbol{\beta}_j, \tag{3}$$

which has the same eigenvalue λ_i as in (2) for λ_j with $j = N - (i+1)$. This means that the Fiedler value λ_F which is the second smallest eigenvalue $(\lambda_i = \lambda_1)$ of the eigen-problem in (2) is the same as the second largest eigenvalue $(\lambda_j = \lambda_{N-2})$ of the eigen-problem in (3) and thus, the Fiedler vector $\mathbf{y}_F$ corresponding to λ_F can be computed based on the projective functions of LPI. However, the LPI [15] method performs a complete SVD to compute the eigenvectors, which results in a cubic-time complexity even though the matrix is sparse. This property of LPI makes it infeasible in applications comprising large number of samples.

D. ESTIMATING THE FIEDLER VECTOR

The Regularized Locality Preserving Indexing (RLPI) [17] method efficiently determines the projective functions of LPI in two consecutive steps for the Fiedler vector estimation. First, the Fiedler vector $\mathbf{y}_F$ associated with the second smallest eigenvalue of (2) must be computed. Then, for the Fiedler vector $\mathbf{y}_F$, the a transformation vector $\boldsymbol{\beta}_F \in \mathbb{R}^M$ that satisfies $\mathbf{y}_F = \mathbf{X}^\top \boldsymbol{\beta}_F$ can be estimated by solving the following least-squares problem

$$\hat{\boldsymbol{\beta}}_F = \underset{\boldsymbol{\beta}_F}{\text{argmin}} \sum_{i=1}^{N} \left(\boldsymbol{\beta}_F^\top \mathbf{x}_i - y_F(i)\right)^2, \tag{4}$$

where $y_F(i)$ is the ith embedding point in $\mathbf{y}_F$ and $\hat{\boldsymbol{\beta}}_F$ is the estimated transformation vector. The RLPI [17] method imposes a penalty on the norm of $\boldsymbol{\beta}_F$ in (4) to control the amount of shrinkage. Under certain conditions,[1] the estimated Fiedler vector based on RLPI is the eigenvector of eigen-problem in (3).

[1] Let $\mathbf{y}_F$ be the Fiedler vector associated with the second largest eigenvalue of the eigen-problem in (3). If $\mathbf{y}_F$ is in the space spanned by row vectors of the data matrix $\mathbf{X}$, the corresponding projective function $\boldsymbol{\beta}_F$ calculated by RLPI will be the eigenvector of eigen-problem in (3) as the penalty parameter decreases to zero (for details, see Theorem 2 in [17]).

In contrast to LPI, the above formulation can benefit from sparsity of the affinity matrix and leading to a quadratic-time complexity of the unsupervised RLPI [17] method for the setting of sparse matrices. Despite its computational advantage, a major drawback of the RLPI algorithm, however, is that the penalty parameter, which has a significant impact in the performance of RLPI, is unknown in real-world scenarios. Additionally, LPI [15] and RLPI [17] do not address robustness which is important drawback considering that real-world data sets often contain impulsive noise and outliers.

To integrate robustness in eigenvector estimation, RLPFM [32] formulates the problem in (4) as a ridge regression task that is solved with a penalized M-estimation approach. In particular, the RLPFM method iteratively reweights the residuals of Laplacian eigenmaps-based prediction to estimate the vector $\hat{\boldsymbol{\beta}}_F$ that provides the best approximation to the Fiedler vector $\mathbf{y}_F$. Even though the method promotes robustness, it leads to an additional cost in comparison to RLPI [17]. In particular, in every iteration, the method has a cubic-time complexity for dense matrices while it has quadratic-time complexity for sparse matrices. This means that the method becomes infeasible when large numbers of iterations are necessary for the Fiedler vector estimation.

III. MOTIVATION

The previous section discussed the applicability of LPI for Fiedler vector computation. In particular, LPI may discover the hidden nonlinear structure by finding linear approximations to the nonlinear Laplacian eigenmaps (for details, see [15] and [16]). However, when using the least-squares objective function, outliers and heavy-tailed noise may have a large impact on the estimation of the transformation vector $\boldsymbol{\beta}_F$. This leads to errors in the computed Fiedler vector and, consequently, an information loss about the representation of the underlying graph structure using such a corrupted Fiedler vector computation. This section analyzes the effects of outliers and noise on the eigen-decomposition of the Laplacian matrix. The analysis provides the theoretical basis and an understanding of the ideas underlying the proposed robust Fiedler vector estimation approach.

A. OUTLIER EFFECT ON EIGEN-DECOMPOSITION

The effect of outliers on the eigen-decomposition is analyzed in terms of two fundamental types of outliers. All examinations that are conducted in this section are made for block affinity matrices unless it is stated otherwise. Motivated by [47], we begin by defining the first fundamental type of outliers as follows.

Definition III.1 (Type I Outliers): The data-points that do not have correlations with any of the samples are called Type I outliers.

Based on this first definition, the correlation coefficient vectors that are associated to outliers, ideally, are zero vectors. More practically speaking, and motivated by real data examples, the data-points whose similarity coefficients have negligibly small values may also be called Type I outliers.

To understand Type I outliers' effect on the Fiedler vector, it is important to remember the relationship between the number of connected components of a k block diagonal affinity matrix $\mathbf{W} \in \mathbb{R}^{N \times N}$ and the spectrum of the associated graph Laplacian matrix $\mathbf{L} \in \mathbb{R}^{N \times N}$. In [48], it has been shown that the multiplicity of the zero-valued eigenvalues of $\mathbf{L}$ equals the number of connected components k and the eigenspace of the zero eigenvalues is spanned by the indicator vectors of those connected components. Clearly, considering Type I outliers as isolated blocks of size one, the addition of N_{I} number of Type I outliers leads to N_{I} additional zero-valued eigenvalues whose eigenspace is spanned by the indicator vectors of the N_{I} outliers. Accordingly, if the affinity matrix has distinct blocks and the Type I outliers are disconnected, the Fiedler vector can be easily determined after removing these outliers. However, in real world scenarios the true blocks are generally not distinct and/or outliers do have a few non-zero similarities which result in non-zero eigenvalues [35], [36]. Since the number of blocks k and the number of outliers N_{I} are unknown, directly using eigenvalues for outlier detection may be impossible in practice.

The following proposition provides a numerical understanding of Type I outliers' effect on the Fiedler vector.

Proposition III.1: For a definite nonnegative k block zero-diagonal symmetric affinity matrix $\mathbf{W} \in \mathbb{R}^{N \times N}$ and the associated Laplacian $\mathbf{L} \in \mathbb{R}^{N \times N}$, let the eigenvectors be orthogonal vectors of different lengths, i.e., $\|\mathbf{y}_i\|_2 = L_i$, where $\mathbf{y}_i$ denotes the eigenvector associated with the ith eigenvalue of $\mathbf{L}$ and L_i is the length of $\mathbf{y}_i$. Further, let $\tilde{\mathbf{y}}_F$ be the Fiedler vector associated with the eigenvalue that corresponds to an additive Type I outlier and let $\tilde{y}_F(o_{\mathrm{I}})$ denote the embedding result of a Type I outlier in $\tilde{\mathbf{y}}_F$. Then, it follows that the Euclidean distance between embeddings of different blocks decreases to zero when the absolute value of the embedding result of the outlier increases to the length of the Fiedler vector, i.e. when $|\tilde{y}_F(o_{\mathrm{I}})| \to L_F$.

Proof: See Appendix A.1 of the supplementary material. ■

Motivated by the eigenvectors' crucial role in cluster analysis, the results of Proposition III.1 can be extended to multiple eigenvectors. If a distance-based clustering approach, such as, spectral clustering is applied on the eigenvectors that are the indicator vectors of N_{I} outliers, all non-outlying points are mapped to the same cluster as a result of Proposition III.1, as $|\tilde{y}_F(o_{\mathrm{I}})| \to L_F$. This explains why spectral clustering breaks down in the presence of Type I outliers. Only if the number of outliers is known or can be deduced from the data set (e.g. because they perfectly match Definition III.1), ignoring the indicator vectors of outliers can overcome this problem. However, in practice, Definition III.1 may only hold approximately, and data-points may vary in the degree of their outlyingness, making a binary detection challenging or inappropriate. In Section IV, we will present a robust M-estimation-based approach to suppress the impact of outliers, when such simple outlier detection and removal strategies do not apply.

Next, we will study the effect of a second fundamental type of outliers that are defined as follows.

Definition III.2 (Type II Outliers): Outliers of Type II are the data-points that have correlations with more than one group of data-points.

The definitions of Type I and Type II outliers are based on the assumed ideal graph model in which all vertices associated with different clusters are unconnected. This model is denoted as "ideal case" in the well-known spectral clustering algorithm of Ng, Jordan and Weiss (NGW) (for details, see Section III-A in [27]). However, in real-world scenarios the data-points do not behave ideally (for further discussion, see see Section III-B in [27]) and even a small deviation from the "ideal case" leads to the presence of Type II outliers. Note that what is referred to as the "general case" in [27], may include both Type I and II outliers, since they are defined as deviations from the "ideal case". Unlike for Type I outliers, simple detection and removal strategies, for example, outlier detection based on zero-eigenvalues are not applicable due to the undesired correlations [48]. In addition to their effect on the eigenvalue structure, it is therefore important to analyze the effect of Type II outliers on the eigenvectors to understand their particular effect on the Fiedler vector. Since Type II outliers may occur in large numbers in practical settings, the extreme case of correlation with more than one group of data-points is examined in the following proposition.

Proposition III.2: For a k block zero-diagonal symmetric nonnegative affinity matrix $\mathbf{W} \in \mathbb{R}^{N\times N}$, let $w_i \in \{w_1, w_2, \ldots, w_k\}$ denote a constant around which the correlation coefficients of the ith block are assumed to be concentrated with negligibly small variations. Further, let $\tilde{w}_u$ denote a constant around which the correlations between blocks are concentrated. Let $\tilde{\mathbf{W}}$ define an affinity matrix, which is equal to $\mathbf{W}$, except that we impose $\tilde{w}_u > 0$, such that the vertices associated with ith and jth block become connected. Then, it follows that the connections between vertices corresponding to different blocks result in embedding all data-points onto the same location on the eigenvector $\tilde{\mathbf{y}}_0$ that is associated with the smallest eigenvalue $\tilde{\lambda}_0$ of the Laplacian matrix $\tilde{\mathbf{L}} \in \mathbb{R}^{N\times N}$ corresponding to $\tilde{\mathbf{W}}$.

Proof: See Appendix A.2 of the supplementary material. ■

Even though Proposition III.2 shows the loss of group structure in the eigenvector associated with the smallest eigenvalue, in real applications, this eigenvector might be the Fiedler vector when the data includes Type I outliers with negligibly small similarity coefficients. A strategy to deal with Type II outliers' effects, which are described in Proposition III.2, is the consideration of the group of eigenvectors spanning the subspaces instead of unstable individual eigenvectors of $\mathbf{L}$ [27]. The NGW algorithm, normalizes the selected orthogonal eigenvectors to have unit length and performs k-means clustering. Considering the eigen-problem in (2) as in [27], the following proposition extends the analysis of the effect of outliers on group structure for multiple eigenvectors that are normalized to have constant length.

Proposition III.3: Let $\mathbf{L} \in \mathbb{R}^{N\times N}$ and $\tilde{\mathbf{L}} \in \mathbb{R}^{N\times N}$ denote Laplacian matrices associated with $\mathbf{W} \in \mathbb{R}^{N\times N}$ and $\tilde{\mathbf{W}} \in \mathbb{R}^{N\times N}$, respectively. Further, let the eigenvectors associated with the K smallest eigenvalues of $\mathbf{L}$ and $\tilde{\mathbf{L}}$, respectively, be the column vectors of the matrices $\mathbf{Y} \in \mathbb{R}^{N\times K}$ and $\tilde{\mathbf{Y}} \in \mathbb{R}^{N\times K}$ where K denotes the number of clusters. Finally, let $\mathbf{e}_i \in \mathbb{R}^K$ and $\tilde{\mathbf{e}}_i \in \mathbb{R}^K$, the ith row vector of $\mathbf{Y}$ and $\tilde{\mathbf{Y}}$, respectively, denote the embedding vectors that represent the M-dimensional ith feature vector in the reduced K-dimensional space. Assuming that the column vectors of $\mathbf{Y}$ and $\tilde{\mathbf{Y}}$ have a constant length based on the eigen-problem in (2), the Euclidean distance between any embedding vector pair $\mathbf{e}_i$ and $\mathbf{e}_j$ associated with different blocks is greater than that of $\tilde{\mathbf{e}}_i$ and $\tilde{\mathbf{e}}_j$, i.e. $\|\mathbf{e}_i - \mathbf{e}_j\|_2 > \|\tilde{\mathbf{e}}_i - \tilde{\mathbf{e}}_j\|_2$.

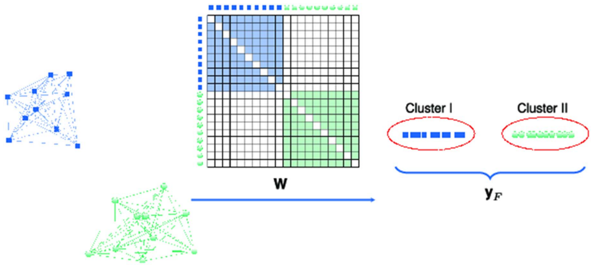


FIGURE 2. Fiedler vector computation for an ideal $k = 2$ blocks affinity matrix.

Proof: See Appendix A.3 of the supplementary material. ■

Proposition III.3 shows that the Type II outliers' effect is not limited to an individual eigenvector and this effect becomes more visible when the eigenvectors have different lengths and the length of the eigenvector associated with the smallest eigenvalue is greater than the others. In such cases, normalizing eigenvectors as in [27] might be helpful to relatively preserve the distance between embeddings of different clusters. However, this may also increase the negative impact of outliers. For example, Type I outliers increase to the length of the eigenvector as in Proposition III.1. Considering the prominent effect of Type I outliers in comparison to outliers of Type II, the proposed method considers a more general case based on the eigenvectors of different lengths as in the spectral clustering method of Meilă-Shi [49] (for a detailed comparisons about spectral clustering methods, see [50]).

To illustrate the different outlier effects, examples of computed Fiedler vectors are shown for ideal and corrupted affinity matrices in Fig. 2 and Fig. 3, respectively. In the ideal case, the vertices of different clusters do not have edges between each other while vertices of the same cluster are connected with strong edges. If such an ideally clustered graph is embedded on the real line using the Fiedler vector, the vertices of the same cluster are concentrated while they are far away from the vertices of a different cluster, see Fig. 2. Therefore, the embedding results of different clusters are easily separable which is crucial for subsequent graph partitioning. On the other hand, the corrupted graph in Fig. 3 includes two typical outlier effects. Based on Propositions III.1–III.3, Type I outliers are embedded far from the clusters while Type II outliers that correlate with more than one cluster are embedded between different clusters making their separation difficult. In

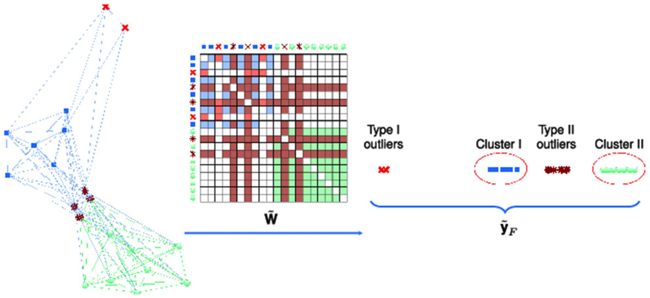
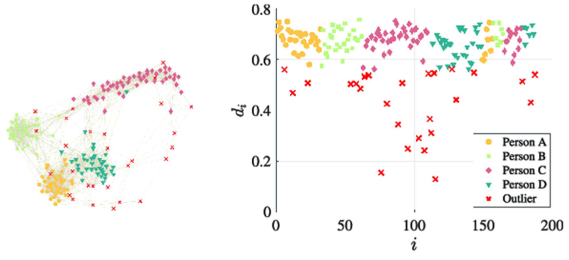
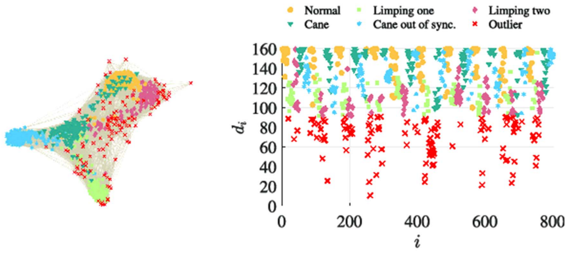


FIGURE 3. Fiedler vector computation for a corrupted $k = 2$ blocks affinity matrix. The corruptions of the affinity matrix by Type I and Type II outliers are highlighted by coloring the corresponding affected elements in light red and dark red, respectively. In the Fiedler vector, outliers are positioned as shown in the right illustration.



(a) Sparse graph model (b) weighted node degrees

FIGURE 4. Examplary outlyingness measure of Person Identification [51] data set based on the weighted node degrees.



(a) Sparse graph model (b) weighted node degrees

FIGURE 5. Examplary outlyingness measure of Human Gait [52] data set based on the weighted node degrees.

such scenarios, the outliers result in a performance degradation because of the computed Fiedler vector that would lead to losing the group structure information of the graph.

B. OUTLYINGNESS MEASURE: WEIGHTED NODE DEGREE

This section introduces and discusses the weighted node degree, an outlyingness measure to suppress both Type I and Type II outliers. Definitions III.1 and III.2, respectively, determine the Type I and Type II based on the assumed (ideal) graph model which comprises a considerable number of intra-cluster edges, while not containing inter-cluster edges. In particular, Definition III.1 determines a Type I outlier as a relatively unconnected (or ideally even totally unconnected) vertex that has noticeably small-valued (or ideally zero-valued) weighted node degree in comparison to the non-outlying samples of the clusters. A natural result of Definition III.1 based on the assumed graph model is that Type I outliers' weighted node degrees will be smaller-valued than the typical ones when the intra-cluster edge weights are large-valued. Even though the values of edge weights rely on the graph construction, large-valued edge weights for the intra-clusters are typical in real world scenarios including the densely connected clusters. Therefore, the results of Definition III.1 point out a frequently observed scenario. In contrast to this comparably simple characterization of Type I outlyingness, Type II outlyingness determination based on weighted node degree depends on cluster sizes in addition to the graph construction. For example, let $\tilde{G} = \{V, E, \tilde{\mathbf{W}}\}$ be a Type II outlier corrupted graph model of balanced cluster sizes. According to Definition III.2, the weighted node degree of a Type II outlier is smaller-valued than that of the typical data-points when the Type II outlier is connected to multiple groups of nodes with small-valued edge weights in $\tilde{G}$. On the other hand, the weighted node degree of a Type II outlier is larger-valued than that of the typical data-points when the intra-cluster edge weights of the Type II outliers are equal or larger-valued than that of the typical data-points in $\tilde{G}$. Thus, while both outlier types behave differently, it is important to note for a connected graph model of comparable cluster sizes both types of outliers have a common characteristic: *their weighted node degrees deviate from that of the typical nodes.*

Type I and Type II outliers' deviating weighted node degree characteristic is commonly observed in many real-world data sets as their existence directly depends on the graph construction where the appropriate level of sparsity is unknown. In particular, the increase in the level of sparsity generates Type I and Type II outliers when the edge weights within the clusters are larger valued than between clusters, which is a common scenario. In contrast to this, a dense graph construction may lead to an increase in the number of Type II outliers with larger valued weighted node degree. To provide a visual understanding of this, exemplary outlier assignments are shown for sparse graph models of the Person Identification [51] and Human Gait [52] real-world data sets in Figs. 4 and 5, respectively. In Figs. 4(a) and 5(a), the red crosses depict the outliers that include the 15% of vertices whose weighted node degrees deviate maximally from the median of weighted node degrees (the median represents the typical weighted node degrees). As can be seen, the outlier assignment based on weighted node degree captures both vertices between different clusters (Type II outliers) and the vertices that are far from every cluster (Type I outliers).

The real data examples confirm the theoretical analysis that the weighted node degree is an informative measure for the determination of outliers. However, since the number of outliers is unknown in real-world scenarios a binary outlier detection based on weighted node degrees may result in an information loss. Therefore, in this work, robustness is introduced

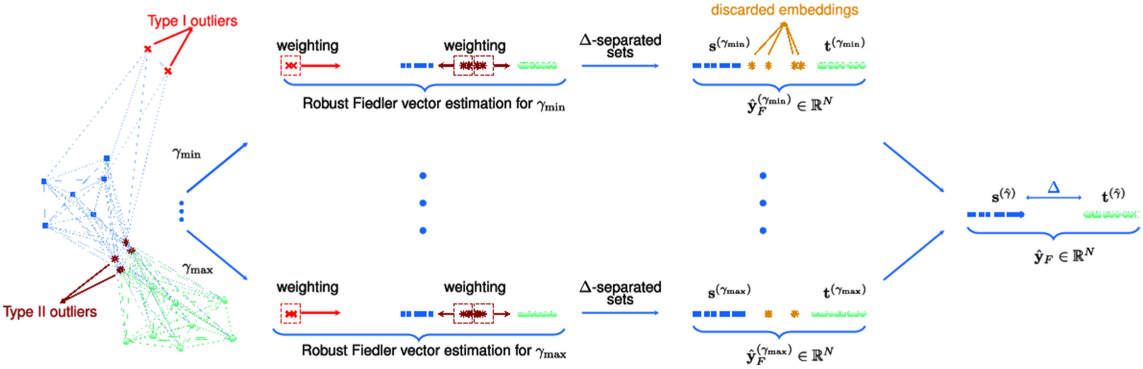

Type I outliers
weighting
weighting
Δ-separated sets
discarded embeddings
$\mathbf{s}^{(\gamma_{\min})}$
$\mathbf{t}^{(\gamma_{\min})}$
Robust Fiedler vector estimation for $\gamma_{\min}$
$\hat{\mathbf{y}}_F^{(\gamma_{\min})} \in \mathbb{R}^N$
$\gamma_{\min}$
$\gamma_{\max}$
Type II outliers
weighting
weighting
Δ-separated sets
$\mathbf{s}^{(\gamma_{\max})}$
$\mathbf{t}^{(\gamma_{\max})}$
Robust Fiedler vector estimation for $\gamma_{\max}$
$\hat{\mathbf{y}}_F^{(\gamma_{\max})} \in \mathbb{R}^N$
$\mathbf{s}^{(\hat{\gamma})}$
Δ
$\mathbf{t}^{(\hat{\gamma})}$
$\hat{\mathbf{y}}_F \in \mathbb{R}^N$

FIGURE 6. High-level flow diagram illustrating the main steps of RRLPI using a generic example with $k = 2$ clusters.

by suppressing outliers' negative impact on the Fiedler vector rather than detecting and eliminating them. In this way, we allow for some uncertainty in our decision, giving moderate but non-zero weight to the points that we are not sure about.

IV. ROBUST FIEDLER VECTOR ESTIMATION

This section introduces the proposed Robust Regularized Locality Preserving Indexing method for Fiedler vector estimation. To understand the key ideas of proposed robust Fiedler vector estimation, a high-level flow diagram illustrating the fundamental steps is given in Fig. 6. Starting from a graph model that is assumed to be corrupted by Type I and Type II outliers, for every candidate penalty parameter, the proposed RRLPI method estimates the Fiedler vector robustly by down-weighting the deviating embedding points based on their weighted node degrees. After projecting graph vertices onto the real line based on robust Fiedler vector estimation, the penalty parameter selection building upon the geometric analysis of *well-separated ℓ_2^2 representations* is performed. In particular, every robust Fiedler estimate $\hat{\mathbf{y}}_F^{(\gamma_i)} \in \mathbb{R}^N$ associated with a candidate penalty parameter γ_i is separated into two subsets $\mathbf{s}^{(\gamma_i)}$ and $\mathbf{t}^{(\gamma_i)}$. Then, the embedding points between the two generated subsets are discarded as long as the subsets being Δ-separated. The penalty parameter providing the minimum number of discarded points is used to determine the final robust Fiedler vector estimate. In the following sections, a step-by-step detailed methodology description of the proposed method as well as its theoretical and computational analysis are presented.

A. RRLPI FOR FIEDLER VECTOR ESTIMATION

Let data matrix $\mathbf{X}$ be subject to heavy tailed noise and outliers that obscure the underlying group structure in the graph $G = \{V, E, \mathbf{W}\}$ that represents $\mathbf{X}$. In the previous section, it was shown that the weighted node degree attached to a vertex is a valuable characteristic of an outlier because it significantly differs from the typical weighted node degree. Thus, the weighted node degree of attached to vertex i is modeled as

$$d_i = d_{\text{typ}} + \epsilon_i. \tag{5}$$

Here, $d_i = \sum_j^N w_{i,j}$ and ϵ_i, respectively, denote the weighted node degree and the error term for the ith vertex, d_{typ} is the typical weighted node degree of the graph G. In practice a robust estimator, such as the median is used, i.e. $\hat{d}_{\text{typ}} = \text{med}(\mathbf{d})$ for a vector of weighted node degrees $\mathbf{d} = [d_1, \ldots, d_N]$. For Fiedler vector estimation, an error vector $\boldsymbol{\epsilon} \in \mathbb{R}^N$ is constructed using the error terms associated with each weighted node degree in $\mathbf{d}$. Then, the transformation vector $\boldsymbol{\beta}_F$ associated with $\mathbf{y}_F$ is computed using penalized ridge regression M-estimation [53] by solving the following zero gradient equation

$$-\sum_{i=1}^{N} \psi\left(\frac{\epsilon_i}{\hat{\sigma}}\right)\left(\frac{\mathbf{x}_i^\top}{\hat{\sigma}}\right) + \gamma \boldsymbol{\beta}_F = \mathbf{0}, \tag{6}$$

where γ denotes the penalty parameter, $\hat{\sigma}$ is a robust scale estimate of $\boldsymbol{\epsilon}$ and ψ is a bounded and continuous odd function called the score-function. A popular M-estimator is defined by Huber's loss function which is convex and piece-wise differentiable. Huber's score function is of the form

$$\psi\left(\frac{\epsilon_i}{\hat{\sigma}}\right) = \begin{cases} \frac{\epsilon_i}{\hat{\sigma}}, & \text{for } \left|\frac{\epsilon_i}{\hat{\sigma}}\right| \leq c \\ c \cdot \text{sign}\left(\frac{\epsilon_i}{\hat{\sigma}}\right), & \text{for } \left|\frac{\epsilon_i}{\hat{\sigma}}\right| > c \end{cases}. \tag{7}$$

where c, commonly set to a default value of $c = 1.345$ for 95% asymptotic relative efficiency (ARE), is the tuning parameter that trades off robustness against outliers and ARE under a Gaussian distribution model for ϵ (see [54] for a detailed discussion). A frequently used robust scale estimate $\hat{\sigma}$ is the normalized median absolute deviation [54] that is defined by

$$\hat{\sigma} = \text{madn}(\boldsymbol{\epsilon}) = 1.4826 \cdot \text{med}|\boldsymbol{\epsilon} - \text{med}(\boldsymbol{\epsilon})|. \tag{8}$$

The motivation for adopting M-estimation for Fiedler vector estimation is that a bounded score function, such as Huber's, ensures that nodes with atypical edge weights are down-weighted in (6). RRLPI, therefore, softly suppresses the negative impact of outliers on the Fiedler vector estimate based on Huber's function. This becomes intuitively clear

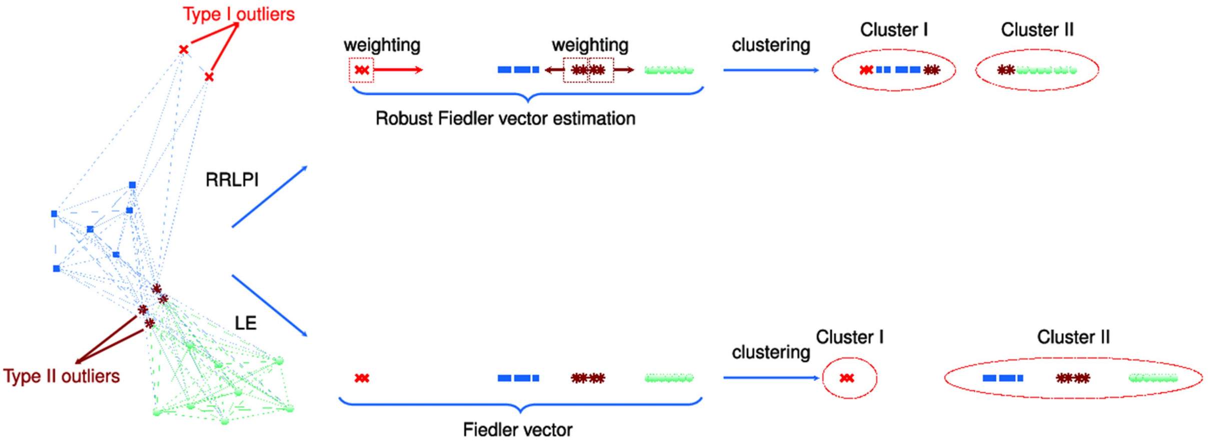


FIGURE 7. Examplary plot of the Fiedler vector computation based on LE and RRLPI methods. The weighting operation in RRLPI on Type I and Type II outliers results in two clusters of concentrated mappings that include the outliers. In this way, the true structure of the non-outlying data becomes visible, even in the presence of outliers. By contrast, for LE, the outliers deteriorate the underlying two-cluster structure. Further, the weights provide a robust measure of outlyingness, which may be used to detect and analyze outliers, which is of high interest in some applications.

when considering Huber's weight function $\omega_i = \omega(\frac{\epsilon_i}{\hat{\sigma}})$:

$$\omega\left(\frac{\epsilon_i}{\hat{\sigma}}\right) = \begin{cases} 1, & \text{for} \quad \left|\frac{\epsilon_i}{\hat{\sigma}}\right| \le c \\ c/\left|\frac{\epsilon_i}{\hat{\sigma}}\right|, & \text{for} \quad \left|\frac{\epsilon_i}{\hat{\sigma}}\right| > c, \end{cases} \tag{9}$$

that gives constant weight up to c and then increasingly down-weights outliers by smoothly descending towards zero. Under some conditions, e.g. when outliers are extremely large valued, a different weighting function instead of Huber's may be used to completely down-weight extreme outliers. For example, in robust statistics, Tukey's weight function

$$\omega\left(\frac{\epsilon_i}{\hat{\sigma}}\right) = \begin{cases} \left(1-\left(\left|\frac{\epsilon_i}{\hat{\sigma}}\right|/c\right)^2\right)^2, & \text{for} \quad \left|\frac{\epsilon_i}{\hat{\sigma}}\right| \le c \\ 0, & \text{for} \quad \left|\frac{\epsilon_i}{\hat{\sigma}}\right| > c \end{cases}, \tag{10}$$

is a popular choice, which gives zero-weight to extreme outliers. However, such a function leads to non-convex optimization problems, which is why, in many cases, Huber's weights are preferred (see [54] and [55] for a detailed discussion).

To provide an intuitive understanding, an examplary plot is provided in Fig. 7 that compares RRLPI with LE. Consistent with the outlier effect analysis that has been detailed in Section III, the data-points which are mapped far away from any other cluster in the Fiedler vector $\mathbf{y}_F$ are Type I outliers while the embeddings between different clusters are outliers of Type II. As can be seen, these outlying embeddings result in a performance degradation for clustering algorithms that are based upon a non-robust computation of $\mathbf{y}_F$. An important property of these outliers is that their occurrence depends on the determined level of sparsity. In more details, a non-sparse graph model results in over-connected vertices (Type II) whereas increasing sparsity redundantly results in less-connected or disconnected vertices (Type I). A robust sparsity level determination, such as proposed in Section IV-C, is therefore essential. Since the number of outliers is unknown, an outlier detection based on the available graph structure may result in misdetection or losing information. Therefore, instead of outlier detection, RRLPI down-weights the deviating embedding points based on their weighted node degrees to achieve a robust Fiedler vector estimate in which the embeddings associated with the same cluster are concentrated while being separated from embeddings of different clusters. The RLPFM approach in [32] also maintains the ℓ_2 norm and builds upon M-estimation to suppress the outliers. However, it performs M-estimation of the eigenvectors by iteratively reweighting the residuals of Laplacian eigenmaps-based prediction, which results in an increased computation cost. In contrast to this well-known M-estimation application, robustly estimating the Fiedler vector based on the typical weighted node degree of the graph fits the weighting operation into a single-step and makes the proposed RRLPI computationally efficient in comparison to [32]. In addition to this, the proposed method introduces a comprehensive theoretical analysis and step-by-step detailed computational complexity evaluation for integrating M-estimation into Fiedler vector estimation in the following sections.

B. THEORETICAL ANALYSIS

The RLPI [17] method represents the Fiedler vector $\mathbf{y}_F$ as the response of a linear regression with input variables $\mathbf{X}$, i.e. $\mathbf{y}_F = \mathbf{X}^\top \boldsymbol{\beta}$. Then, it determines the transformation vector $\boldsymbol{\beta}_F$ that minimizes a penalized residual sum of squares problem $\sum_{i=1}^{N}(\boldsymbol{\beta}_F^\top \mathbf{x}_i - y_F(i))^2 + \gamma\|\boldsymbol{\beta}_F\|^2$. In RRLPI, this approach is generalized by defining $\boldsymbol{\beta}_F$ as the solution to (6) whose matrix form leads to (for detailed information, see [53])

$$\hat{\boldsymbol{\beta}}_F = \left(\mathbf{X}\boldsymbol{\Omega}\mathbf{X}^\top + \gamma\sigma^2\mathbf{I}\right)^{-1}\mathbf{X}\boldsymbol{\Omega}\mathbf{y}_F, \tag{11}$$

where $\boldsymbol{\Omega} \in \mathbb{R}^{N\times N}$ is diagonal matrix of weights defined by (9).

As discussed in Section II, the LPI method uses a linearization of the embedding operation for the Fiedler vector computation. To understand the relationship between LPI and RRLPI, we first clarify the relation between RLPI and RRLPI.

Theorem 1: RRLPI is a robustly weighted RLPI [17], and for $\mathbf{\Omega} = \mathbf{I}$, it gives identical solutions to RLPI based Fiedler vector computation.

Proof: See Appendix B.1 of the supplementary material. ■

From Theorem 1, it follows that for $\gamma > 0$ and/or $\mathbf{\Omega} \neq \mathbf{I}$, the estimated tranformation vector $\hat{\boldsymbol{\beta}}_F$ is not the eigenvector of the eigen-problem in (3) which means that it is not associated with the Fiedler value. However, the following theorem shows in which cases $\boldsymbol{\beta}_F$ gives exactly the eigenvector of the eigen-problem in (3).

Theorem 2: Suppose $\mathbf{y}_F$ is the Fiedler vector associated with the second largest eigenvalue of the eigen-problem in (3). Further, let $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top\mathbf{\Omega}$ be the SVD of the data matrix where $\mathbf{\Sigma} \in \mathbb{R}^{\tau\times\tau}$ is a diagonal matrix of singular values, $\mathbf{U} \in \mathbb{R}^{M\times\tau}$ is the matrix of left singular vectors, $\mathbf{V} \in \mathbb{R}^{N\times\tau}$ is the matrix of right singular vectors for $\text{rank}(\mathbf{X}) = \tau$. Finally, let $\mathbf{\Omega} \in \mathbb{R}^{N\times N}$ and $\mathbf{\Psi} \in \mathbb{R}^{M\times M}$ be two weighting matrices such that $\mathbf{U}^\top\mathbf{\Psi}\mathbf{U} = \mathbf{I}$ and $\mathbf{V}^\top\mathbf{\Omega}\mathbf{V} = \mathbf{I}$. If $\mathbf{y}_F$ is in the space spanned by row vectors of the weighted data matrix $\mathbf{X}^*$, for $\mathbf{X}^* = \mathbf{X}\mathbf{\Omega}$, the corresponding transformation vector $\hat{\boldsymbol{\beta}}_F$ estimated with RRLPI is the eigenvector of the eigen-problem in (3) as γ decreases to zero.

Proof: See Appendix B.2 of the supplementary material. ■

Based on Theorem 2, the estimated transformation vector $\hat{\boldsymbol{\beta}}_F$ is the solution of (3) for $\gamma \to 0$, and $\mathbf{U}^\top\mathbf{\Psi}\mathbf{U} = \mathbf{I}$, $\mathbf{V}^\top\mathbf{\Omega}\mathbf{V} = \mathbf{I}$. In practice, the algorithm determines γ as $\gamma \to 0$ when the Fiedler vector $\mathbf{y}_F$ of the (2) is Δ-separated. In more details, the estimated transformation vector $\hat{\boldsymbol{\beta}}_F$ is the solution of (3) when the data set comprises clearly separable uncorrupted data-points providing Δ-separated embedding points and weighted node degrees that do not deviate from the typical weighted node degree. To understand the relationship between RRLPI and LPI, the results of this theorem are extended for all transformation vectors $\hat{\boldsymbol{\beta}}_i \in [\hat{\boldsymbol{\beta}}_0, \ldots, \hat{\boldsymbol{\beta}}_{N-1}]$ for the case that the data space M is greater than the number of data-points N and the data-points are linearly independent, i.e. $\text{rank}(\mathbf{X}) = N$.

Corollary 2.1: If the data-points are linearly independent, i.e. $\text{rank}(\mathbf{X}) = N$, all transformation vectors are solutions of (3) for $\gamma \to 0$, and $\mathbf{U}^\top\mathbf{\Psi}\mathbf{U} = \mathbf{I}$, $\mathbf{V}^\top\mathbf{\Omega}\mathbf{V} = \mathbf{I}$ which means that RRLPI is identical to LPI.

Proof: See Appendix B.3 of the supplementary material. ■

C. Δ-SEPARATED SETS FOR PENALTY PARAMETER SELECTION

This section introduces the key ideas of proposed penalty parameter selection algorithm from a clustering point of view. The graph partitioning problem, which can be interpreted as a clustering problem, is the separation of graph vertices into multiple clusters while minimizing the number of edges that cross the cut [46]. This problem is NP-hard [46], [56] and thus, most graph partitioning approaches are heuristic, e.g., [57],[58]. Different from existing heuristic graph partitioning approaches, the proposed penalty parameter selection is built upon the theoretical approximation algorithms for the sparsest cut, edge expansion, balanced separator and graph conductance problems (for details about these problems, see Section II in [46]). Arora, Rao and Varizani [46] has introduced a theoretical approximation for these problems based on the geometric analysis of *well-separated* ℓ_2^2 *representations*. In particular, an ℓ_2^2 representation, which embeds the vertices on a line such that the squared Euclidean distance form a metric, is called *well-separated* if every pair of points $s_i \in \mathbf{s}$ and $t_j \in \mathbf{t}$ are mapped at least $\Delta = \phi(1/\log^{-2/3} N)$ apart in ℓ_2^2 distance [46]. Inspired by *well-separated* ℓ_2^2 *representations*, we propose a penalty parameter selection algorithm by projecting graph vertices onto a real line using RRLPI-based Fiedler vector estimation such that every pair of two sets $s_i \in \mathbf{s}$ and $t_j \in \mathbf{t}$ is at least $\Delta = \phi(1/\log^{-2/3} N)$ apart in ℓ_2^2 distance for the estimated penalty paramater.

Let $\gamma_i \in \boldsymbol{\gamma}$ be the ith candidate penalty parameter in (11) from a given vector of candidate penalty parameters $\boldsymbol{\gamma} = [\gamma_{\min}, \ldots, \gamma_{\max}] \in \mathbb{R}^N$. Further, suppose that for each candidate penalty parameter γ_i, there exists an associated Fiedler vector estimate $\hat{\mathbf{y}}_F^{(\gamma_i)}$ that projects the graph vertices onto the real line. The geometric structure of *well-separated* ℓ_2^2 *representations* allows for designing the sets $\mathbf{s}$ and $\mathbf{t}$ by projecting the points on a random line such that, for a suitable constant κ, the points that are located on the left and right hand sides of κ are the initial candidates for the sets $\mathbf{s}$ and $\mathbf{t}$, respectively [46].

It has been shown (see, e.g. [59]) that it is possible to split a candidate Fiedler vector $\hat{\mathbf{y}}_F^{(\gamma_i)}$ into the two subsets $\mathbf{s}^{(\gamma_i)}$ and $\mathbf{t}^{(\gamma_i)}$ for $\kappa = 0$. Another possible option for κ is the median of embeddings such that $\kappa = \text{med}(\hat{\mathbf{y}}_F^{(\gamma_i)})$. From the definition of the Δ-separated sets, the projection subsets $\mathbf{s}^{(\gamma_i)}$ and $\mathbf{t}^{(\gamma_i)}$ associated with γ_i taking values between zero and one. Therefore, after selecting the members of the two sets $\mathbf{s}^{(\gamma_i)} \in \mathbb{R}^{N_\mathbf{s}}$ and $\mathbf{t}^{(\gamma_i)} \in \mathbb{R}^{N_\mathbf{t}}$ associated with γ_i, the final design of the sets $\mathbf{s}^{(\gamma_i)}$ and $\mathbf{t}^{(\gamma_i)}$ is performed using the rescaled estimated Fiedler vector $\bar{\mathbf{y}}^{(\gamma_i)}$ as

$$\mathbf{s}^{(\gamma_i)} = \left\{\bar{y}_F^{(\gamma_i)}(j) : \hat{y}_F^{(\gamma_i)}(j) > \kappa\right\}$$
$$\mathbf{t}^{(\gamma_i)} = \left\{\bar{y}_F^{(\gamma_i)}(j) : \hat{y}_F^{(\gamma_i)}(j) \leq \kappa\right\}. \qquad (12)$$

Here, $\hat{y}_F^{(\gamma_i)}(j)$ denotes the jth element of the estimated Fiedler vector $\hat{\mathbf{y}}_F^{(\gamma_i)}$ and $\bar{y}_F^{(\gamma_i)}(j)$ is the jth element of the rescaled estimated Fiedler vector $\bar{\mathbf{y}}_F^{(\gamma_i)}$. If the rescaled Fiedler vector $\bar{\mathbf{y}}_F^{(\gamma_i)}$ is not sufficiently sparse, it contains pairs of points $\bar{y}_F^{(\gamma_i)}(i) \in \mathbf{s}$ and $\bar{y}_F^{(\gamma_i)}(j) \in \mathbf{t}$ whose squared Euclidean distance is less than Δ. Thus, for a set of pairs of projections $\bar{y}_F^{(\gamma_i)}(i) \in \mathbf{s}$ and $\bar{y}_F^{(\gamma_i)}(j) \in \mathbf{t}$, a vector of discarded projections $\mathbf{r}^{(\gamma_i)} \in \mathbb{R}^{N_\mathbf{r}^{(\gamma_i)}}$ is designed as

$$\mathbf{r}^{(\gamma_i)} = \left\{\bar{y}_F^{(\gamma_i)}(i), \bar{y}_F^{(\gamma_i)}(j) : \|\bar{y}_F^{(\gamma_i)}(i) - \bar{y}_F^{(\gamma_i)}(j)\|_2^2 \leq \Delta\right\}, \quad (13)$$

Algorithm 1: Robust Fiedler Vector Estimation.

```
Input: A data matrix $\mathbf{X} \in \mathbb{R}^{M\times N}$ and an associated affinity
        matrix $\mathbf{W} \in \mathbb{R}^{N\times N}$, $N_{\min}$
for $\gamma_i = \gamma_{\min}, \ldots, \gamma_{\max}$ do
    Initialization:
      Evaluate the Fiedler vector $\mathbf{y}_F \in \mathbb{R}^N$ via (2)
      Compute $\boldsymbol{\beta}_F \in \mathbb{R}^M$ for $\mathbf{y}_F = \mathbf{X}^\top \boldsymbol{\beta}_F$
      RRLPI
      Update the error vector $\boldsymbol{\epsilon} \in \mathbb{R}^N$ using (5)
      Compute $\hat{\sigma}$ via (8)
      Calculate the weights $\omega_i = \omega(\frac{\epsilon_i}{\hat{\sigma}})$, $\boldsymbol{\Omega} = \mathrm{diag}(\boldsymbol{\omega})$
      Solve (11) and estimate $\hat{\boldsymbol{\beta}}_F^{(\gamma_i)}$
      Estimate the Fiedler vector for $\hat{\mathbf{y}}_F^{(\gamma_i)} = \mathbf{X}^\top \hat{\boldsymbol{\beta}}_F^{(\gamma_i)}$
      $\Delta$-separated sets
      Generate sets $\mathbf{s}^{(\gamma_i)}$ and $\mathbf{t}^{(\gamma_i)}$ via (12)
      Calculate $\|\bar{y}_F^{(\gamma_i)}(\min) - \bar{y}_F^{(\gamma_i)}(\max)\|_2^2$ s.t. $\bar{y}_F^{(\gamma_i)}(\min) \in \mathbf{s}$
      and $\bar{y}_F^{(\gamma_i)}(\max) \in \mathbf{t}$ and collect in a vector $\mathbf{z} \in \mathbb{R}^{N_\gamma}$
      while $N_{\mathbf{s}} \geq N_{\min}$ and $N_{\mathbf{t}} \geq N_{\min}$ do
        Create $\mathbf{r} \in \mathbb{R}^{N_{\mathbf{r}}^{(\gamma_i)}}$ using (13)
         Update $N_{\mathbf{r}}^{(\gamma_i)}$
         if $\mathbf{s}^{(\gamma_i)}$ and $\mathbf{t}^{(\gamma_i)}$ are $\Delta$-separated then
           break
         end
    end
    Collect $N_{\mathbf{r}}^{(\gamma_i)}$ into a vector $\mathbf{h} \in N_\gamma$
end
if at least one pair of $\Delta$-separated sets exist then
    Estimate $\hat{\gamma}$ using (14)
else
    Estimate $\hat{\gamma}$ using (15)
end
Estimate transformation vector $\hat{\boldsymbol{\beta}}_F^{\hat{\gamma}}$ in Eq. (11) for $\hat{\gamma}$
Estimate the Fiedler vector for $\hat{\mathbf{y}}_F^{(\hat{\gamma})} = \mathbf{X}^\top \hat{\boldsymbol{\beta}}_F^{(\hat{\gamma})}$
Output: A robust estimate of the Fiedler vector $\hat{\mathbf{y}}_F^{(\hat{\gamma})}$
```

as long as the two sets $\mathbf{s}^{(\gamma_i)}$ and $\mathbf{t}^{(\gamma_i)}$ have a reasonable number $N_{\min}$ of projections. The proposed strategy to estimate penalty parameter γ is, therefore, to minimize the number of discarded points i.e.,

$$\hat{\gamma} = \underset{\gamma_i=\gamma_{\min},\ldots,\gamma_{\max}}{\arg\min} \left\{ N_{\mathbf{r}}^{(\gamma_i)} \right\}, \tag{14}$$

where $N_{\mathbf{r}}^{(\gamma_i)}$ denotes the number of discarded projections for candidate penalty parameter γ_i, and $\hat{\gamma}$ is the estimated penalty parameter. In practice, there might not exist Δ-separated sets $\mathbf{s}^{(\gamma_i)}$ and $\mathbf{t}^{(\gamma_i)}$ for any candidate penalty parameter such that $\gamma_i \in \{\gamma_{\min}, \ldots, \gamma_{\max}\}$. For example, the sets might not be Δ-separated, although $N_{\mathbf{r}}^{(\gamma_i)}$ has reached its maximum value. Additionally, the initial sets may not satisfy $N_{\mathbf{s}} < N_{\min}$ or $N_{\mathbf{t}} < N_{\min}$. In such cases, the penalty parameter can alternatively be estimated by maximizing the squared Euclidean distance between the closely valued projections from the two

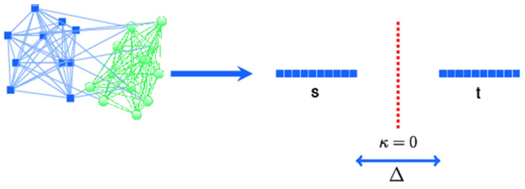


FIGURE 8. Example of Δ-separated sets s and t

sets $\mathbf{s}^{(\gamma_i)}$ and $\mathbf{t}^{(\gamma_i)}$,

$$\hat{\gamma} = \underset{\gamma_i=\gamma_{\min},\ldots,\gamma_{\max}}{\arg\max} \left\{ \left\| \bar{y}_F^{(\gamma_i)}(\min) - \bar{y}_F^{(\gamma_i)}(\max) \right\|_2^2 \right\}, \tag{15}$$

where $\bar{y}_F^{(\gamma_i)}(\min) \in \mathbf{s}$ and $\bar{y}_F^{(\gamma_i)}(\max) \in \mathbf{t}$ are the minimum and the maximum valued projections from the sets $\mathbf{s}^{(\gamma_i)}$ and $\mathbf{t}^{(\gamma_i)}$, respectively. The determined penalty parameter $\hat{\gamma}$ using (14) or (15) is then substituted in (11) to compute $\hat{\boldsymbol{\beta}}^{\hat{\gamma}}$ and the Fiedler vector estimate $\hat{\mathbf{y}}_F^{(\hat{\gamma})}$, finally, obtained for $\hat{\mathbf{y}}_F^{(\hat{\gamma})} = \mathbf{X}^\top \hat{\boldsymbol{\beta}}_F^{(\hat{\gamma})}$.

In terms of robustness, approaches based on Δ-separated sets in (14) are advantageous compared to directly using (15). In particular, (15) may maximize the distance between Type I outliers and true samples while the necessity of a reasonable number $N_{\min}$ of projections in different sets makes the Δ-separated sets robust against Type I outliers. Moreover, the Fiedler vector estimate $\hat{\mathbf{y}}_F^{(\gamma_i)}$ may contain Type II outliers which are embedded between true samples of sets $\mathbf{s}^{(\gamma_i)}$ and $\mathbf{t}^{(\gamma_i)}$. In such cases, usage of (15) may result in losing a good penalty parameter due to the Type II outliers that obscure the real distance between sets $\mathbf{s}^{(\gamma_i)}$ and $\mathbf{t}^{(\gamma_i)}$. In contrast, Δ-separated sets discard these Type II outliers up to a certain point and provide a robust measure of separation between the sets.

The main steps of the proposed Fiedler vector estimation are summarized in Algorithm 1 and an exemplary plot of Δ-separated sets is given in Fig. 8.

D. COMPUTATIONAL COMPLEXITY

As computational complexity is essential for the scalability of graph embedding techniques, the computational complexity of the proposed approach is analyzed in terms of its main operations. The computational complexity of operations is detailed using the term flam [60], which is a compound operation that includes one addition and one multiplication. For the cases when the complexity is not specified as flam, the Landau's big O symbol is used. The computational complexity of the proposed approach is given as follows:

Graph Construction: The pairwise cosine similarity which takes $\frac{1}{2}N^2M + 2NM$ as in [17] can be used for contructing graph G.

Initialization: For large eigen-problems, e.g. MATLAB uses a Krylov Schur decomposition [61]. The algorithm includes two main phases that are known as expansion and contraction. When N is larger than p, where p denotes the number of Lanczos basis vectors (preferably chosen as $p \geq 2q$

for q eigenvectors), the computational complexity of the algorithm can mainly be attributed to expansion and contraction phases. The expansion phase requires between $N(p^2 - q^2)$ flam and $2N(p^2 - q^2)$ flam. The contraction phase requires Npq flam [62].

Robust Regularized Locality Preserving Indexing (RRLPI): The proposed projection algorithm requires an estimate of scale that uses repetitive medians. The complexity of repetitive medians is $O(N)$ [63]. Further, for a densely connected matrix, the complexity is mainly attributed to the Cholesky decomposition which is of complexity $O(N^3)$ or, more specifically, $\frac{1}{6}N^3$ flam [60]. This complexity can be reduced to $O(N)$ using [64] if the matrix is rank deficient. If the matrix is sparse, the computation cost of decomposition can be reduced to $t(2Ns + 3N + 5M)$ flam using a least squares algorithm such as [65] where s denotes the average number of nonzero features and t is the number of iterations.

Δ-Separated Sets: To split the projection into two sets as $\mathbf{s}$ and $\mathbf{t}$, the vector $\mathbf{y}$ must be sorted which is of complexity $O(N \log N)$ and there are computationally efficient alternatives such as [66] for which the complexity is reduced to $O(N\sqrt{\log N})$. To compute Δ-separated sets, a maximum of N projections can be subtracted which means that this operation maximally takes N flam.

Summing up the terms with respect to flam yields minimally

$$t\left(2Ns + 3N + 5M\right) + \frac{1}{2}N^2M + N\left(2M + p^2 - k^2 + pk + 1\right)$$

flam. Hence, the complexity is of order $O(N^2)$. Based on the information that both $O(N)$ and $O(N \log N)$ are considerably smaller than $O(N^2)$, the minimum computational cost can be summarized as $O(N^2)$ for each candidate penalty parameter. Overall, the algorithm is, at least, of complexity $O(N_\gamma N^2)$ for a number N_γ of candidate penalty parameters.

E. EXAMPLE APPLICATIONS

Eigenvector decomposition has a large variety of applications, such as, dimension reduction [13], [14], [15], [16], [17], [18], [19], [20], [21], recognition [22], [23], [24] and localization [38]. Considering images as high-dimensional data sets, it is not surprising that eigen-decomposition is a fundamental research area also in image segmentation, e.g. [13], [14], [15], [16]. A frequently encountered problem is that the image is subject to noise, which may result in embedding noisy pixels far from the neighboring group of pixels in the embedding space and, consequently, losing the underlying structure. This problem may also occur in cluster enumeration approaches that attempt to find densely connected groups of embeddings in the projection space, which necessitates the application of a robust embedding technique. In the following section, the example of robust graph-based cluster enumeration is discussed.

1) CLUSTER ENUMERATION

Assume that for each candidate number of clusters $K_{\text{cand}} \in \{K_{\min}, \ldots, K_{\max}\}$ there is a clustering algorithm, e.g. [27], [28], that partitions $\hat{\mathbf{y}}^{(\hat{\gamma})}$ into K_{cand} number of clusters and provides an estimated label vector $\hat{\mathbf{c}}_{K_{\text{cand}}}$. After estimating label vectors for each candidate number of clusters K_{cand}, the cluster number K can be estimated by comparing quality of partitions using modularity as [67]

$$\hat{K} = \underset{\hat{K}_{\min},\ldots,\hat{K}_{\max}}{\arg\max} \left\{ Q_{\hat{K}_{\text{cand}}} \right\}, \tag{16}$$

where

$$Q_{\hat{K}_{\text{cand}}} = \frac{1}{2g} \sum_{i,j}^{N} \left[w_{i,j} - \frac{d_i d_j}{2g} \right] \delta\left(\hat{c}_i, \hat{c}_j\right) \tag{17}$$

denotes the modularity score for a candidate number of clusters $\hat{K}_{\text{cand}}$, $w_{i,j}$ is the edge weight between the ith and the jth data-point of $\mathbf{X}$, d_i is the weighted node degree attached to vertex i, $\hat{c}_i$ is the estimated community label associated to vertex i, $g = \frac{1}{2}\sum_{i,j} w_{i,j}$, and the function $\delta(\hat{c}_i, \hat{c}_j)$ equals 1 if $\hat{c}_i = \hat{c}_j$ and is zero, otherwise.

V. EXPERIMENTAL EVALUATION

This section contains the numerical experimental evaluation of the proposed RRLPI method on a broad range of simulated and real-world data sets with applications to robust cluster enumeration and image segmentation. In the following, a detailed information about experimental setting is provided. A MATLAB implementation of RRLPI is available at: https://github/A-Tastan/RRLPI

Benchmark Methods: The effects of Type I and Type II outliers on the Fiedler vector computation are studied for the LE [39], LPI [15], RLPI [17], RLPFM [32] and RRLPI embedding-based approaches by designing synthetic data Monte Carlo experiments. Then, in addition to the above mentioned embedding approaches, the proposed RRLPI is benchmarked against three state-of-the-art graph-based cluster enumeration approaches, i.e., Le Martelot [68], Combo [69] and Sparcode [70] and two state-of-the art spectral partitioning approaches, i.e., FastEFM [30] and LSC [31] in terms of image segmentation capabilities.

Parameter Settings: Some of the competitors are parameter free approaches, i.e. LE [39], LPI [15], Martelot [68], Combo [69], Sparcode [70] and LSC [31]. For the FastEFM approach [30], the Gaussian scale parameter σ is the mean distance among all data-points as the authors suggested. In terms of accuracy, the authors suggested to increase the desired dimension of explicit features. Therefore, the desired dimension is $D = 500$ as suggested by the authors. Further, to analyze the performance of proposed penalty parameter selection and to provide fair comparisons, the RLPI [17] and RLPFM [32] approaches are all run using the proposed penalty parameter selection algorithm. The remaining parameters of

RLPI, RLPFM and RRLPI are defined using the default setting: $\gamma_{\min} = 10^{-8}$, $\gamma_{\max} = 1000$, $K_{\min} = 1$, $K_{\max} = 10$ and $N_{\min} = \frac{N}{K_{\max}}$.

Affinity Matrix Construction: To analyze the robustness of RRLPI, cosine similarity is used as the affinity matrix construction method in all experiments, unless otherwise specified.

Performance Measures: The average partition accuracy $\bar{p}_{\text{acc}}$ is measured by evaluating

$$\bar{p}_{\text{acc}} = \frac{1}{NN_E} \sum_{i=1}^{N_E} \sum_{j=1}^{N} \mathbb{1}_{\{\hat{c}_j = c_j\}}, \tag{18}$$

where

$$\mathbb{1}_{\{\hat{c}_j = c_j\}} = \begin{cases} 1, & \text{if } \hat{c}_j = c_j \\ 0, & \text{otherwise} \end{cases}, \tag{19}$$

N is the number of observations, N_E is the total number of experiments, and $\hat{c}_j$ and c_j are the estimated and ground truth labels for the jth observation, respectively.

The empirical probability of detection p_{det} is used to assess cluster enumeration performance as follows

$$p_{\text{det}} = \frac{1}{N_E} \sum_{i=1}^{N_E} \mathbb{1}_{\{\hat{K} = K\}}, \tag{20}$$

where $\hat{K}$ denotes the estimated number of clusters and $\mathbb{1}_{\{\hat{K}=K\}}$ is the indicator function.

The contour matching score F_{score} for boundaries and the Jaccard index J are used for the numerical performance analysis in the case of image segmentation [71]. The F_{score} quantifies whether a boundary has a match on the ground truth boundary as follows

$$F_{\text{score}} = 2\frac{P \cdot R}{R + P}, \tag{21}$$

where P and R denote precision and recall values, respectively. The Jaccard index evaluates similarity between estimated and ground truth segmentations according to

$$J\left(\hat{\boldsymbol{I}}_{\text{seg}}, \boldsymbol{I}_{\text{seg}}\right) = \frac{\text{TP}}{\text{TP} + \text{FP} + \text{FN}}, \tag{22}$$

where $\hat{\boldsymbol{I}}_{\text{seg}}$ and $\boldsymbol{I}_{\text{seg}}$ denote estimated and ground truth segmentations for image $\boldsymbol{I}$ and TP, FP, and FN are true positives, false positives and false negatives, respectively.

A. OUTLIER EFFECTS AND ROBUSTNESS

To visualize outlier effects on the eigen-decomposition, a synthetic data set is generated for $K = 3$ easily separable clusters, see Fig. 9. The M-dimensional data-points of each cluster c_k, with $k = 1, \ldots, K$, and $M = 6$ are generated as $\mathbf{x}_{i,k} = \boldsymbol{\mu}_{c_k} + \vartheta_{c_k} \boldsymbol{\upsilon}$, where $\mathbf{x}_{i,k}$ is the ith data-point associated with the kth cluster, $\boldsymbol{\mu}_{c_k}$ is the kth cluster centroid, ϑ_{c_k} is the kth scaling constant, and $\boldsymbol{\upsilon}$ is a vector of independently and identically distributed random variables from a uniform distribution on the interval $[-0.5, 0.5]$. All details and parameter values to generate the data are provided in Appendix D.1.1 of the supplementary material.

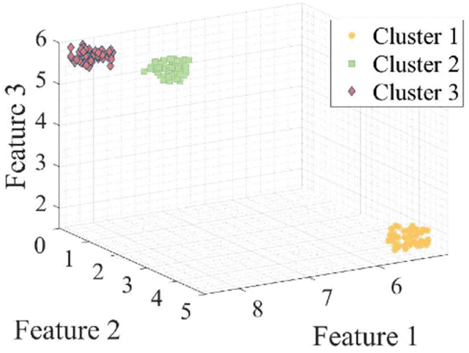
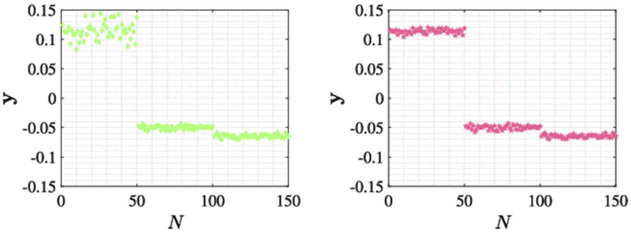


FIGURE 9. Examplary plot of the first three features of the uncorrupted synthetic data set.



(a) LE **(b) RRLPI**

FIGURE 10. Computed eigenvectors for the uncorrupted data set.

Representative examples of the computed eigenvectors are shown for LE and for the proposed RRLPI in Fig. 10(a) and (b), respectively. In the absence of outliers, both algorithms provide embeddings where the embedding points that are associated with the same cluster are concentrated, and the different clusters are separated. To study robustness, the data set is contaminated with two outlier types, i.e., outliers that do not correlate with any cluster (Type I) and outliers correlate with more than one cluster (Type II); see Section III for a definition and a discussion. An example showing the first three features of the contaminated data set is shown in Fig. 11, where both Type I and Type II outliers are highlighted as red crosses.

The Type I and Type II outliers are, respectively, generated as $\tilde{\mathbf{x}}_i^{(1)} = \mathbf{x}_{i,k} + \vartheta_1 \boldsymbol{\upsilon}$ and $\tilde{\mathbf{x}}_i^{(2)} = \boldsymbol{\mu}_2 + \vartheta_2 \boldsymbol{\upsilon}$ where $\tilde{\mathbf{x}}_i^{(j)}$, $j = 1, 2$ denotes the type of the outlier, ϑ_j, $j = 1, 2$ is a scaling constant associated with the outlier type and $\boldsymbol{\mu}_2$ is a vector associated with the location of Type II outliers. A detailed explanation including all parameter values, is provided in the supplementary material Appendix D.1.1. Examples of the eigenvector computations based on the corrupted data set are shown for LE and RRLPI in Fig. 12(a) and (b), respectively. As can be seen, for the LE method, Type I outliers in the data produce outliers in the embedding results that obscure the underlying structure of $K = 3$ clusters. In contrast, the proposed

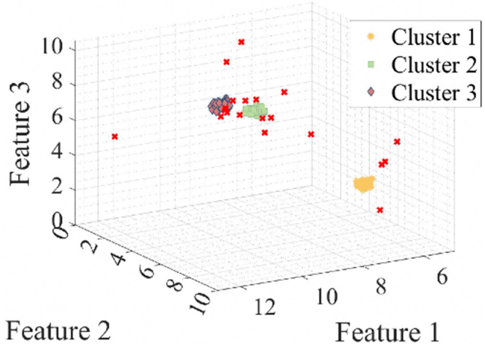


FIGURE 11. Examplary plot of the first three features of the synthetic data set after corruption with Type I and Type II outliers (red crosses). See Section III, for a discussion.

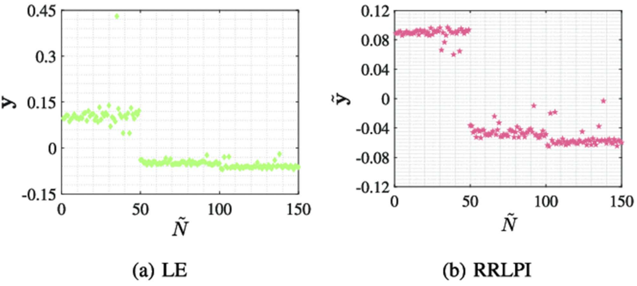


FIGURE 12. Computed eigenvectors for the corrupted data set.

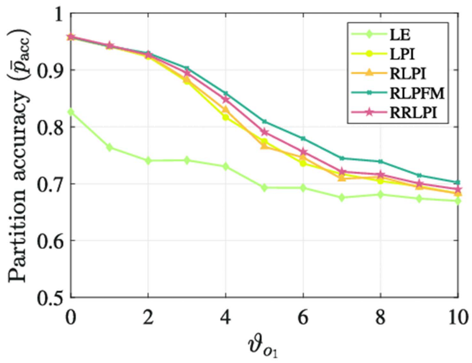


FIGURE 13. $\bar{p}_{\text{acc}}$ for increasing ϑ_1 associated with Type I outliers ($N = 300, N_{\text{out}} = 10, \vartheta_2 = 1.5, \vartheta_{c_K} = 0.5$ s.t. $k = 1, \ldots, K$).

RRLPI provides an embedding that is less influenced by the outliers.

Figs. 13 and 14 report the average partition accuracy as a function of the constant ϑ_1 associated with Type I outliers and the number of outliers N_{out} for each outlier type, respectively. The value of ϑ_2 is kept constant to generate points that lie between clusters two and three. The robust methods show best performance while the performance of LE quickly decreases in the presence of outliers.

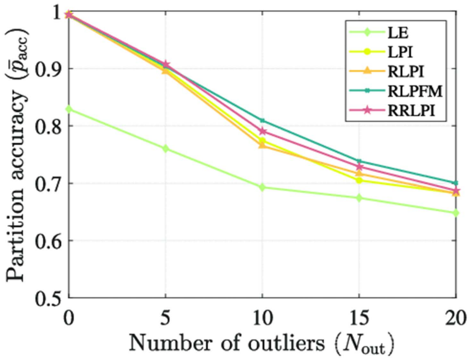


FIGURE 14. $\bar{p}_{\text{acc}}$ for each of outlier type with increasing N_{out} ($N = 300, \vartheta_1 = 5, \vartheta_2 = 1.5, \vartheta_{c_K} = 0.5$ s.t. $k = 1, \ldots, K$).

B. COMPUTATION TIME

The computation time (t) is reported as a function of increasing number of data-points in the synthetic data set of $K = 3$ clusters. The experiments are performed based on three different scenarios. First, computation time is analyzed for an uncorrupted data set that has been explained in the previous section. Then, the data set is contaminated with two outlier types (Type I and II) where 10% of the data set are outliers for every type. The graphical models of these two data sets are generated based on the cosine similarity measure. Finally, to analyze the effect of sparsity, a sparse graph model of an uncorrupted data set is computed by using nearest neighbor graphs where the number of neighbors is set according to the cluster sizes. In all experiments, the penalty parameter is set to one and t is averaged over 100 Monte Carlo runs.

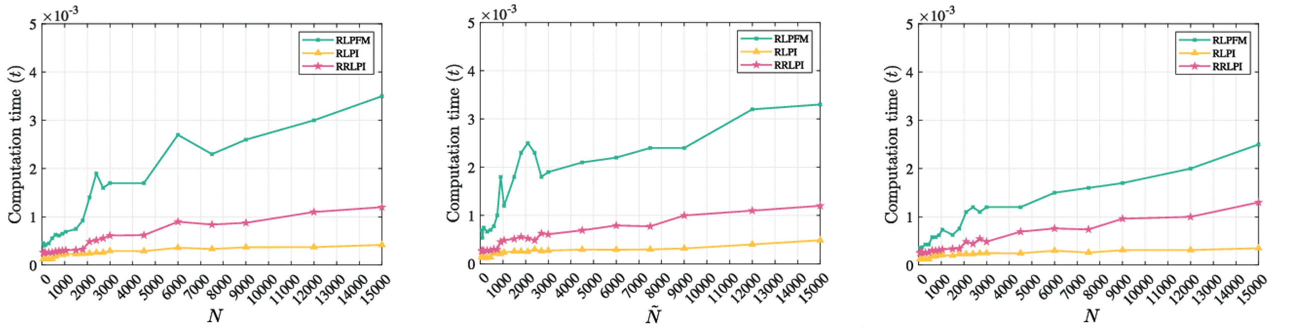
The performance of RRLPI is benchmarked against its main competitors RLPI and RLPFM in Fig. 15. Even though robustness results in an increased computation cost, the single-step weighting procedure of RRLPI is considerably more efficient than the iterative weighting in RLPFM. The LPI method [15] is excluded in the computation time analysis due to its different operational procedure. However, our own run-time analysis confirmed the theoretical analysis that RRLPI has quadratic complexity with respect to N. This means that RRLPI is computationally more efficient than LPI which has cubic complexity in unsupervised settings as it has been stated in [17].

C. CLUSTER ENUMERATION

In this section, the cluster enumeration performance of different approaches is benchmarked in terms of their empirical probability of detection using the following data sets:

Human Gait Data Set: The experimental data set [52] was collected in an office environment at Technische Universität Darmstadt using a 24 GHz radar system [72]. The data set consists of 800 measurements from five different gait types measured from two different directions [73].

Breast Cancer Wisconsin Data Set: The data set includes 569 observations from two classes that define benign or malignant tumors [74].



(a) t for the uncorrupted data set

(b) t for the corrupted data set

(c) t for the sparse graph model of uncorrupted data set

FIGURE 15. Computation time performance analysis. The results are reported in seconds. (For step-by-step detailed computational analysis, see Section IV-D.)

TABLE 1. Performance of Different Cluster Enumeration Approaches on Well-Known Clustering Data Sets

	$\hat{K}$ for Different Cluster Enumeration Methods									
Data Set	Martelot	Combo	Sparcode	RLPFM	LE	LPI	RLPI	RRLPI	K	Similarity
Human Gait [52],	4	6	5	4	4	4	4	5	5	enet
Breast Cancer Wisconsin [74],	1	2	2	2	4	2	2	2	2	cos
Iris [75],	2	3	2	3	5	3	3	3	3	enet
Person Identification [51],	6	7	4	5	10	4	4	4	4	enet
Sonar [76],	2	2	2	2	6	2	2	2	2	cos
Ionosphere [77],	3	3	4	2	7	3	2	2	2	cos
D. Retinopathy [78],	2	2	2	2	2	2	2	2	2	cos
Gesture Phase S. [79],	2	3	3	5	10	2	6	5	5	cos

The results summarized for similarity measures cosine (cos) and elastic net (enet) using a penalty parameter of $\rho = 0.5$.

Iris Data Set: The iris data set consists of 150 measurements of three different iris flower species [75].

Person Identification Data Set: The experimental data set [51] was collected using the same settings as in the human gait data set. The data set includes radar observations of four different subjects walking towards and away from the radar system.

Connectionist Bench Data Set (Sonar): The data set includes sonar returns collected from a metal cylinder and a cylindric rock positioned on a sandy ocean floor [76]. The number of observations is equal to 208 for two object clusters.

Ionosphere Data Set: The data set includes 351 radar returns from the ionosphere for two clusters [77].

Diabetic Retinopathy Debrecen Data Set (D. Retinopathy): The data set consists of 1151 observations of two clusters using image-based features of diabetic retinopathy [78].

Gesture Phase Segmentation Data Set (Gesture Phase S.): The processed features that contain scalar velocity and acceleration of hand and wrist movements have been used for videos A1, A2, A3, C1 and C3. The data set includes five phases which are rest, preparation, stroke, hold and retraction [79].

If none of the cluster enumeration approaches estimates the cluster number correctly with the default cosine similarity, the elastic net similarity measure as in [54], is used with ten candidate penalty parameters ρ on an equidistant grid ranging from 0.1 to one. Results are reported for $\rho = 0.5$, which gave the best average overall detection performance for all methods. Tukey's distance function [54] where the threshold defined as $c_{\text{Tukey}} = 3$ is used as an initialization for K-medoids partitioning in the proposed algorithm. For a detailed discussion about different similarity measures and partitioning results, see Appendix D.2.1 of the supplementary material. The estimated cluster numbers are reported for the different cluster enumeration approaches in Table 1. As can be seen from the table, the human gait and gesture phase data sets include a considerable number of outliers that result in misdetection of the cluster number for almost all competitors. The proposed method is the only one that consistently estimates the correct cluster numbers for all data sets.

The empirical probability of detection with respect to different penalty parameters is detailed in Fig. 16. Then, the performance is summarized in Fig. 17 by averaging the results over all penalty parameters. The results for cluster enumeration demonstrate that the proposed RRLPI shows the best probability of detection performance for all candidate penalty parameters with an average score of 79 %, whereas the best competitors (RLPFM and RLPI) have scores of 73 % and 63 %, respectively.

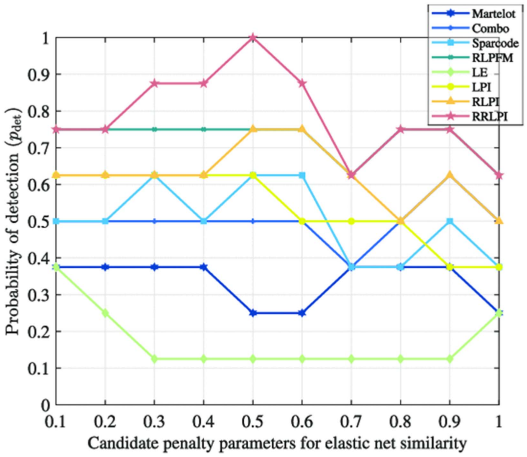


FIGURE 16. $\bar{p}_{\rm det}$ **with respect to different penalty parameters for** K**-medoids partitioning with Tukey's distance function [54] for the initialization (**$c_{\rm Tukey} = 3$**).**

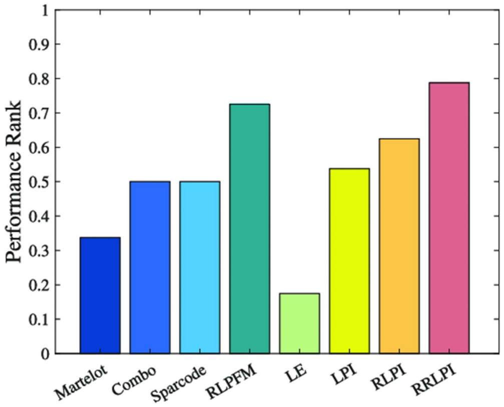


FIGURE 17. Performance rank for K**-medoids partitioning with Tukey's distance function for the initialization (**$c_{\rm Tukey} = 3$**).**

The sensitivity of the cluster enumeration results to different partitioning algorithms is summarized in Fig. 18. The results demonstrate that the cluster enumeration performance of the RLPFM approach is very sensitive to the determined partitioning algorithm while proposed RRLPI method performs best among all partitioning methods.

D. IMAGE SEGMENTATION

ADE20K [80], is a large-scale dataset that includes high quality pixel-level annotations of 25210 images (20210, 2000, and 3000 for the training, validation, and test sets, respectively.). In our experiments, 10 images from the ADE20K data set containing different objects, where each object has a different color, have been selected for color-based image segmentation. The selected and corresponding annotated images are denoted as $\boldsymbol{I}$ and $\boldsymbol{I}_{\rm seg}$, respectively. The images are down-sampled, where the dimension of data set $\mathbf{X}$ is $M = 3$ and $N \cong 15000$ using RGB color codes associated to down-sampled image pixels as features. To analyze robustness, the images are corrupted by adding multiplicative noise using the equation $\tilde{\boldsymbol{I}} = \boldsymbol{I} + \xi \times \boldsymbol{I}$, where $\tilde{\boldsymbol{I}}$ denotes the corrupted image and ξ is uniformly distributed random noise with zero mean and variance $\sigma^{(\xi)}$.

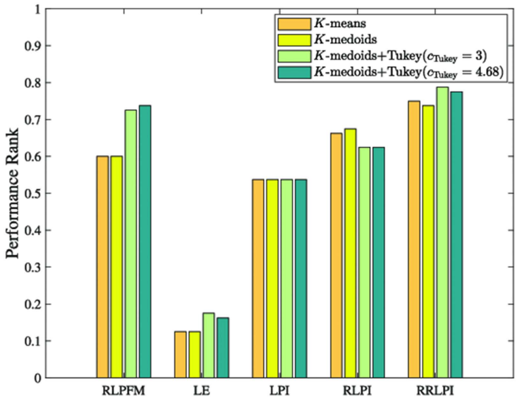


FIGURE 18. Overall performance rank for different partitioning algorithms.

The down-sampled images are segmented for a pre-defined number of segments K using the default setting which performs K-means partitioning for the data sets that have more than $N = 3000$ samples. In Fig. 19, examples of the original image $\boldsymbol{I}$ and associated segmented images using the computed Fiedler vectors for seven different embedding approaches are shown along with the ground truth segmented image $\boldsymbol{I}_{\rm seg}$. The uncorrupted images $\boldsymbol{I}$ may also contain outlying pixels and/or noisy pixels. The effect of outliers is that a small number of pixels are mapped far away from the group of pixels and, thus, the remaining group of pixels assigned to a single large segment based on the distance-based partitioning methods.

A typical example of a segmentation result illustrating the outlier effects is provided in Fig. 20(a). As can be seen, the described outlier effect is observed even for the embeddings of the uncorrupted (original) image when using LE. To exemplify the robust Fiedler vector estimation, the segmentation result of RRLPI is shown in Fig. 20(b). The segmentation result demonstrates that the proposed robust Fiedler vector estimation suppress outlier effects on the eigen-decomposition and provides segmentation results that are more consistent with the annotated image $\boldsymbol{I}_{\rm seg}$. Further, in Fig. 21, examples of segmented images are presented for the corrupted images where $\sigma^{(\xi)} = 10^{-3}$. The results show that the outlier effect on eigen-decomposition causes a breakdown of the FastEFM, LSC, and LE approaches. For further examples and detailed numerical results, see supplementary material Appendix D.3.2.

The experiments are evaluated quantitatively using $\bar{F}_{\rm score}$, $\bar{J}$ and $\bar{p}_{\rm acc}$, and the results are summarized in Fig. 22. All performance measures are evaluated by comparing each estimated segmented image $\hat{\boldsymbol{I}}_{\rm seg}$ with the annotated image $\boldsymbol{I}_{\rm seg}$. The LE and FastEFM show poor performance, even for the original images. Although LSC shows a reasonably good performance for the original images, its performance reduces drastically in the outlier-corrupted case in terms of $\bar{F}_{\rm score}$ and $\bar{J}$. The LPI,

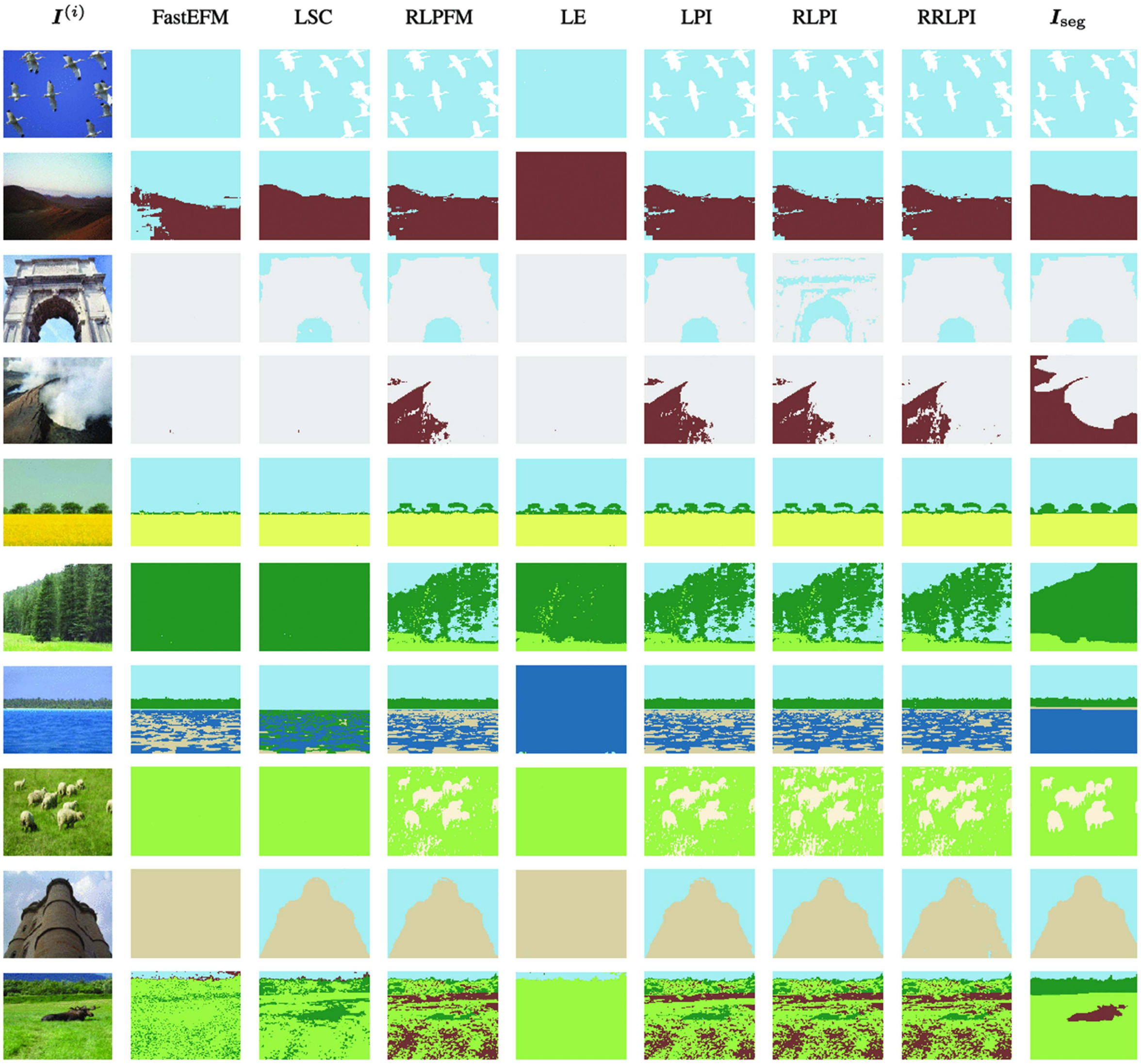


FIGURE 19. Image segmentation results for the original images.

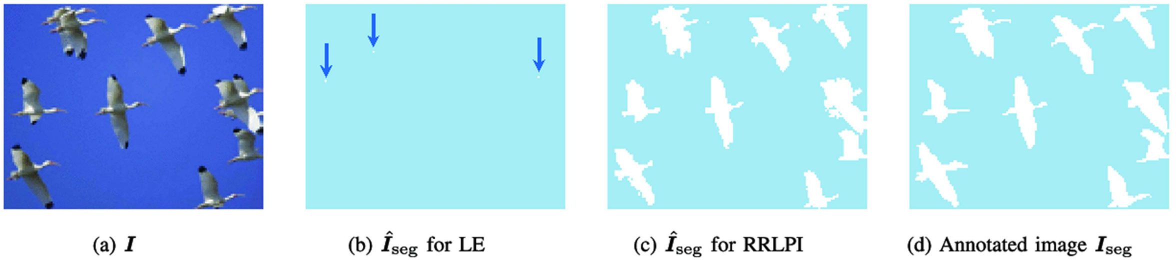

(a) I
(b) $\hat{I}_{\text{seg}}$ for LE
(c) $\hat{I}_{\text{seg}}$ for RRLPI
(d) Annotated image I_{seg}

FIGURE 20. Example segmentations for LE and RRLPI methods. The embeddings that are mapped far away from the group of pixels are pointed out using arrows.

RLPFM and RRLPI are the top three methods in all performance measures and RLPI follows them with a reasonably good performance, which indicates that the proposed penalty parameter selection algorithm is a promising approach, even when using non-robust methods.

In summary, the proposed RRLPI shows considerably better performance than all methods in cluster enumeration and provides more stable results than its main competitor, i.e., the RLPFM method in different partitioning settings. The image segmentation results showed that LPI, RLPFM and the

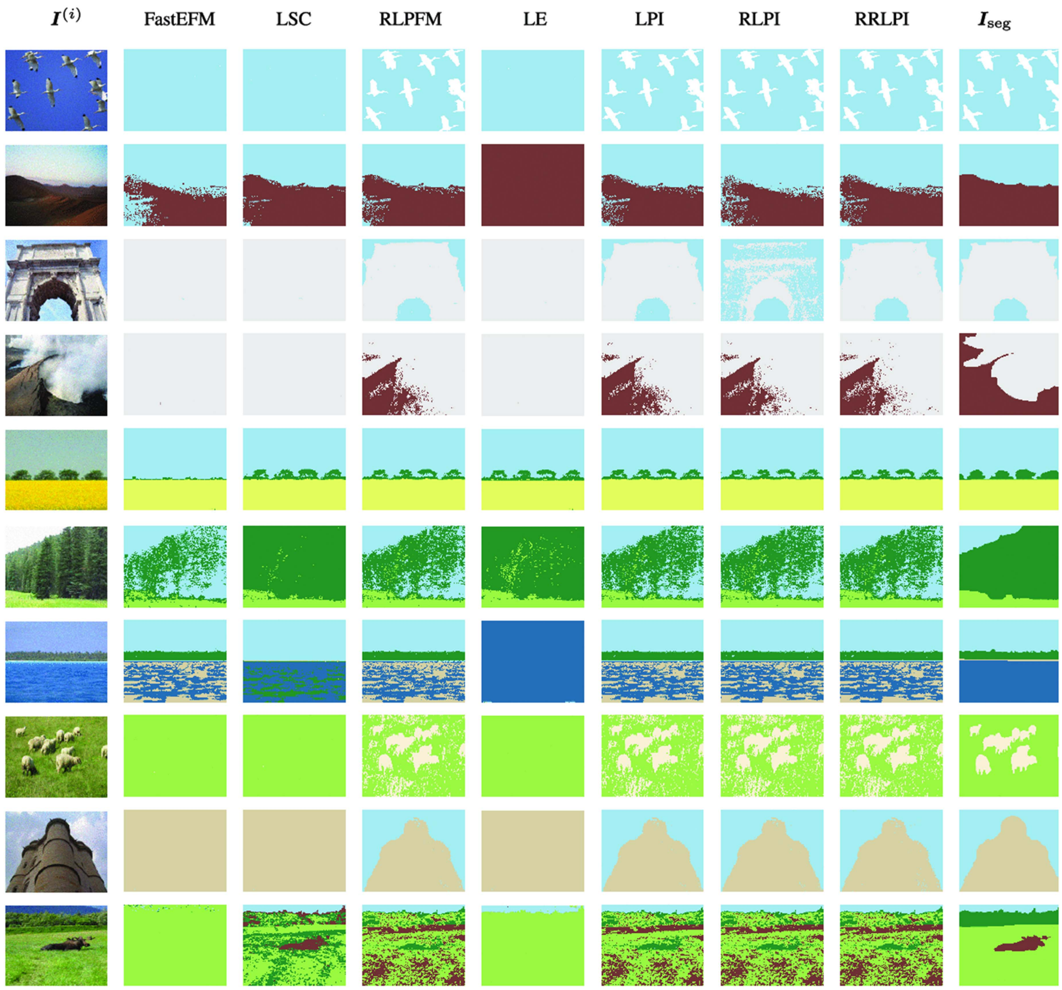
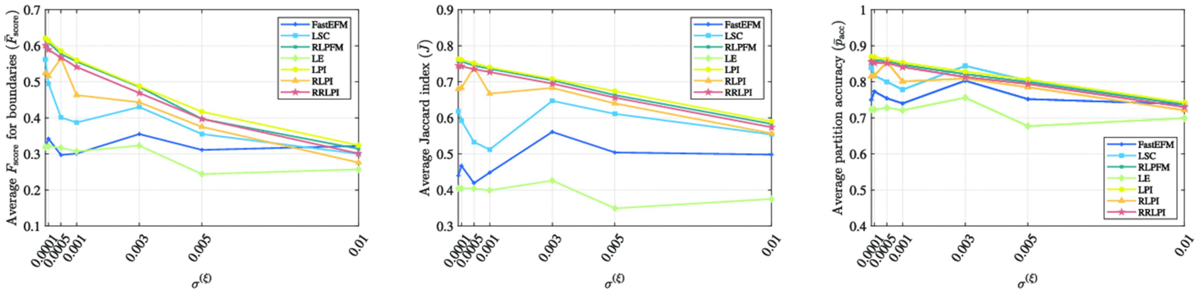


FIGURE 21. Image segmentation results for the corrupted images. ($\sigma^{(\xi)} = 10^{-3}$).



(a) $\bar{F}_{\text{score}}$ for increasing $\sigma^{(\xi)}$

(b) $\bar{J}$ for increasing $\sigma^{(\xi)}$

(c) $\bar{p}_{\text{acc}}$ for increasing $\sigma^{(\xi)}$

FIGURE 22. Numerical results for the image segmentation.

proposed RRLPI method perform similarly while its computational efficiency provides an advantage compared to its competitors.

VI. CONCLUSION

The effect of outliers on eigen-decomposition has been analyzed and the importance of weighted node degree attached to a vertex has been shown to be a useful measure of outlyingness. Based on the derived theoretical results, we proposed RRLPI, a method to robustly estimate the Fiedler vector that down-weights embeddings, for which the weighted node degree deviates from the typical weighted node degree of a given graph. The objective function to estimate the Fiedler vector is penalized using the proposed unsupervised penalty parameter selection algorithm that builds upon Δ-separated sets. The performance of RRLPI is benchmarked for different applications on a variety of real-world data sets. The numerical results for cluster analysis and image segmentation showed that the RRLPI is a promising approach for Fiedler vector estimation in situations where robustly determining the group structure in a data set is essential.

In future work, we plan to extend our knowledge about the location of outliers in the embedding space by analyzing the eigenvectors associated with the K smallest eigenvalues of the Laplacian matrix which are used in the well-known spectral clustering method. The obtained information about the location of outliers can be useful to reduce the computational cost of outlier detection algorithms that are evaluating the possibility of being outlier for every data point. The findings can also be integrated to the design of robust spectral clustering approaches. Moreover, the assumptions on the affinity matrix may in future be relaxed by considering random graphs. Finally, in future work, to further generalize the methodology other metrics for outlier identification [81] may be investigated.

ACKNOWLEDGMENT

The authors would like to thank Ayfer Taştan for checking the eigen-decompositions.

REFERENCES

[1] A. Bertrand and M. Moonen, “Seeing the bigger picture: How nodes can learn their place within a complex ad hoc network topology,” *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 71–82, May 2013.

[2] M. Fiedler, “Algebraic connectivity of graphs,” *Czechoslovak Math. J.*, vol. 23, pp. 298–305, 1973.

[3] R. Aragues, G. Shi, D. V. Dimarogonas, C. Sagues, and K. H. Johansson, “Distributed algebraic connectivity estimation for adaptive event-triggered consensus,” in *Proc. IEEE Amer. Control Conf.*, 2012, pp. 32–37.

[4] E. Tam and D. Dunson, “Fiedler regularization: Learning neural networks with graph sparsity,” in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 9346–9355.

[5] P. Yang, R. A. Freeman, G. J. Gordon, K. M. Lynch, S.S. Srinivasa, and R. Sukthankar, “Decentralized estimation and control of graph connectivity for mobile sensor networks,” *Automatica*, vol. 46, pp. 390–396, 2010.

[6] P. Di Lorenzo and S. Barbarossa, “Distributed estimation and control of algebraic connectivity over random graphs,” *IEEE Trans. Signal Process.*, vol. 62, no. 21, pp. 5615–5628, Nov. 2014.

[7] D. A. Spielman and S.-H. Teng, “Spectral partitioning works: Planar graphs and finite element meshes,” in *Proc. IEEE 37th Conf. Found. Comput. Sci.*, 1996, pp. 96–105.

[8] T. Sahai, A. Speranzon, and A. Banaszuk, “Hearing the clusters of a graph: A distributed algorithm,” *Automatica*, vol. 48, pp. 15–24, 2012.

[9] S. E. Schaeffer, “Graph clustering,” *Comput. Sci. Rev.*, vol. 1, pp. 27–64, 2007.

[10] P. Orponen and S. E. Schaeffer, “Local clustering of large graphs by approximate Fiedler vectors,” in *Proc. Int. Workshop Exp. Efficient Algorithms*, 2005, pp. 524–533.

[11] B. Hendrickson, “Latent semantic analysis and Fiedler retrieval,” *Linear Algebra Appl.*, vol. 421, pp. 345–355, 2007.

[12] A. DePavia and S. Steinerberger, “Spectral clustering revisited: Information hidden in the Fiedler vector,” 2020. [Online]. Available: https://arxiv.org/abs/2003.09969

[13] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, “Indexing by latent semantic analysis,” *J. Amer. Soc. Inf. Sci.*, vol. 41, pp. 391–407, 1990.

[14] D. Cai, X. He, and J. Han, “Document clustering using locality preserving indexing,” *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 12, pp. 1624–1637, Dec. 2005.

[15] X. He, D. Cai, H. Liu, and W.-Y. Ma, “Locality preserving indexing for document representation,” in *Proc. 27th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2004, pp. 96–103.

[16] X. He and P. Niyogi, “Locality preserving projections,” in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, 2004, vol. 16, pp. 153–160.

[17] D. Cai, X. He, W. V. Zhang, and J. Han, “Regularized locality preserving indexing via spectral regression,” in *Proc. 16th ACM Conf. Inf. Knowl. Manage.*, 2007, pp. 741–750.

[18] A. Gang and W. U. Bajwa, “A linearly convergent algorithm for distributed principal component analysis,” *Signal Process.*, vol. 193, 2022, Art. no. 108408.

[19] X. Chang, F. Nie, S. Wang, Y. Yang, X. Zhou, and C. Zhang, “Compound rank-k projections for bilinear analysis,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 7, pp. 1502–1513, Jul. 2016.

[20] C. Yan et al., “Self-weighted robust LDA for multiclass classification with edge classes,” *ACM Trans. Intell. Syst. Technol.*, vol. 12, pp. 1–19, 2020.

[21] R. Zhou, X. Chang, L. Shi, Y.-D. Shen, Y. Yang, and F. Nie, “Person reidentification via multi-feature fusion with adaptive graph learning,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 5, pp. 1592–1601, May 2020.

[22] W. Yu, X. Teng, and C. Liu, “Face recognition using discriminant locality preserving projections,” *Image Vis. Comput.*, vol. 24, pp. 239–248, 2006.

[23] J. Lu and Y.-P. Tan, “Regularized locality preserving projections and its extensions for face recognition,” *IEEE Trans. Syst. Man Cybern. Part B. Cybern.*, vol. 40, no. 3, pp. 958–963, Jun. 2010.

[24] Y. Zhu, C. Zhu, and X. Li, “Improved principal component analysis and linear regression classification for face recognition,” *Signal Process.*, vol. 145, pp. 175–182, 2018.

[25] M. Artac, M. Jogan, and A. Leonardis, “Incremental PCA for on-line visual learning and recognition,” in *Proc. IEEE Int. Conf. Pattern Recognit.*, 2002, vol. 3, pp. 781–784.

[26] A. Ortega, P. Frossard, J. Kovačević, J. M. Moura, and P. Vandergheynst, “Graph signal processing: Overview, challenges, and applications,” *Proc. IEEE*, vol. 106, no. 5, pp. 808–828, May 2018.

[27] A. Y. Ng, M. I. Jordan, and Y. Weiss, “On spectral clustering: Analysis and an algorithm,” in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, 2001, vol. 14, pp. 849–856.

[28] T. Xiang and S. Gong, “Spectral clustering with eigenvector selection,” *Pattern Recognit.*, vol. 41, pp. 1012–1029, 2008.

[29] R. Goldenberg, R. Kimmel, E. Rivlin, and M. Rudzsky, “Behavior classification by eigendecomposition of periodic motions,” *Pattern Recognit.*, vol. 38, pp. 1033–1043, 2005.

[30] L. He, N. Ray, Y. Guan, and H. Zhang, “Fast large-scale spectral clustering via explicit feature mapping,” *IEEE Trans. Cybern.*, vol. 49, no. 3, pp. 1058–1071, Mar. 2019.

[31] D. Cai and X. Chen, “Large scale spectral clustering with landmark-based sparse representation,” *IEEE Trans. Cybern.*, vol. 45, no. 8, pp. 1669–1680, Aug. 2015.

[32] A. Taştan, M. Muma, and A. M. Zoubir, “Robust spectral clustering: A locality preserving feature mapping based on M-estimation,” in *Proc. IEEE 29th Eur. Signal Process. Conf.*, 2021, pp. 851–855.

[33] X. Cheng and G. Mishne, “Spectral embedding norm: Looking deep into the spectrum of the graph Laplacian,” *SIAM J. Imag. Sci.*, vol. 13, pp. 1015–1048, 2020.

[34] U. Shaham, K. Stanton, H. Li, B. Nadler, R. Basri, and Y. Kluger, “SpectralNet: Spectral clustering using deep neural networks,” 2018. [Online]. Available: https://arxiv.org/abs/1801.01587

[35] L. Zelnik-Manor and P. Perona, “Self-tuning spectral clustering,” in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, vol. 17, 2004.

[36] A. Cloninger and W. Czaja, “Eigenvector localization on data-dependent graphs,” in *Proc. IEEE Int. Conf. Sampling Theory Appl.*, 2015, pp. 608–612.

[37] B. Nadler and M. Galun, “Fundamental limitations of spectral clustering,” in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, vol. 19, 2006, pp. 1017–1024.

[38] S. Shahbazpanahi, S. Valaee, and M. H. Bastani, “Distributed source localization using ESPRIT algorithm,” *IEEE Trans. Signal Process.*, vol. 49, no. 10, pp. 2169–2178, Oct. 2001.

[39] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, 2001, vol. 14, pp. 585–591.

[40] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Comput.*, vol. 15, no. 6, pp. 1373–1396, Jun. 2003.

[41] Z. Li, F. Nie, X. Chang, L. Nie, H. Zhang, and Y. Yang, "Rank-constrained spectral clustering with flexible embedding," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 12, pp. 6073–6082, Dec. 2018.

[42] J. Feng, Z. Lin, H. Xu, and S. Yan, "Robust subspace segmentation with block diagonal prior," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 3818–3825.

[43] M. Abdolali, N. Gillis, and M. Rahmati, "Scalable and robust sparse subspace clustering using randomized clustering and multilayer graphs," *Signal Process.*, vol. 163, pp. 166–180, 2019.

[44] X. Zhu, S. Zhang, Y. Li, J. Zhang, L. Yang, and Y. Fang, "Low-rank sparse subspace for spectral clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 8, pp. 1532–1543, Aug. 2019.

[45] X. Peng, Z. Yi, and H. Tang, "Robust subspace clustering via thresholding ridge regression," in *Proc. AAAI Conf. Artif. Intell.*, 2015,vol. 29, pp. 3827–3833.

[46] S. Arora, S. Rao, and U. Varizani, "Expander flows, geometric embeddings and graph partitioning," *J. ACM*, vol. 56, pp. 1–37, 2009.

[47] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2765–2781, Nov. 2013.

[48] U. Von Luxburg, "A tutorial on spectral clustering," *Statist. Comput.*, vol. 17, pp. 395–416, 2007.

[49] M. Meilă and J. Shi, "A random walks view of spectral segmentation," in *Proc. Int. Workshop Artif. Intell. Stat.*, 2001, pp. 203–208.

[50] D. Verma and M. Meilă, "A comparison of spectral clustering algorithms," Univ. Washington, Seattle, WA, USA, Tech. Rep. UWCSE030501, vol. 1, pp. 1–18, 2003.

[51] F. K. Teklehaymanot, A. -K. Seifert, M. Muma, M. G. Amin, and A. M. Zoubir, "Bayesian target enumeration and labeling using radar data of human gait," in *Proc. IEEE 26th Eur. Signal Process. Conf.*, 2018, pp. 1342–1346.

[52] A.-K. Seifert, M. Amin, and A. M. Zoubir, "Toward unobtrusive in-home gait analysis based on radar micro-Doppler signatures," *IEEE Trans. Biomed. Eng.*, vol. 66, no. 9, pp. 2629–2640, Sep. 2019.

[53] S. A. Razavi, E. Ollila, and V. Koivunen, "Robust greedy algorithms for compressed sensing," in *Proc. IEEE 20th Eur. Signal Process. Conf.*, 2012, pp. 969–973.

[54] A. M. Zoubir, V. Koivunen, E. Ollila, and M. Muma, *Robust Statistics for Signal Processing*. Cambridge, U.K.: Cambridge Univ. Press, 2018.

[55] P. J. Huber and E. M. Ronchetti, *Robust Statistics*. Hoboken, NJ, USA: Wiley, 2009.

[56] T. Leighton and S. Rao, "Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms," *J. ACM*, vol. 46, pp. 787–832, 1999.

[57] W. Fan et al., "Application driven graph partitioning," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2020, pp. 1765–1779.

[58] A. Nazi, W. Hang, A. Goldie, S. Ravi, and A. Mirhoseini, "GAP: Generalizable approximate graph partitioning framework," 2019. [Online]. Available: https://arxiv.org/abs/1903.00614

[59] D. A. Spielman and S.-H. Teng, "Spectral partitioning works: Planar graphs and finite element meshes," *Linear Algebra Appl.*, vol. 421, pp. 284–305, 2007.

[60] G. W. Stewart, *Matrix Algorithms: Volume I Basic Decompositions*. Philadelphia, PA, USA: SIAM, 1998.

[61] G. W. Stewart, "A Krylov–Schur algorithm for large eigenproblems," *SIAM J. Matrix Anal. Appl.*, vol. 23, pp. 601–614, 2002.

[62] G. W. Stewart, *Matrix Algorithms: Volume II Eigensystems*. Philadelphia, PA, USA: SIAM, 2001.

[63] P. J. Rousseeuw and G. W. Bassett, "The remedian: A robust averaging method for large data sets," *J. Amer. Stat. Assoc.*, vol 85, pp. 97–104, 1990.

[64] P. Courrieu, "Fast computation of Moore-Penrose inverse matrices," 2008. [Online]. Available: https://arxiv.org/abs/0804.4809

[65] C. C. Paige and M. A. Saunders, "Algorithm 583 LSQR: Sparse linear equations and least squares problems," *ACM Trans. Math. Softw.*, vol. 8, pp. 195–209, 1982.

[66] Y. Han, "Sorting real numbers in $O(n\sqrt{\log n})$ time and linear space," *Algorithmica*, vol. 82, pp. 966–978, 2020.

[67] A. Clauset, M. E. J. Newman, and C. Moore, "Finding community structure in very large networks," *Phys. Rev. E*, vol. 70, 2004, Art. no. 066111.

[68] E. L. Martelot and C. Hankin, "Multi-scale community detection using stability as optimization criterion in a greedy algorithm," in *Proc. Int. Conf. Knowl. Discov. Inf. Retrieval*, 2011, pp. 208–217.

[69] S. Sobolevsky, R. Campari, A. Belyi, and C. Ratti, "General optimization technique for high-quality community detection in complex networks," *Phys. Rev. E*, vol. 90, 2014, Art. no. 012811.

[70] A. Taştan, M. Muma, and A. M. Zoubir, "Sparsity-aware robust community detection," *Signal Process.*, vol. 187, 2021, Art. no. 108147.

[71] G. Csurka, D. Larlus, F. Perronnin, and F. Meylan, "What is a good evaluation measure for semantic segmentation?," in *Proc. Brit. Mach. Vis. Conf.*, 2013, vol. 27, pp. 1–32.

[72] Ancortek Inc., "SDR-KIT2400AD," Accessed on: Jul. 13, 2017. [Online]. Available: http://ancortek.com/sdr-kit-2400ad

[73] A. Taştan, M. Muma, and A. M. Zoubir, "An unsupervised approach for graph-based robust clustering of human gait signatures," in *Proc. IEEE Radar Conf.*, 2020, pp. 1–6.

[74] W. H. Wolberg and O. L. Mangasarian, "Multisurface method of pattern separation applied to breast cytology diagnosis," in *Proc. Nat. Acad. Sci*, vol. 87, pp. 9193–9196, 1989.

[75] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Ann. Eugenics*, vol. 7, pp. 179–188, 1936.

[76] R. P. Gorman and T. J. Sejnowski, "Analysis of hidden units in a layered network trained to classify sonar targets," *Neural Netw.*, vol. 1, pp. 75–89, 1988.

[77] V. G. Sigilitto, S. P. Wing, L. V. Hutton, and K. B. Baker, "Classification of radar returns from the ionosphere using neural networks," *Johns Hopkins APL Tech. Dig.*, vol. 10, pp. 262–266, 1989.

[78] B. Antal and A. Hajdu, "An ensemble-based system for automatic screening of diabetic retinopathy," *Knowl. Based Syst.*, vol. 60, pp. 20–27, 2014.

[79] P. K. Wagner, S. M. Peres, R. C. B. Madeo, C. A. M. Lima, and F. A. Freitas, "Gesture unit segmentation using spatial-temporal information and machine learning," in *Proc. 27th Int. Flairs Conf.*, 2014, pp. 101–106.

[80] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene parsing through ADE20K dataset," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 633–641.

[81] K. D. Polyzos, Q. Lu, and G. B. Giannakis, "Ensemble Gaussian processes for online learning over graphs with adaptivity and scalability," *IEEE Trans. Signal Process.*, vol. 70, pp. 17–30, 2022.