SIAM J. SCI. COMPUT.
Vol. 39, No. 6, pp. A2508–A2537

THE FIEDLER VECTOR OF A LAPLACIAN TENSOR FOR HYPERGRAPH PARTITIONING*

YANNAN CHEN†, LIQUN QI‡, AND XIAOYAN ZHANG§

Abstract. Based on recent advances in spectral hypergraph theory [L. Qi and Z. Luo, *Tensor Anaysis: Spectral Theory and Special Tensors*, SIAM, Philadelphia, 2017], we explore the Fiedler vector of an even-uniform hypergraph, which is the Z-eigenvector associated with the second smallest Z-eigenvalue of a normalized Laplacian tensor arising from the hypergraph. Then, we develop a novel tensor-based spectral method for partitioning vertices of the hypergraph. For this purpose, we extend the normalized Laplacian matrix of a simple graph to the normalized Laplacian tensor of an even-uniform hypergraph. The corresponding Fiedler vector is related to the Cheeger constant of the hypergraph. Then, we establish a feasible optimization algorithm to compute the Fiedler vector according to the normalized Laplacian tensor. The convergence of the proposed algorithm and the probability of obtaining the Fiedler vector of the hypergraph are analyzed theoretically. Finally, preliminary numerical experiments illustrate that the new approach based on a hypergraph-based Fiedler vector is effective and promising for some combinatorial optimization problems arising from subspace partitioning and face clustering.

Key words. eigenvalue and eigenvector, face clustering, Fiedler vector, hypergraph partitioning, Laplacian tensor, optimization

AMS subject classifications. 05C50, 05C65, 05C90, 65F15, 90C90

DOI. 10.1137/16M1094828

1. Introduction.

Graph and hypergraph partitioning find groups and clusters in a set of objects according to their inherent similarity and affinity. If pairwise similarity among objects is available, graph-based models could be constructed and spectral graph theory provides a unified and heuristic approach for graph partitioning [15, 16]. However, pairwise relations are unprofitable in some situations. For example, to separate points on two intersecting circles as shown in Figure 1, the distance between two points is useless [25]. To model multiwise similarity, hypergraphs were established and studied in many disciplines. Duchenne et al. [19] and Rota Bulò and Pelillo [51] applied hypergraphs for face clustering and object matching. Zien, Schlag, and Chan [60] and Karypis et al. [35] used hypergraph partitioning to design very large scale integration systems. In large scale parallel scientific computing [8, 9, 26, 36, 55], hypergraph partitioning methods provided various heuristics for

*Submitted to the journal's Methods and Algorithms for Scientific Computing section September 21, 2016; accepted for publication (in revised form) September 1, 2017; published electronically November 9, 2017.
http://www.siam.org/journals/sisc/39-6/M109482.html
Funding: The first author's work was supported by the National Natural Science Foundation of China (grants 11401539 and 11771405), the Development Foundation for Excellent Youth Scholars of Zhengzhou University (grant 1421315070), and the Hong Kong Polytechnic University Postdoctoral Fellowship. The second author's work was partially supported by the Hong Kong Research Grant Council (grants PolyU 501212, 501913, 15302114, and 15300715). The third author's work was supported by the National Natural Science Foundation of China (grant 11471003) and the Qing Lan Project.
†School of Mathematics and Statistics, Zhengzhou University, Zhengzhou 450001, China (ynchen@zzu.edu.cn).
‡Department of Applied Mathematics, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong (maqilq@polyu.edu.hk).
§School of Mathematical Sciences & Institute of Mathematics, Nanjing Normal University, Nanjing 210023, China (zhangxiaoyan@njnu.edu.cn).

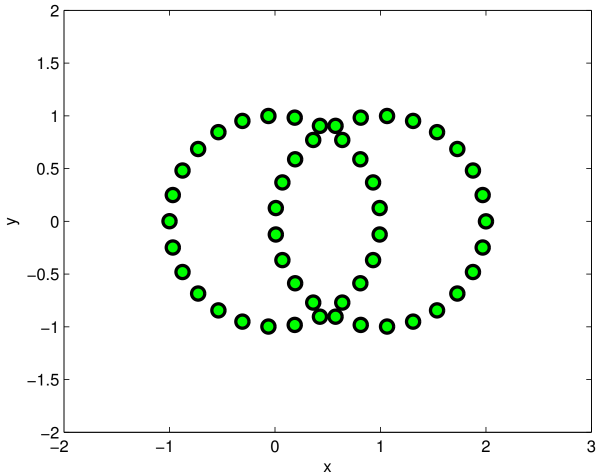


Fig. 1. *Intersecting circles.*

assigning computational loads to multiprocessors.

In 1975, Fiedler [21] first proposed using the eigenvector corresponding to the second smallest eigenvalue of a graph Laplacian matrix to construct an effective partition for vertices of the graph. Indeed, the Laplacian matrix of a graph is symmetric and positive semidefinite. The smallest eigenvalue of the Laplacian matrix is zeros associated with an all-one eigenvector. Hence, this smallest eigenpair is trivial. Fiedler [20] called the second smallest eigenvalue of a graph Laplacian matrix the algebraic connectivity of the graph because the algebraic connectivity is positive if and only if the graph is connected. To break down the connectivity and to obtain a graph partitioning, a natural idea is to explore the eigenvector corresponding to the algebraic connectivity of the graph.

With the aid of efficient numerical algorithms for a spectrum of matrices [24, 41, 56], the Fiedler vector of a graph is well-studied and widely used in science and engineering [15, 42, 44, 53]. To generalize this spectral method to the case of a hypergraph, there are mostly two kinds of approaches. The first one is to approximate the hypergraph by a graph [3, 36, 14]. A common approach is to expand an edge of the hypergraph by a clique defined on the same vertex set. Then, graph-based approaches are applied for constructing a hypergraph partitioning [30, 60, 50]. However, in general cases, any graph substitutions for a hypergraph are unsatisfactory theoretically [31].

The tensor representation for a hypergraph is the other sort of method. Since tensor and hypergraph are generalizations of matrix and graph, respectively, tensor approaches for hypergraphs attract more and more attention from researchers. In 2012, Hu and Qi [28] first defined a Laplacian tensor for an even-uniform hypergraph. They proved that the smallest Z-eigenvalue of the Laplacian tensor is zero. Furthermore, the hypergraph is connected if and only if the second smallest Z-eigenvalue of the Laplacian tensor is positive. Hence, the second smallest Z-eigenvalue was called the algebraic connectivity of the hypergraph. Li, Qi, and Yu [39] and Xie and Chang [59] gave two variations of this kind of Laplacian tensor for an even-uniform hypergraph. For odd- and even-uniform hypergraphs, Qi [47] proposed another definition of a Laplacian tensor. He verified that the smallest H-eigenvalue of Qi's Laplacian tensor is zero. Bu, Fan, and Zhou [7] argued that zero is also a Z-eigenvalue of Qi's Laplacian tensor. A normalized version of Qi's Laplacian tensor for a uniform hypergraph was considered in [29].

On the other hand, inspired by the duality between the adjacency matrix and the Laplacian matrix of a graph [58], many researchers studied the adjacency tensor defined for a uniform hypergraph [17]. Duchenne et al. [19] proposed using the Z-eigenvector corresponding to the largest Z-eigenvalue of the adjacency tensor of a hypergraph to perform object matching. The tensor trace maximization approach was also related to eigenvalues of the adjacency tensor of a uniform hypergraph [4, 23]. Rota Bulò and Pelillo [51] extracted clusters of a hypergraph using similarities represented in the adjacency tensor. In fact, this model is related to the Lagrangian number of the hypergraph [37, 34]. Govindu [25] and Shashua, Zass, and Hazan [52] preferred to decompose the nonnegative adjacency tensor of a hypergraph. Then, a probabilistic clustering was developed for problems of motion segmentation and face clustering.

In this paper, we present a new definition of the normalized Laplacian tensor of an even-uniform hypergraph, which is a generalization of the normalized Laplacian matrix of a simple graph. We verify that the smallest Z-eigenvalue of the normalized Laplacian tensor is zero. The hypergraph is connected if and only if the second smallest Z-eigenvalue of the normalized Laplacian tensor (i.e., the algebraic connectivity of the hypergraph) is positive. Then, the Fiedler vector of an even-uniform hypergraph is defined as the Z-eigenvector of the normalized Laplacian tensor corresponding to the algebraic connectivity. We prove that the Fiedler vector is related to the Cheeger constant of the hypergraph.

Since the smallest Z-eigenvalue of the normalized Laplacian tensor of an even-uniform hypergraph is trivial, we investigate a compact Laplacian tensor, whose smallest Z-eigenvalue is equal to the algebraic connectivity. To obtain the smallest Z-eigenvalue of the compact Laplacian tensor, we propose a trust region algorithm for minimizing a zero-order homogeneous function under a spherical constraint. At each iteration, a quadratic approximation of the objective function is minimized in a proper trust region. Whereafter, we apply the Cayley transform for preserving iterates on the sphere. We prove that the sequence of iterates generated by the trust region algorithm converges to a Z-eigenvector of the compact Laplacian tensor. When we start the trust region algorithm from multiple random initial points sampled from a sphere uniformly, we could obtain the algebraic connectivity of the hypergraph with a high probability. Then, it is straightforward to get the Fiedler vector of the even-uniform hypergraph from the Z-eigenvector corresponding to the smallest Z-eigenvalue of the compact Laplacian tensor.

Finally, we apply the Fiedler vector heuristics for partitioning even-uniform hypergraphs arising from subspace partitioning and face clustering. Compared with some existing methods, the new approach based on the distribution of components of the Fiedler vector of a hypergraph is effective and promising. In image segmentation, with the aid of superpixels, the Fiedler vector of an even-uniform hypergraph could extract interesting objects from images without being supervised.

The outline of this paper is drawn as follows. The new normalized Laplacian tensor and the Fiedler vector of an even-uniform hypergraph are studied in section 2. The relationship between the Fiedler vector and the Cheeger constant of the hypergraph is also discussed here. In section 3, we propose a trust region algorithm and a global strategy for computing the Fiedler vector of an even-uniform hypergraph. Section 4 illustrates applications of the Fiedler vector heuristics for hypergraph partitioning and clustering. Finally, some concluding remarks are presented in section 5.

2. Hypergraph partitioning using a Fiedler vector.

2.1. Preliminary and motivation. Spectral tensor theory provides useful instruments for spectral hypergraph theory. First, we introduce some basic conceptions on tensors. Let $\mathbb{R}^{[k,n]}$ be the space of kth order n dimensional symmetric tensors. As $\mathcal{T} \in \mathbb{R}^{[k,n]}$, we have

$$\mathcal{T} = [t_{i_1 i_2 \cdots i_k}], \quad \text{where } i_s = 1, 2, \ldots, n \text{ and } s = 1, 2, \ldots, k.$$

By the symmetry of $\mathcal{T}$, the value of $t_{i_1 i_2 \cdots i_k}$ is invariable under any permutation of its indices. There is a one-to-one correspondence between a symmetric tensor $\mathcal{T}$ and a homogeneous polynomial

$$\mathcal{T}\mathbf{x}^k \equiv \sum_{i_1=1}^{n} \cdots \sum_{i_k=1}^{n} t_{i_1 \cdots i_k} x_{i_1} \cdots x_{i_k}. \tag{1}$$

The above equality could be viewed as a product of the tensor $\mathcal{T}$ and a vector $\mathbf{x} \in \mathbb{R}^n$. Moreover, we define a vector $\mathcal{T}\mathbf{x}^{k-1} \in \mathbb{R}^n$ and a symmetric matrix $\mathcal{T}\mathbf{x}^{k-2} \in \mathbb{R}^{n\times n}$ in a componentwise manner as

$$[\mathcal{T}\mathbf{x}^{k-1}]_i \equiv \sum_{i_2=1}^{n} \cdots \sum_{i_k=1}^{n} t_{i i_2 \cdots i_k} x_{i_2} \cdots x_{i_k} \quad \text{for } i = 1, 2, \ldots, n,$$
$$[\mathcal{T}\mathbf{x}^{k-2}]_{ij} \equiv \sum_{i_3=1}^{n} \cdots \sum_{i_k=1}^{n} t_{ij i_3 \cdots i_k} x_{i_3} \cdots x_{i_k} \quad \text{for } i, j = 1, 2, \ldots, n,$$

respectively. In fact, $\mathcal{T}\mathbf{x}^k = \mathbf{x}^\top(\mathcal{T}\mathbf{x}^{k-1}) = \mathbf{x}^\top(\mathcal{T}\mathbf{x}^{k-2})\mathbf{x}$. Next, we consider a product of the tensor $\mathcal{T} \in \mathbb{R}^{[k,n]}$ and a matrix $Q = [q_{ij}] \in \mathbb{R}^{n\times \ell}$. The resulting tensor $\mathcal{T}Q^k \in \mathbb{R}^{[k,\ell]}$ has elements

$$[\mathcal{T}Q^k]_{j_1 \cdots j_k} \equiv \sum_{i_1=1}^{n} \cdots \sum_{i_k=1}^{n} t_{i_1 \cdots i_k} q_{i_1 j_1} \cdots q_{i_k j_k}$$

for $j_s = 1, 2, \ldots, \ell$ and $s = 1, 2, \ldots, k$.

Eigenvalues and eigenvectors of a tensor were defined by Qi [45], Qi and Luo [48], and Lim [40] in 2005 independently. If there exist a scalar $\lambda \in \mathbb{R}$ and a nonzero vector $\mathbf{x} \in \mathbb{R}^n$ such that

$$\mathcal{T}\mathbf{x}^{k-1} = \lambda \mathbf{x} \quad \text{and} \quad \mathbf{x}^\top \mathbf{x} = 1,$$

we call λ a Z-eigenvalue of $\mathcal{T}$ and $\mathbf{x}$ its associated Z-eigenvector. Qi [45] proved that Z-eigenvalues of a tensor are invariant under orthogonal transformations.

Second, let $G = (V, E, \mathbf{w})$ be a k-uniform hypergraph, where $V = \{1, 2, \ldots, n\}$ is the vertex set, $E = \{e_p \subseteq V : |e_p| = k \text{ for } p = 1, 2, \ldots, m\}$ is the edge set, and $\mathbf{w} = [w_p] \in \mathbb{R}^m$ is a positive vector whose component w_p denotes the weight of an edge $e_p \in E$. Here, $|\cdot|$ means the cardinality of a set. For each vertex $i \in V$, its degree is $d_i = \sum_{e_p \in E: i \in e_p} w_p$. We assume that $\mathbf{d} = [d_i] \in \mathbb{R}^n$ is positive, i.e., the hypergraph has no isolated vertices. If k is even, G is called an even-uniform hypergraph. Specially, G is a simple graph if $k = 2$. For example, Figures 2 and 3 illustrate a simple graph and a 4-uniform hypergraph, respectively.

Two vertices i and j are called connected if there is a finite sequence of vertices $\{i, \ell_1, \ell_2, \ldots, \ell_t, j\}$ such that every two adjacent vertices belong to one edge of G. If any two vertices in G are connected to each other, G is called a connected hypergraph. A connected component of a hypergraph G is a connected subhypergraph which is not contained in any connected subhypergraph of G having more vertices or edges.

We now consider the problem of bipartitioning the vertex set V of G [21]. For a nonempty and proper subset X of V, we denote $\overline{X} \equiv V \backslash X$. Assigning two different labels (say ± 1) to the partition $(X, \overline{X})$, we get an indicator $\mathbf{x} = [x_i] \in \mathbb{R}^n$ such that $x_i = 1$ if $i \in X$ and $x_i = -1$ if $i \in \overline{X}$.

At the beginning, we review the case of a simple graph, i.e., $k = 2$. If there is an edge $\{i, j\} = e_p \in E$ such that $i \in X$ and $j \in \overline{X}$, we should pay a weighted square cost for cutting this edge $w_p(x_i - x_j)^2$. Otherwise, there is no cost since $\{i, j\} \subseteq X$ or $\{i, j\} \subseteq \overline{X}$. A natural objective of a good partition is to minimize the total cost for cutting all edges connecting X and $\overline{X}$:

$$\mathbf{x}^\top L \mathbf{x} = \sum_{\{i,j\}=e_p \in E} w_p(x_i - x_j)^2. \tag{2}$$

Setting $x_i = 1$ if $i \in X$ and $x_i = -1$ if $i \in \overline{X}$, the number of cut edges for this partition is $\frac{1}{4}\mathbf{x}^\top L \mathbf{x}$. Here, the graph Laplacian $L \in \mathbb{R}^{n \times n}$ is a symmetric positive semidefinite matrix. The smallest eigenvalue of L is zero associated with an all-one eigenvector. Fiedler [20, 21] called the second smallest eigenvalue of L the algebraic connectivity of G and proposed using the corresponding eigenvector (Fiedler vector) to construct a graph partitioning.

Example 1. The graph Laplacian matrix L could be established directly. For instance, we consider a simple graph shown in Figure 2. For this graph $G = (V, E, \mathbf{w})$, we see that $V = \{1, 2, 3, 4\}$, $E = \{e_1 = \{1, 2\}, e_2 = \{1, 3\}, e_3 = \{2, 3\}, e_4 = \{3, 4\}\}$, and $\mathbf{w} = (5, 5, 5, 1)^\top$.

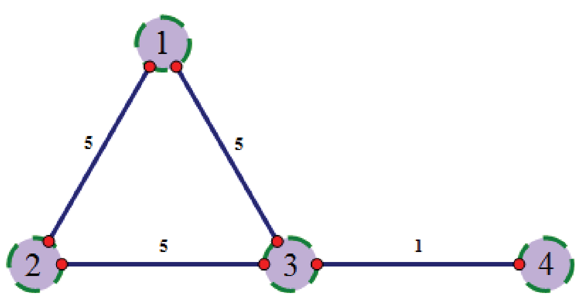


Fig. 2. *A simple graph.*

Generally speaking, there are two ways to establish the Laplacian matrix of G. On the one hand, we define the adjacency matrix of G as

$$A = \begin{bmatrix} 0 & 5 & 5 & 0 \\ 5 & 0 & 5 & 0 \\ 5 & 5 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

Calculating the sums of each row of A, we get the degree vector $\mathbf{d} = (10, 10, 11, 1)^\top$ and the degree matrix $D = \mathrm{diag}(\mathbf{d})$. Then, using a "minus" operation, we compute the Laplacian matrix of G by

$$L = D - A = \begin{bmatrix} 10 & -5 & -5 & 0 \\ -5 & 10 & -5 & 0 \\ -5 & -5 & 11 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix}. \tag{3}$$

On the other hand, we define the edge-vertex incidence matrix of G

$$M = \begin{bmatrix} 1 & -1 & 0 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix}.$$

Using a "multiply" operation, we obtain the Laplacian matrix by

$$L = M^\top \operatorname{diag}(\mathbf{w}) M = \begin{bmatrix} 10 & -5 & -5 & 0 \\ -5 & 10 & -5 & 0 \\ -5 & -5 & 11 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix}. \tag{4}$$

By multiplying $\mathbf{x}$ to both sides of $M^\top \operatorname{diag}(\mathbf{w})M$, we see that (4) is consistent with (2). Clearly, the "minus" form (3) and the "multiply" form (4) are equivalent for graphs.

In 2012, Hu and Qi [28] generalized the Laplacian matrix of a graph to a Laplacian tensor of an even-uniform hypergraph, using the star distance [33]. For each edge $e_p \in E$, we imagine an auxiliary vertex with label $\overline{x}_p \equiv \frac{1}{k}\sum_{i\in e_p} x_i$. An edge e_p of a hypergraph is replaced by an inauthentic star centered at this auxiliary vertex. The cost for cutting this edge e_p of a hypergraph is the sum of costs for cutting all edges of the star $\sigma w_p \sum_{i\in e_p}(x_i - \overline{x}_p)^k$, where $\sigma \equiv \frac{k^k}{(k-1)^k+k-1}$ is a constant. We remark here that the auxiliary vertex of a hyperedge is unreal and its value $\overline{x}_p$ is variable for various partitions of vertices in the hyperedge. If a hyperedge is cut, labels of some vertices in the hyperedge are $+1$ and labels of other ones are -1. Then, the label of the auxiliary vertex is $-1 < \overline{x}_p < 1$, and hence the cost for cutting the hyperedge is positive. Otherwise, all labels of vertices in the hyperedge are the same, and so it is the label of the auxiliary vertex. Then, there is no cost. Therefore, the cost of a partition $(X, \overline{X})$ can be written approximately as

$$\sum_{e_p\in E} \sigma w_p \sum_{i\in e_p} \left(x_i - \frac{1}{k}\sum_{j\in e_p} x_j \right)^k. \tag{5}$$

The homogeneous polynomial (5) corresponds to a unique symmetric tensor $\mathcal{L} \in \mathbb{R}^{[k,n]}$, which is called the Laplacian tensor of an even-uniform hypergraph. The cost (5) is only an approximation of the number of cut edges in a partition $(X, \overline{X})$. We note that Hu and Qi [28] studied unweighted hypergraphs, i.e., $\mathbf{w} \equiv (1, \ldots, 1)^\top$. Similar results hold for weighted hypergraphs. If $k = 2$, (5) reduces to (2) and hence $\mathcal{L}$ reduces to the graph Laplacian L.

Example 2. To see a detailed Laplacian tensor $\mathcal{L}$, we take a 4-uniform hypergraph illustrated in Figure 3, for example. Obviously, we have $V = \{1, 2, \ldots, 10\}$, $E = \{\{1, 2, 5, 6\}, \{1, 3, 5, 7\}, \{2, 3, 6, 8\}, \{3, 4, 9, 10\}\}$, and $\mathbf{w} = (5, 5, 5, 1)^\top$.

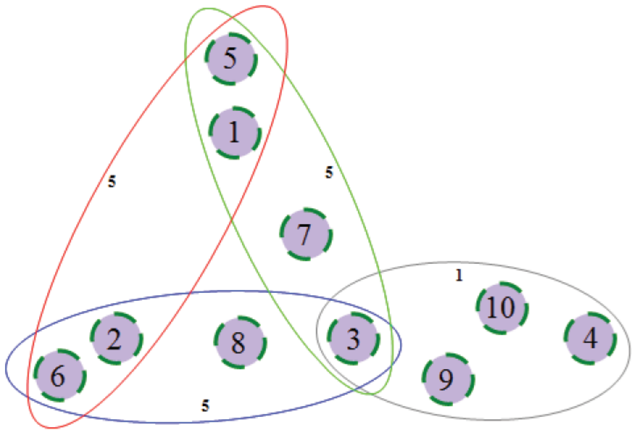


FIG. 3. *A 4-uniform hypergraph.*

According to (1) and (5), the Laplacian tensor $\mathcal{L} \in \mathbb{R}^{[4,10]}$ has $\binom{4+10-1}{4} = 715$ independent elements by symmetry. By examining the homogeneous polynomial (5), we find that there are 128 similar items. Hence, the Laplacian tensor $\mathcal{L}$ has 128 nonzero independent elements, which are shown in Table 1. Obviously, off-diagonal elements of the Laplacian tensor $\mathcal{L}$ may be positive, negative, and zero. By the symmetric decomposition formula similar to the one addressed in (9), we say that the Laplacian tensor (5) of even-uniform hypergraphs is a generalization of the "multiply" form (4) but not a "minus" form (3).

The normalized Laplacian matrix $\widetilde{L} \in \mathbb{R}^{n\times n}$ of a graph was studied in [15]. Using $\widetilde{L}$, Shi and Malik [53] proposed a normalized graph partitioning whose objective is

$$\mathbf{x}^\top \widetilde{L}\mathbf{x} = \sum_{\{i,j\}=e_p\in E} w_p \left(\frac{x_i}{\sqrt{d_i}} - \frac{x_j}{\sqrt{d_j}}\right)^2. \tag{6}$$

Whereafter, normalized partitioning methods were widely studied and applied in science and engineering [27, 57]. The main advantage of normalized partitioning methods is the robustness for outliers; i.e., the resulting partition of vertices is more balanced.

Comparing the Laplacian matrix (2) with the normalized Laplacian matrix (6) of a simple graph, we see that the normalization makes all diagonal elements of the graph Laplacian be one, i.e., coefficients of (x_i^2)'s items are all one. Applying the idea of normalization for the homogeneous polynomial (5), we obtain a new cost of a partition $(X, \overline{X})$ which is approximately

$$\sum_{e_p\in E} \sigma w_p \sum_{i\in e_p} \left(\frac{x_i}{\sqrt[k]{d_i}} - \frac{1}{k}\sum_{j\in e_p}\frac{x_j}{\sqrt[k]{d_j}}\right)^k. \tag{7}$$

By some calculations, we see that the coefficient of x_i^k is one for all $i \in V$. In this way, all vertices of the hypergraph are treated equally. Note that the homogeneous polynomial (7) determines a unique symmetric tensor $\widetilde{\mathcal{L}} \in \mathbb{R}^{[k,n]}$. In this sense, we will propose the new normalized Laplacian tensor $\widetilde{\mathcal{L}}$ which satisfies $\widetilde{\mathcal{L}}\mathbf{x}^k$ being (7).

TABLE 1
Nonzero independent elements of a Laplacian tensor $\mathcal{L} = \frac{1}{21}[\ell_{i_1,i_2,i_3,i_4}]$.

$\ell_{1,1,1,1} = 210$	$\ell_{1,1,1,2} = -35$	$\ell_{1,1,1,3} = -35$	$\ell_{1,1,1,5} = -70$
$\ell_{1,1,1,6} = -35$	$\ell_{1,1,1,7} = -35$	$\ell_{1,1,2,2} = 25$	$\ell_{1,1,2,5} = 5$
$\ell_{1,1,2,6} = 5$	$\ell_{1,1,3,3} = 25$	$\ell_{1,1,3,5} = 5$	$\ell_{1,1,3,7} = 5$
$\ell_{1,1,5,5} = 50$	$\ell_{1,1,5,6} = 5$	$\ell_{1,1,5,7} = 5$	$\ell_{1,1,6,6} = 25$
$\ell_{1,1,7,7} = 25$	$\ell_{1,2,2,2} = -35$	$\ell_{1,2,2,5} = 5$	$\ell_{1,2,2,6} = 5$
$\ell_{1,2,5,5} = 5$	$\ell_{1,2,5,6} = -15$	$\ell_{1,2,6,6} = 5$	$\ell_{1,3,3,3} = -35$
$\ell_{1,3,3,5} = 5$	$\ell_{1,3,3,7} = 5$	$\ell_{1,3,5,5} = 5$	$\ell_{1,3,5,7} = -15$
$\ell_{1,3,7,7} = 5$	$\ell_{1,5,5,5} = -70$	$\ell_{1,5,5,6} = 5$	$\ell_{1,5,5,7} = 5$
$\ell_{1,5,6,6} = 5$	$\ell_{1,5,7,7} = 5$	$\ell_{1,6,6,6} = -35$	$\ell_{1,7,7,7} = -35$
$\ell_{2,2,2,2} = 210$	$\ell_{2,2,2,3} = -35$	$\ell_{2,2,2,5} = -35$	$\ell_{2,2,2,6} = -70$
$\ell_{2,2,2,8} = -35$	$\ell_{2,2,3,3} = 25$	$\ell_{2,2,3,6} = 5$	$\ell_{2,2,3,8} = 5$
$\ell_{2,2,5,5} = 25$	$\ell_{2,2,5,6} = 5$	$\ell_{2,2,6,6} = 50$	$\ell_{2,2,6,8} = 5$
$\ell_{2,2,8,8} = 25$	$\ell_{2,3,3,3} = -35$	$\ell_{2,3,3,6} = 5$	$\ell_{2,3,3,8} = 5$
$\ell_{2,3,6,6} = 5$	$\ell_{2,3,6,8} = -15$	$\ell_{2,3,8,8} = 5$	$\ell_{2,5,5,5} = -35$
$\ell_{2,5,5,6} = 5$	$\ell_{2,5,6,6} = 5$	$\ell_{2,6,6,6} = -70$	$\ell_{2,6,6,8} = 5$
$\ell_{2,6,8,8} = 5$	$\ell_{2,8,8,8} = -35$	$\ell_{3,3,3,3} = 231$	$\ell_{3,3,3,4} = -7$
$\ell_{3,3,3,5} = -35$	$\ell_{3,3,3,6} = -35$	$\ell_{3,3,3,7} = -35$	$\ell_{3,3,3,8} = -35$
$\ell_{3,3,3,9} = -7$	$\ell_{3,3,3,10} = -7$	$\ell_{3,3,4,4} = 5$	$\ell_{3,3,4,9} = 1$
$\ell_{3,3,4,10} = 1$	$\ell_{3,3,5,5} = 25$	$\ell_{3,3,5,7} = 5$	$\ell_{3,3,6,6} = 25$
$\ell_{3,3,6,8} = 5$	$\ell_{3,3,7,7} = 25$	$\ell_{3,3,8,8} = 25$	$\ell_{3,3,9,9} = 5$
$\ell_{3,3,9,10} = 1$	$\ell_{3,3,10,10} = 5$	$\ell_{3,4,4,4} = -7$	$\ell_{3,4,4,9} = 1$
$\ell_{3,4,4,10} = 1$	$\ell_{3,4,9,9} = 1$	$\ell_{3,4,9,10} = -3$	$\ell_{3,4,10,10} = 1$
$\ell_{3,5,5,5} = -35$	$\ell_{3,5,5,7} = 5$	$\ell_{3,5,7,7} = 5$	$\ell_{3,6,6,6} = -35$
$\ell_{3,6,6,8} = 5$	$\ell_{3,6,8,8} = 5$	$\ell_{3,7,7,7} = -35$	$\ell_{3,8,8,8} = -35$
$\ell_{3,9,9,9} = -7$	$\ell_{3,9,9,10} = 1$	$\ell_{3,9,10,10} = 1$	$\ell_{3,10,10,10} = -7$
$\ell_{4,4,4,4} = 21$	$\ell_{4,4,4,9} = -7$	$\ell_{4,4,4,10} = -7$	$\ell_{4,4,9,9} = 5$
$\ell_{4,4,9,10} = 1$	$\ell_{4,4,10,10} = 5$	$\ell_{4,9,9,9} = -7$	$\ell_{4,9,9,10} = 1$
$\ell_{4,9,10,10} = 1$	$\ell_{4,10,10,10} = -7$	$\ell_{5,5,5,5} = 210$	$\ell_{5,5,5,6} = -35$
$\ell_{5,5,5,7} = -35$	$\ell_{5,5,6,6} = 25$	$\ell_{5,5,7,7} = 25$	$\ell_{5,6,6,6} = -35$
$\ell_{5,7,7,7} = -35$	$\ell_{6,6,6,6} = 210$	$\ell_{6,6,6,8} = -35$	$\ell_{6,6,8,8} = 25$
$\ell_{6,8,8,8} = -35$	$\ell_{7,7,7,7} = 105$	$\ell_{8,8,8,8} = 105$	$\ell_{9,9,9,9} = 21$
$\ell_{9,9,9,10} = -7$	$\ell_{9,9,10,10} = 5$	$\ell_{9,10,10,10} = -7$	$\ell_{10,10,10,10} = 21$

Remark. Hu and Qi [29] defined a normalized Laplacian tensor $\widetilde{\mathcal{L}}_{HQ15}$ for a uniform hypergraph G. The corresponding cost of a partition $(X, \overline{X})$ is

$$\sum_{e_p \in E} w_p \left(\sum_{i \in e_p} \frac{x_i^k}{d_i} - k \prod_{j \in e_p} \frac{x_j}{\sqrt[k]{d_j}} \right). \tag{8}$$

We argue that (8) is an insufficient cost function for evaluating a partition of G. For example, we consider a 4-uniform hypergraph with eight vertices $V = \{1, 2, \dots, 8\}$, four edges $E = \{\{1,2,3,4\}, \{3,4,5,6\}, \{5,6,7,8\}, \{7,8,1,2\}\}$, and weights for all edges being one. Let $\mathbf{x} = (1, -1, 1, -1, 1, -1, 1, -1)^\top$ be an indicator that means cutting all edges of G. By some calculations, we find that (8) vanishes, which is the same value as the case that no edges of G are cut. This fact motives us to propose the new normalized Laplacian tensor. Note that the value of (7) is $\frac{512}{21} > 0$ for this example. Hence, the new normalized Laplacian tensor is favorable for partitioning even-uniform hypergraphs.

2.2. The normalized Laplacian tensor. According to the homogeneous polynomial (7), we write down the normalized Laplacian tensor formally.

LEMMA 2.1. *The normalized Laplacian tensor $\widetilde{\mathcal{L}}$ of an even-uniform hypergraph $G = (V, E, \mathbf{w})$ has a symmetric decomposition*

$$\widetilde{\mathcal{L}} = \sigma \sum_{e_p \in E} w_p \sum_{i \in e_p} \underbrace{\mathbf{u}_{p,i} \circ \mathbf{u}_{p,i} \circ \cdots \circ \mathbf{u}_{p,i}}_{k \ times}, \tag{9}$$

where $\sigma = \frac{k^k}{(k-1)^k + k - 1}$, $\mathbf{u}_{p,i} = \frac{1}{\sqrt[k]{d_i}}\mathbf{e}_i - \frac{1}{k}\sum_{j \in e_p} \frac{1}{\sqrt[k]{d_j}}\mathbf{e}_j$ *only if* $i \in e_p$, $\mathbf{e}_i$ *being the ith column of an identity matrix, and* $\circ$ *denotes the outer product. Hence,* $\widetilde{\mathcal{L}}$ *is positive semidefinite since* k *is even.*

Proof. By some calculations, we have

$$\begin{aligned}
&\left(\sigma \sum_{e_p \in E} w_p \sum_{i \in e_p} \mathbf{u}_{p,i} \circ \mathbf{u}_{p,i} \circ \cdots \circ \mathbf{u}_{p,i}\right) \mathbf{x}^k \\
&= \sum_{i_1=1}^{n} \sum_{i_2=1}^{n} \cdots \sum_{i_k=1}^{n} \sigma \sum_{e_p \in E} w_p \sum_{i \in e_p} [\mathbf{u}_{p,i}]_{i_1} [\mathbf{u}_{p,i}]_{i_2} \cdots [\mathbf{u}_{p,i}]_{i_k} x_{i_1} x_{i_2} \cdots x_{i_k} \\
&= \sigma \sum_{e_p \in E} w_p \sum_{i \in e_p} \left(\sum_{i_1=1}^{n} [\mathbf{u}_{p,i}]_{i_1} x_{i_1}\right) \left(\sum_{i_2=1}^{n} [\mathbf{u}_{p,i}]_{i_2} x_{i_2}\right) \cdots \left(\sum_{i_k=1}^{n} [\mathbf{u}_{p,i}]_{i_k} x_{i_k}\right) \\
&= \sigma \sum_{e_p \in E} w_p \sum_{i \in e_p} (\mathbf{u}_{p,i}^\top \mathbf{x})^k, \qquad (10)
\end{aligned}$$

which is exactly the homogeneous polynomial in (7). Because there is a one-to-one correspondence between a homogeneous polynomial and a symmetric tensor, we know that the normalized Laplacian tensor has the symmetric decomposition (9). ∎

Qi [45, 46] pointed out that Z-eigenvalues of a symmetric tensor are invariant under orthogonally transformations. Next, we study the smallest Z-eigenvalue of the normalized Laplacian tensor of an even-uniform hypergraph.

THEOREM 2.2. *Suppose that an even-uniform hypergraph G has s connected components and $\widetilde{\mathcal{L}}$ is its normalized Laplacian tensor. Then, the set*

$$\mathbb{K} = \left\{\mathbf{x} \in \mathbb{R}^n : \widetilde{\mathcal{L}}\mathbf{x}^{k-1} = \mathbf{0}\right\}$$

forms a linear subspace with dimension s exactly.

Proof. Let connected components of $G = (V, E, \mathbf{w})$ be G^ℓ for $\ell = 1, 2, \ldots, s$. The corresponding vertex subsets $V^1, V^2, \ldots, V^s$ are nonempty and satisfy $\bigcup_{\ell=1}^{s} V^\ell = V$ and $V^\ell \cap V^t = \emptyset$ if $\ell \neq t$.

For each connected component G^ℓ, we define a vector $\mathbf{a}^\ell \in \mathbb{R}^n$ whose elements are

$$(\mathbf{a}^\ell)_i = \begin{cases} \sqrt[k]{d_i} & \text{if } i \in V^\ell, \\ 0 & \text{otherwise.} \end{cases}$$

Obviously, vectors $\mathbf{a}^1, \ldots, \mathbf{a}^s$ are linear independent. Hence, they could span a linear subspace $\text{span}(\mathbf{a}^1, \ldots, \mathbf{a}^s)$ with dimension s. By some calculations, we have

$$\mathbf{u}_{p,i}^\top \mathbf{a}^\ell = 0 \qquad \text{for } i \in e_p \in E \text{ and } \ell = 1, \ldots, s. \tag{11}$$

Hence, for any vector $\mathbf{x} \in \operatorname{span}(\mathbf{a}^1, \ldots, \mathbf{a}^s)$, we have $\mathbf{u}_{p,i}^\top \mathbf{x} = 0$ for $e_p \in E$ and $i \in e_p$. Therefore, from (9), the product $\widetilde{\mathcal{L}}\mathbf{x}^{k-1} = \mathbf{0}$ and $\mathbf{x} \in \mathbb{K}$.

On the other hand, suppose $\mathbf{x} \in \mathbb{K}$. Then, $\widetilde{\mathcal{L}}\mathbf{x}^k = \mathbf{x}^\top(\widetilde{\mathcal{L}}\mathbf{x}^{k-1}) = 0$. Recalling (7), we have $\sum_{i\in e_p}\left(\frac{x_i}{\sqrt[k]{d_i}} - \frac{1}{k}\sum_{j\in e_p}\frac{x_j}{\sqrt[k]{d_j}}\right)^k = 0$ for all $e_p \in E$. That is to say, $\frac{x_i}{\sqrt[k]{d_i}} = \frac{1}{k}\sum_{j\in e_p}\frac{x_j}{\sqrt[k]{d_j}}$ for all $i \in e_p$. We now consider a connected component G^ℓ. It is easy to see that the values of $\left\{\frac{x_i}{\sqrt[k]{d_i}}\right\}_{i\in V^\ell}$ are the same. Hence, $\mathbf{x} \in \operatorname{span}(\mathbf{a}^1, \ldots, \mathbf{a}^s)$. Therefore, $\mathbb{K} = \operatorname{span}(\mathbf{a}^1, \ldots, \mathbf{a}^s)$ is a linear subspace with dimension s exactly. ∎

COROLLARY 2.3. *The smallest Z-eigenvalue of the normalized Laplacian tensor $\widetilde{\mathcal{L}}$ of an even-uniform hypergraph G is zero, and the associated eigenvectors form an eigenspace* $\mathbb{K} = \operatorname{span}(\mathbf{a}^1, \ldots, \mathbf{a}^s)$.

2.3. The Fiedler vector. For a connected even-uniform hypergraph, the smallest Z-eigenvalue zero of the normalized Laplacian tensor $\widetilde{\mathcal{L}}$ is trivial. Now, we focus on nontrivial Z-eigenvalues of $\widetilde{\mathcal{L}}$.

DEFINITION 2.4. *Suppose that G is an even-uniform hypergraph and $\widetilde{\mathcal{L}}$ is its normalized Laplacian tensor. Let all the Z-eigenvalues of $\widetilde{\mathcal{L}}$ be ordered as*

$$0 = \lambda_0 \leq \lambda_1 \leq \cdots .$$

Then, we call λ_1 the algebraic connectivity of G, denoted as $\lambda_1(G)$. Moreover, the Z-eigenvector of $\widetilde{\mathcal{L}}$ corresponding to $\lambda_1(G)$ is called the Fiedler vector of G.

Immediately, we get the following property on the second smallest Z-eigenvalue of the normalized Laplacian tensor $\widetilde{\mathcal{L}}$ of G.

COROLLARY 2.5. *An even-uniform hypergraph G is connected if and only if its algebraic connectivity $\lambda_1(G)$ is positive.*

Next, we show a variational characterization for the Fiedler vector of the hypergraph.

THEOREM 2.6. *Suppose that G is an even-uniform hypergraph and $\widetilde{\mathcal{L}}$ is its normalized Laplacian tensor. Then, the algebraic connectivity of G could be characterized as*

$$\lambda_1(G) = \begin{cases} \min & \widetilde{\mathcal{L}}\mathbf{x}^k \\ s.t. & \mathbf{x}^\top\mathbf{x} = 1, \mathbf{x}^\top\boldsymbol{\delta} = 0, \end{cases} \tag{12}$$

where $\boldsymbol{\delta} = [\sqrt[k]{d_i}] \in \mathbb{K}$. Moreover, the optimal solution of (12) *is the Fiedler vector of G.*

Proof. First, we assume that G has multiple ($s \geq 2$) connected components. By Corollary 2.3, we know $\lambda_1 = \lambda_0 = 0$. Furthermore, there exists a unit vector $\mathbf{x} \in \mathbb{K} = \operatorname{span}(\mathbf{a}^1, \ldots, \mathbf{a}^s)$ such that $\mathbf{x}^\top\boldsymbol{\delta} = 0$ and $\widetilde{\mathcal{L}}\mathbf{x}^{k-1} = \mathbf{0}$. Hence, $\mathbf{x}$ is a feasible solution of (12) with objective $\widetilde{\mathcal{L}}\mathbf{x}^k = \mathbf{x}^\top(\widetilde{\mathcal{L}}\mathbf{x}^{k-1}) = 0$. Since k is even, $\widetilde{\mathcal{L}}\mathbf{x}^k \geq 0$ for all $\mathbf{x} \in \mathbb{R}^n$. Hence, the global minimum of (12) is also zero.

Second, we consider the case that G is connected. Suppose that $\mathbf{x}_1$ is a Z-eigenvector of $\widetilde{\mathcal{L}}$ associated with the algebraic connectivity $\lambda_1 > 0$. Then, we have $\lambda_1 = \lambda_1\mathbf{x}_1^\top\mathbf{x}_1 = \mathbf{x}_1^\top(\widetilde{\mathcal{L}}\mathbf{x}_1^{k-1}) = \widetilde{\mathcal{L}}\mathbf{x}_1^k$. Hence, λ_1 is the minimum value of the following optimization problem:

$$\lambda_1 = \min\{\widetilde{\mathcal{L}}\mathbf{x}^k : \widetilde{\mathcal{L}}\mathbf{x}^{k-1} = \lambda\mathbf{x}, \mathbf{x}^\top\mathbf{x} = 1, \lambda > 0\}. \tag{13}$$

Next, we turn to (12). Since $\mathbf{x}^\top\mathbf{x} = 1$, $\mathbf{x} \neq \mathbf{0}$. By $\mathbf{x}^\top\boldsymbol{\delta} = 0$, we know that gradients $2\mathbf{x}$ and $\boldsymbol{\delta}$ of equality constraints in (12) are linear independent. According to the KKT condition [43, 54], the optimal solution $\mathbf{x}$ of (12) satisfies

$$\widetilde{\mathcal{L}}\mathbf{x}^{k-1} = \lambda\mathbf{x} + \mu\boldsymbol{\delta},$$

where scalars λ and μ are Lagrangian multipliers. From (11), we have $\boldsymbol{\delta}^\top(\widetilde{\mathcal{L}}\mathbf{x}^{k-1}) = 0$. Moreover, by $\boldsymbol{\delta}^\top\mathbf{x} = 0$, we get $\mu\boldsymbol{\delta}^\top\boldsymbol{\delta} = 0$. Hence, $\mu = 0$ and $\widetilde{\mathcal{L}}\mathbf{x}^{k-1} = \lambda\mathbf{x}$. Therefore, the optimization problem (12) is equal to

$$\min\{\widetilde{\mathcal{L}}\mathbf{x}^k : \widetilde{\mathcal{L}}\mathbf{x}^{k-1} = \lambda\mathbf{x}, \mathbf{x}^\top\mathbf{x} = 1, \mathbf{x}^\top\boldsymbol{\delta} = 0\}. \tag{14}$$

Finally, we prove the equivalence between (13) and (14). If $\lambda > 0$, we take inner products of both sides of $\widetilde{\mathcal{L}}\mathbf{x}^{k-1} = \lambda\mathbf{x}$ with $\boldsymbol{\delta}$ and get $\mathbf{x}^\top\boldsymbol{\delta} = 0$ since $\boldsymbol{\delta}^\top(\widetilde{\mathcal{L}}\mathbf{x}^{k-1}) = 0$ by (11). Conversely, we assume $\mathbf{x}^\top\boldsymbol{\delta} = 0$. Since $\mathbf{x}$ is a Z-eigenvector of $\widetilde{\mathcal{L}}$ associated with Z-eigenvalue $\lambda = \widetilde{\mathcal{L}}\mathbf{x}^k$, we know $\lambda \geq 0$. Because G is connected, $\mathbb{K} = \{c\boldsymbol{\delta}\}$, where c is an arbitrary scalar by Corollary 2.3. According to $\mathbf{x}^\top\mathbf{x} = 1$ and $\mathbf{x}^\top\boldsymbol{\delta} = 0$, we get that $\mathbf{x}$ does not belong to $\mathbb{K}$ and hence $\lambda \neq 0$. Therefore, we have $\lambda > 0$. The proof is complete. ∎

Whereafter, we reveal a theorem which is a generalization of one part of the Cheeger inequality [15]. Before we start, we prove a lemma.

Lemma 2.7. *For an even k, we have*

$$q(k-q)^k + (k-q)q^k \leq (k-1)^k + k - 1$$

for all $q = 1, 2, \ldots, k-1$.

Proof. Let

$$p(x) \equiv x(1-x)^k + (1-x)x^k.$$

We are going to prove a stronger inequality

$$p(x) \leq p(\tfrac{1}{k}) \qquad \text{for all } x \in \left[\tfrac{1}{k}, 1 - \tfrac{1}{k}\right]. \tag{15}$$

Multiplying k^{k+1} to both sides of (15) and taking $x = \frac{q}{k}$, we get this lemma.

The inequality (15) is trivial if $k = 2$. We now consider the case of $k = 4$. Owing to $p(x) = p(1-x)$, we only need to verify (15) for $x \in [\frac{1}{4}, \frac{1}{2}]$. By some calculations, we have

$$p'(x) = (1-2x)(6x^2 - 6x + 1).$$

Since $p'(x) < 0$ for $x \in [\frac{1}{4}, \frac{1}{2})$, $p(x)$ decreases monotonously. Hence, the inequality (15) holds for $k = 4$.

Next, we assume $k \geq 6$ and prove that (15) is valid for $x \in [\frac{1}{k}, \frac{2}{k}]$. By some calculations, we obtain

$$\begin{aligned} p'(x) &= x^{k-1}(k - kx - x) - (1-x)^{k-1}(kx + x - 1) \\ &< x^{k-1}(1-x)(k+1) - (1-x)^{k-1}x \\ &= x(1-x)[(k+1)x^{k-2} - (1-x)^{k-2}], \end{aligned}$$

where the above inequality holds because $kx - 1 \geq 0$ as $x \in [\frac{1}{k}, \frac{2}{k}]$. Since $(k+1)x^{k-2} - (1-x)^{k-2}$ increases monotonically in x, we know

$$\begin{aligned} p'(x) &< x(1-x)[(k+1)(\tfrac{2}{k})^{k-2} - (1 - \tfrac{2}{k})^{k-2}] \\ &= \frac{x(1-x)}{k^{k-2}}[(k+1)2^{k-2} - (k-2)^{k-2}]. \end{aligned}$$

It is straightforward to see that $(k+1)2^{k-2}-(k-2)^{k-2} < 0$ for $k \geq 6$. Hence, $p'(x) < 0$ and the inequality (15) holds for $k \geq 6$ and $x \in [\frac{1}{k}, \frac{2}{k}]$ as well as $x \in [1-\frac{2}{k}, 1-\frac{1}{k}]$ for symmetry.

Finally, we focus on $x \in [\frac{2}{k}, 1-\frac{2}{k}]$. By some calculations, we have

$$p''(x) = k(1-x)^{k-2}[(k+1)x-2] + kx^{k-2}[-(k+1)x+k-1].$$

Since $x \in [\frac{2}{k}, 1-\frac{2}{k}]$, we have $(k+1)x-2 \geq \frac{2}{k}$ and $-(k+1)x+k-1 \geq \frac{2}{k}$. Hence, we obtain $p''(x) \geq 2[(1-x)^{k-2}+x^{k-2}] > 0$. Therefore, $p(x)$ is a convex function in the interval $[\frac{2}{k}, 1-\frac{2}{k}]$. Because $p(\frac{2}{k}) = p(1-\frac{2}{k})$, we get

$$p(\tfrac{2}{k}) \geq p(x) \qquad \text{for } x \in [\tfrac{2}{k}, 1-\tfrac{2}{k}].$$

Therefore, the inequality (15) is valid. ∎

Let $(X, \overline{X})$ be a partition of an even-uniform hypergraph G. We denote the volume of X as $\mathrm{vol}X = \sum_{i \in X} d_i$, the boundary of X as $\partial X = \{e_p \in E : e_p \cap X \neq \emptyset, e_p \cap \overline{X} \neq \emptyset\}$, and the cut between X and $\overline{X}$ as $\mathrm{cut}(X) = \sum_{e_p \in \partial X} w_p$. The Cheeger ratio of X is

$$h(X) = \frac{\mathrm{cut}(X)}{\min(\mathrm{vol}X, \mathrm{vol}\overline{X})}.$$

Hence, $h(X) = h(\overline{X})$. The smallest one $h_G = \min_{X \subseteq V} h(X)$ is called the Cheeger constant of G. The partition $(X, \overline{X})$ attaining the Cheeger constant $h_G = h(X)$ is an ideal partitioning for an even-uniform hypergraph G.

THEOREM 2.8. *Let $\lambda_1(G)$ be the algebraic connectivity of an even-uniform hypergraph G. Then, we have*

$$\lambda_1(G) \leq 2^{k/2} h_G. \tag{16}$$

Proof. The inequality (16) is trivial if $X = \emptyset$ or $X = V$. Next, we suppose that X is a nonempty and proper subset of V in the following analysis.

For a given subset X, we define scalars

$$\alpha = \left(\sum_{i \in X} \left(\sqrt[k]{d_i} \right)^2 \right)^{1/2}, \quad \beta = \left(\sum_{i \in \overline{X}} \left(\sqrt[k]{d_i} \right)^2 \right)^{1/2}, \quad \text{and } \gamma = \sqrt{\alpha^2 + \beta^2}.$$

Using these scalars, we define a vector $\mathbf{x} = [x_i] \in \mathbb{R}^n$ such that

$$x_i = \begin{cases} \dfrac{\beta}{\alpha\gamma} \sqrt[k]{d_i} & \text{if } i \in X, \\ \dfrac{-\alpha}{\beta\gamma} \sqrt[k]{d_i} & \text{if } i \in \overline{X}. \end{cases}$$

It is easy to see that $\mathbf{x}^\top \mathbf{x} = 1$ and $\mathbf{x}^\top \boldsymbol{\delta} = 0$. From (12), we know

$$\begin{aligned} \lambda_1(G) \leq \widetilde{\mathcal{L}}\mathbf{x}^k &= \sum_{e_p \in E} \sigma w_p \sum_{i \in e_p} \left(\frac{x_i}{\sqrt[k]{d_i}} - \frac{1}{k} \sum_{j \in e_p} \frac{x_j}{\sqrt[k]{d_j}} \right)^k \\ &= \sum_{e_p \in \partial X} \sigma w_p \sum_{i \in e_p} \left(\frac{x_i}{\sqrt[k]{d_i}} - \frac{1}{k} \sum_{j \in e_p} \frac{x_j}{\sqrt[k]{d_j}} \right)^k. \end{aligned}$$

Let $|e_p \cap X| = q_p \in \{1, \ldots, k-1\}$. Then, $|e_p \cap \overline{X}| = k - q_p$. By some calculations, we obtain

$$\begin{aligned}
\lambda_1(G) &\leq \sum_{e_p \in \partial X} \sigma w_p \left(\frac{\alpha^2+\beta^2}{\alpha\beta\gamma}\right)^k \frac{q_p(k-q_p)^k + (k-q_p)q_p^k}{k^k} \\
&\leq \sigma \frac{(k-1)^k + k - 1}{k^k} \left(\frac{\alpha^2+\beta^2}{\alpha^2\beta^2}\right)^{k/2} \sum_{e_p \in \partial X} w_p \\
&= \left(\frac{1}{\alpha^2} + \frac{1}{\beta^2}\right)^{k/2} \mathrm{cut}(X),
\end{aligned}$$

where the last inequality holds for Lemma 2.7.

For p-norms, we know $\|x\|_p \geq \|x\|_q$ if $q > p > 1$. We look at a vector $\mathbf{z} = [z_i] \in \mathbb{R}^n$ such that $z_i = \sqrt[k]{d_i}$ if $i \in X$ and $z_i = 0$ if $i \in \overline{X}$. Then, by $k \geq 2$, we get

$$\alpha = \|\mathbf{z}\| \geq \|\mathbf{z}\|_k = \left(\sum_{i \in X} d_i\right)^{1/k} = (\mathrm{vol}X)^{1/k}.$$

Similarly, $\beta \geq (\mathrm{vol}\overline{X})^{1/k}$. Without loss of generality, we suppose $\mathrm{vol}X \leq \mathrm{vol}\overline{X}$. Hence,

$$\lambda_1(G) \leq \left(\frac{1}{(\mathrm{vol}X)^{2/k}} + \frac{1}{(\mathrm{vol}\overline{X})^{2/k}}\right)^{k/2} \mathrm{cut}(X) \leq \frac{2^{k/2}}{\mathrm{vol}X}\mathrm{cut}(X) = 2^{k/2} h(X).$$

Due to the arbitrariness of $X \subseteq V$, we get this theorem. ∎

Theorem 2.8 means that the Fiedler vector of an even-uniform hypergraph generated by the optimization model (12) is well-defined.

3. Computing the Fiedler vector. Because a hypergraph may contain plenty of vertices, numerical methods for small tensors such as [18, 12] are inefficient for finding the second smallest Z-eigenvalue of the normalized Laplacian tensor and the associated Z-eigenvector. To obtain the Fiedler vector of an even-uniform hypergraph, we establish a customized approach for solving the optimization model (12). First, we give an equivalent optimization problem with a simple spherical constraint. Second, a trust region method for the spherical optimization is presented.

Because the smallest Z-eigenvalue of the normalized Laplacian tensor is zero, that is, trivial, we propose investigating a compact Laplacian tensor, whose smallest Z-eigenvalue is the algebraic connectivity of the even-uniform hypergraph. Let $\boldsymbol{\delta}^\perp \equiv \{\mathbf{x} \in \mathbb{R}^n : \mathbf{x}^\top \boldsymbol{\delta} = 0\}$ be a subspace of $\mathbb{R}^n$. Recalling (11), we get $\mathbf{u}_{p,i} \in \boldsymbol{\delta}^\perp$ for $i \in e_p \in E$. This fact motivates us to construct a basis of $\boldsymbol{\delta}^\perp$, using the Householder transform [24]. Let $\mathbf{v} = \boldsymbol{\delta} - \|\boldsymbol{\delta}\|\mathbf{e}_1$ be a Householder vector and $\beta = \frac{2}{\mathbf{v}^\top\mathbf{v}}$. The corresponding Householder matrix $P = I - \beta\mathbf{v}\mathbf{v}^\top$ satisfies $P\boldsymbol{\delta} = (\|\boldsymbol{\delta}\|, 0, \ldots, 0)^\top$. The second to last columns of P form an orthonormal basis of $\boldsymbol{\delta}^\perp$, which we denote as $Q \in \mathbb{R}^{n\times(n-1)}$. Then, $Q^\top\boldsymbol{\delta} = \mathbf{0}_{n-1}$ and $Q^\top Q = I_{n-1}$.

The compact Laplacian tensor $\mathcal{L}_C \in \mathbb{R}^{[k,n-1]}$ of an even-uniform hypergraph G is defined as

$$\mathcal{L}_C \equiv \widetilde{\mathcal{L}} Q^k = \sigma \sum_{e_p \in E} w_p \sum_{i \in e_p} \underbrace{(Q^\top\mathbf{u}_{p,i}) \circ (Q^\top\mathbf{u}_{p,i}) \circ \cdots \circ (Q^\top\mathbf{u}_{p,i})}_{k \text{ times}}, \tag{17}$$

where the last equality holds because

$$\begin{aligned}[\widetilde{\mathcal{L}}Q^k]_{j_1\cdots j_k} &= \sum_{i_1=1}^{n}\cdots\sum_{i_k=1}^{n}[\widetilde{\mathcal{L}}]_{i_1\cdots i_k}q_{i_1j_1}\cdots q_{i_kj_k} \\ [\text{for } (9)] &= \sum_{i_1=1}^{n}\cdots\sum_{i_k=1}^{n}\sigma\sum_{e_p\in E}w_p\sum_{i\in e_p}[\mathbf{u}_{p,i}]_{i_1}\cdots[\mathbf{u}_{p,i}]_{i_k}q_{i_1j_1}\cdots q_{i_kj_k} \\ &= \sigma\sum_{e_p\in E}w_p\sum_{i\in e_p}\left(\sum_{i_1=1}^{n}q_{i_1j_1}[\mathbf{u}_{p,i}]_{i_1}\right)\cdots\left(\sum_{i_k=1}^{n}q_{i_kj_k}[\mathbf{u}_{p,i}]_{i_k}\right) \\ &= \sigma\sum_{e_p\in E}w_p\sum_{i\in e_p}[Q^\top\mathbf{u}_{p,i}]_{j_1}\cdots[Q^\top\mathbf{u}_{p,i}]_{j_k}\end{aligned}$$

for $j_s = 1,2,\ldots,n-1$ and $s = 1,2,\ldots,k$.

Using the compact Laplacian tensor, we address another characterization of the algebraic connectivity of G in the following theorem.

THEOREM 3.1. *Let $\mathcal{L}_C$ be the compact Laplacian tensor of an even-uniform hypergraph G. Then, the algebraic connectivity $\lambda_1(G)$ is the smallest Z-eigenvalue of $\mathcal{L}_C$, i.e.,*

$$\lambda_1(G) = \begin{cases}\min & \mathcal{L}_C\mathbf{y}^k \\ s.t. & \mathbf{y}^\top\mathbf{y} = 1.\end{cases} \tag{18}$$

Let $\mathbf{y}^* \in \mathbb{R}^{n-1}$ be the optimal solution of the spherical optimization (18)*. Then, the Fiedler vector of G is $Q\mathbf{y}^*$.*

Proof. First, we prove the equivalence between two sets

$$\{\mathbf{x}\in\mathbb{R}^n : \mathbf{x}^\top\boldsymbol{\delta} = 0, \mathbf{x}^\top\mathbf{x} = 1\} \quad\text{and}\quad \{\mathbf{x}\in\mathbb{R}^n : \mathbf{x} = Q\mathbf{y}, \mathbf{y}^\top\mathbf{y} = 1\}. \tag{19}$$

We suppose $\mathbf{x}^\top\boldsymbol{\delta} = 0$ and $\mathbf{x}^\top\mathbf{x} = 1$. By $\mathbf{x}^\top\boldsymbol{\delta} = 0$, there exists a unique vector $\mathbf{y} \in \mathbb{R}^{n-1}$ such that $\mathbf{x} = Q\mathbf{y}$. Moreover, $\mathbf{y}^\top\mathbf{y} = \mathbf{y}^\top Q^\top Q\mathbf{y} = \mathbf{x}^\top\mathbf{x} = 1$. On the other hand, if $\mathbf{x} = Q\mathbf{y}$ and $\mathbf{y}^\top\mathbf{y} = 1$, we have $\mathbf{x}^\top\boldsymbol{\delta} = \mathbf{y}^\top Q^\top\boldsymbol{\delta} = \mathbf{y}^\top\mathbf{0}_{n-1} = 0$ and $\mathbf{x}^\top\mathbf{x} = \mathbf{y}^\top Q^\top Q\mathbf{y} = \mathbf{y}^\top\mathbf{y} = 1$. Hence, the two sets in (19) are equal.

Finally, by Theorem 2.6, we get

$$\begin{aligned}\lambda_1(G) &= \min\{\widetilde{\mathcal{L}}\mathbf{x}^k : \mathbf{x}^\top\boldsymbol{\delta} = 0, \mathbf{x}^\top\mathbf{x} = 1\} \\ &= \min\{\widetilde{\mathcal{L}}\mathbf{x}^k : \mathbf{x} = Q\mathbf{y}, \mathbf{y}^\top\mathbf{y} = 1\} \\ &= \min\{\widetilde{\mathcal{L}}(Q\mathbf{y})^k : \mathbf{y}^\top\mathbf{y} = 1\} \\ &= \min\{\mathcal{L}_C\mathbf{y}^k : \mathbf{y}^\top\mathbf{y} = 1\}.\end{aligned}$$

The proof is complete. ∎

3.1. A trust region algorithm. Since the optimization problem (18) using the compact Laplacian tensor of an even-uniform hypergraph is easier than (12) based on a normalized Laplacian tensor, we consider the following spherical optimization:

$$\begin{cases}\min & f(\mathbf{y}) = \dfrac{\mathcal{L}_C\mathbf{y}^k}{\|\mathbf{y}\|^k} \\ \text{s.t.} & \mathbf{y}\in\mathbb{S}^{n-2},\end{cases} \tag{20}$$

where the spherical constraint is $\mathbb{S}^{n-2} \equiv \{\mathbf{y} \in \mathbb{R}^{n-1} : \mathbf{y}^\top \mathbf{y} = 1\}$. The objective function $f(\mathbf{y})$ is zero-order homogeneous, while the denominator $\|\mathbf{y}\|^k$ is indeed one when $\mathbf{y} \in \mathbb{S}^{n-2}$. The advantage of using the denominator is reflected in the freedom of a multiplier corresponding to the spherical constraint.

LEMMA 3.2 (Theorem 4.1 in [10]). *Suppose that k is even and $\mathbf{y}_* \in \mathbb{S}^{n-2}$. Then, $\mathbf{y}_*$ is a first-order stationary point of $f(\cdot)$, i.e., $\mathbf{g}(\mathbf{y}_*) = 0$, if and only if $\mathbf{y}_*$ is a Z-eigenvector of the compact Laplacian tensor $\mathcal{L}_C$. The associated Z-eigenvalue of $\mathcal{L}_C$ is $f(\mathbf{y}_*)$.*

The gradient and the Hessian of the objective function $f(\mathbf{y})$ at $\mathbf{y} \in \mathbb{S}^{n-2}$ [13] are

$$\mathbf{g}(\mathbf{y}) = \frac{k}{\|\mathbf{y}\|^{k+2}} \left(\|\mathbf{y}\|^2 \cdot \mathcal{L}_C \mathbf{y}^{k-1} - \mathcal{L}_C \mathbf{y}^k \cdot \mathbf{y} \right) \tag{21}$$

and

$$\begin{aligned} H(\mathbf{y}) = {} & \frac{k(k-1)}{\|\mathbf{y}\|^k} \mathcal{L}_C \mathbf{y}^{k-2} - \frac{k^2}{\|\mathbf{y}\|^{k+2}} \left(\mathbf{y} (\mathcal{L}_C \mathbf{y}^{k-1})^\top + (\mathcal{L}_C \mathbf{y}^{k-1}) \mathbf{y}^\top \right) \\ & - \frac{k \mathcal{L}_C \mathbf{y}^k}{\|\mathbf{y}\|^{k+2}} I + \frac{k(k+2) \mathcal{L}_C \mathbf{y}^k}{\|\mathbf{y}\|^{k+4}} \mathbf{y}\mathbf{y}^\top, \end{aligned} \tag{22}$$

respectively. Obviously, $H(\mathbf{y})$ is a singular matrix since $\mathbf{y}^\top H(\mathbf{y}) \mathbf{y} = 0$.

The trust region method is an iterative algorithm [54, 43]. Suppose that the current iterate is $\mathbf{y}_c$ and we are going to find the next iterate $\mathbf{y}_{c+1}$. A quadratic model $m_c(\cdot)$ is established to approximate the objective function f around a neighborhood of $\mathbf{y}_c$. This local model leads to a trust region subproblem

$$\begin{cases} \min & m_c(\mathbf{s}) \equiv f(\mathbf{y}_c) + \mathbf{g}(\mathbf{y}_c)^\top \mathbf{s} + \dfrac{1}{2} \mathbf{s}^\top H(\mathbf{y}_c) \mathbf{s} \\ \text{s.t.} & \|\mathbf{s}\| \le \Delta_c, \end{cases} \tag{23}$$

where the neighborhood $\{\mathbf{y}_c + \mathbf{s} : \|\mathbf{s}\| \le \Delta_c\}$ is called a trust region at $\mathbf{y}_c$ with a trust region radius Δ_c. To obtain $\mathbf{y}_{c+1}$ quickly, the trust region subproblem (23) is usually solved inexactly.

If we restrict $\mathbf{s}$ along the negative gradient direction in the trust region, the optimal solution $\mathbf{s}_c^C$ is called the Cauchy point of (23). We pick an approximate solution $\mathbf{s}_c$ of (23) that is as good as the Cauchy point [1], i.e., the estimation of model reduction is at least

$$m_c(\mathbf{0}) - m_c(\mathbf{s}_c) \ge C_1 \left[m_c(\mathbf{0}) - m_c(\mathbf{s}_c^C) \right] \tag{24}$$

and the step size is at most

$$\|\mathbf{s}_c\| \le \min \left(C_2 \|\mathbf{s}_c^C\|, \Delta_c \right), \tag{25}$$

where $0 < C_1 \le 1 \le C_2$.

LEMMA 3.3. *Assume that $\mathbf{s}_c$ is an approximate solution of the trust region subproblem* (23) *which satisfies* (24) *and* (25). *Then, we have*

$$m_c(\mathbf{0}) - m_c(\mathbf{s}_c) \ge \frac{C_1}{2} \|\mathbf{g}(\mathbf{y}_c)\| \min \left(\Delta_c, \frac{\|\mathbf{g}(\mathbf{y}_c)\|}{\|H(\mathbf{y}_c)\|} \right) \tag{26}$$

and

$$m_c(\mathbf{0}) - m_c(\mathbf{s}_c) \ge \frac{C_1}{2C_2} \|\mathbf{g}(\mathbf{y}_c)\| \|\mathbf{s}_c\|. \tag{27}$$

Proof. By some calculations, we know the Cauchy point [54, 43]

$$\mathbf{s}_c^C = -\min\left(\Delta_c, \frac{\|\mathbf{g}(\mathbf{y}_c)\|^3}{\mathbf{g}(\mathbf{y}_c)^\top H(\mathbf{y}_c)\mathbf{g}(\mathbf{y}_c)}\right)\frac{\mathbf{g}(\mathbf{y}_c)}{\|\mathbf{g}(\mathbf{y}_c)\|},$$

which triggers a reduction

$$m_c(\mathbf{0}) - m_c(\mathbf{s}_c^C) \geq \frac{1}{2}\|\mathbf{g}(\mathbf{y}_c)\|\|\mathbf{s}_c^C\|.$$

Obviously, we have $\|\mathbf{s}_c^C\| \geq \min(\Delta_c, \|\mathbf{g}(\mathbf{y}_c)\|/\|H(\mathbf{y}_c)\|)$, and hence (26) holds by (24). Moreover, from (24) and (25), we get

$$\frac{m_c(\mathbf{0}) - m_c(\mathbf{s}_c)}{\|\mathbf{s}_c\|} \geq \frac{C_1}{C_2}\frac{m_c(\mathbf{0}) - m_c(\mathbf{s}_c^c)}{\|\mathbf{s}_c^C\|} \geq \frac{C_1}{2C_2}\|\mathbf{g}(\mathbf{y}_c)\|.$$

The proof is complete. ∎

To generate a trial point $\mathbf{y}_c^+$ on the unit sphere $\mathbb{S}^{n-2}$, we employ the Cayley transform [24, 13]. Let

$$W_c = \frac{1}{2}\left(\mathbf{y}_c\mathbf{s}_c^\top - \mathbf{s}_c\mathbf{y}_c^\top\right) \tag{28}$$

be a skew-symmetric matrix. Then $I + W_c$ is invertible. By the Cayley transform, we know that

$$O_c = (I + W_c)^{-1}(I - W_c) \tag{29}$$

is an orthogonal matrix. Hence, the trial point

$$\mathbf{y}_c^+ = O_c\mathbf{y}_c \tag{30}$$

stays still on the sphere $\mathbb{S}^{n-2}$ if $\mathbf{y}_c \in \mathbb{S}^{n-2}$. We remark here that the trial point $\mathbf{y}_c^+$ could be computed efficiently by about $4n$ multiplications, and matrices W_c and O_c are not required to form explicitly.

LEMMA 3.4 (see [32, 10]). *Suppose $\mathbf{y}_c \in \mathbb{S}^{n-2}$ and $\mathbf{s}_c \neq \mathbf{0}$. Using the Cayley transform* (28)–(30), *we obtain*

$$\mathbf{y}_c^+ = \frac{[(2 - \mathbf{s}_c^\top\mathbf{y}_c)^2 - \|\mathbf{s}_c\|^2]\mathbf{y}_c + 4\mathbf{s}_c}{4 + \|\mathbf{s}_c\|^2 - (\mathbf{s}_c^\top\mathbf{y}_c)^2} \tag{31}$$

and

$$\|\mathbf{y}_c^+ - \mathbf{y}_c\| = 2\left(\frac{\|\mathbf{s}_c\|^2 - (\mathbf{s}_c^\top\mathbf{y}_c)^2}{4 + \|\mathbf{s}_c\|^2 - (\mathbf{s}_c^\top\mathbf{y}_c)^2}\right)^{1/2}. \tag{32}$$

Whereafter, we compute the ratio of the actual reduction and the predicted reduction

$$\rho_c = \frac{f(\mathbf{y}_c) - f(\mathbf{y}_c^+)}{m_c(\mathbf{0}) - m_c(\mathbf{s}_c)}. \tag{33}$$

If this ratio is sufficiently positive, we accept the trial point as the next iterate. We would enlarge the trust region radius when the ratio is large enough. If the ratio is poor, we reduce the trust region radius for the next iteration. The detailed trust region algorithm is addressed in Algorithm 1.

Algorithm 1. A trust region algorithm.

Choose an initial iterate $\mathbf{y}_1 \in \mathbb{S}^{n-2}$, parameters $\Delta_1 \leq \overline{\Delta}$, $0 < \zeta \leq \eta_1 < \eta_2 < 1$, $0 < \gamma_1 < \gamma_2 < 1 < \gamma_3$, and $c = 1$.

while $\mathbf{g}(\mathbf{y}_c) \neq \mathbf{0}$ **do**

  Solve a trust region subproblem (23) and obtain $\mathbf{s}_c$ satisfying (24) and (25).

  Compute $\mathbf{y}_c^+$ and ρ_c by (31) and (33), respectively.

  Update iterate:

$$\mathbf{y}_{c+1} = \begin{cases} \mathbf{y}_c^+ & \text{if } \rho_c \geq \zeta, \\ \mathbf{y}_c & \text{otherwise.} \end{cases}$$

  Adjust trust region radius:

$$\Delta_{c+1} = \begin{cases} \min(\gamma_1 \Delta_c, \gamma_2 \|\mathbf{s}_c\|) & \text{if } \rho_c < \eta_1, \\ \Delta_c & \text{if } \rho_c \in [\eta_1, \eta_2), \\ \min(\gamma_3 \Delta_c, \overline{\Delta}) & \text{if } \rho_c \geq \eta_2. \end{cases}$$

  $c \leftarrow c + 1$.

end while

3.2. Computational complexity. We turn to analyze the computational cost in each iteration of Algorithm 1. First of all, it is unnecessary to store the n-by-$(n-1)$ matrix Q explicitly [24]. In practice, we only save the Householder vector $\mathbf{v} \in \mathbb{R}^n$ and a scalar $\beta \in \mathbb{R}$. For a given vector $\mathbf{y} \in \mathbb{R}^{n-1}$, we could calculate $\mathbf{x} = Q\mathbf{y}$ by

$$\mathbf{x} \leftarrow \begin{bmatrix} 0 \\ \mathbf{y} \end{bmatrix} \quad \text{and then} \quad \mathbf{x} \leftarrow \mathbf{x} - (\beta \mathbf{v}^\top \mathbf{x})\mathbf{v}. \tag{34}$$

For a given vector $\mathbf{x} \in \mathbb{R}^n$, we compute $\mathbf{y} = Q^\top \mathbf{x}$ using

$$\mathbf{x} \leftarrow \mathbf{x} - (\beta \mathbf{v}^\top \mathbf{x})\mathbf{v} \quad \text{and then} \quad \mathbf{y} \leftarrow \mathbf{x}(2:n). \tag{35}$$

The computational cost of $\mathbf{x} = Q\mathbf{y}$ or $\mathbf{y} = Q^\top \mathbf{x}$ is about $2n$ multiplications, which is cheap.

Let $\widetilde{\mathbf{u}}_{p,i} \equiv \sqrt[k]{\sigma w_p}\mathbf{u}_{p,i} \in \mathbb{R}^n$ for $i \in e_p$ and $e_p \in E$. Obviously, each vector $\widetilde{\mathbf{u}}_{p,i}$ has only k nonzero elements. From (17), we rewrite the compact Laplacian tensor as

$$\mathcal{L}_C = \sum_{e_p \in E} \sum_{i \in e_p} (Q^\top \widetilde{\mathbf{u}}_{p,i}) \circ \cdots \circ (Q^\top \widetilde{\mathbf{u}}_{p,i}).$$

At the initial step of Algorithm 1, we save a sparse matrix $U \in \mathbb{R}^{n \times mk}$ in which each column is $\widetilde{\mathbf{u}}_{p,i}$. Hence, the total number of nonzero elements of U is mk^2.

Given a vector $\mathbf{y} \in \mathbb{R}^{n-1}$, by a similar process used in (10), we have

$$\mathcal{L}_C \mathbf{y}^k = \sum_{e_p \in E} \sum_{i \in e_p} (\widetilde{\mathbf{u}}_{p,i}^\top Q\mathbf{y})^k = \text{sum}\left((U^\top (Q\mathbf{y})) \texttt{.\^{}k} \right),$$

where $(\cdot)$`.^k` means the kth componentwise power of a vector and sum$(\cdot)$ stands for the sum of all elements of a vector. By (34), it is about $2n$ multiplications to compute $Q\mathbf{y}$. For computing $U^\top(\cdot)$, we need mk^2 multiplications since U is sparse. Then, the operation $(\cdot)$`.^k` costs mk^2 multiplications at most. Hence, there are $2(n + mk^2)$ multiplications for computing $\mathcal{L}_C \mathbf{y}^k$ in total.

Using similar skills, we get

$$\begin{aligned}\mathcal{L}_C\mathbf{y}^{k-1} &= \sum_{e_p\in E}\sum_{i\in e_p}(\widetilde{\mathbf{u}}_{p,i}^\top Q\mathbf{y})^{k-1}(Q^\top\widetilde{\mathbf{u}}_{p,i}) \\ &= Q^\top\left(\sum_{e_p\in E}\sum_{i\in e_p}(\widetilde{\mathbf{u}}_{p,i}^\top Q\mathbf{y})^{k-1}\widetilde{\mathbf{u}}_{p,i}\right) \\ &= Q^\top\left(U[(U^\top Q\mathbf{y})\texttt{.\^{}(k-1)}]\right).\end{aligned}$$

Similar to the discussion for computing $(U^\top(Q\mathbf{y}))\texttt{.\^{}k}$, we need $2(n+mk^2)$ multiplications to calculate the vector $(U^\top Q\mathbf{y})\texttt{.\^{}(k-1)}$. Then, by the sparsity of U, costs for computing $U\cdot$ and $Q^\top(\cdot)$ are mk^2 and $2n$ multiplications, respectively. Hence, the total cost for $\mathcal{L}_C\mathbf{y}^{k-1}$ is about $4n+3mk^2$ multiplications.

To solve the trust region subproblem (23), we could employ iterative methods such as the conjugate gradient method [43, 54]. If so, the time-consuming computation in each iteration is $(\mathcal{L}_C\mathbf{y}^{k-2})\mathbf{s}$ for some vectors $\mathbf{s}\in\mathbb{R}^{n-1}$. By some calculations, we obtain

$$(\mathcal{L}_C\mathbf{y}^{k-2})\mathbf{s} = Q^\top\left(\sum_{e_p\in E}\sum_{i\in e_p}(\widetilde{\mathbf{u}}_{p,i}^\top Q\mathbf{y})^{k-2}(\widetilde{\mathbf{u}}_{p,i}^\top Q\mathbf{s})\widetilde{\mathbf{u}}_{p,i}\right).$$

Computational costs for $(\widetilde{\mathbf{u}}_{p,i}^\top Q\mathbf{y})$'s or $(\widetilde{\mathbf{u}}_{p,i}^\top Q\mathbf{s})$'s are $2n+mk^2$ multiplications. Their products require mk^2 multiplications. Owing to the sparsity of $\widetilde{\mathbf{u}}_{p,i}$'s, we get the sum in the bracket with mk^2 more multiplications. Multiplying $Q^\top$ needs $2n$ multiplications additionally. Therefore, the total cost for $(\mathcal{L}_C\mathbf{y}^{k-2})\mathbf{s}$ is about $6n+4mk^2$ multiplications.

Other operations in Algorithm 1 are about $\mathcal{O}(n)$ multiplications. Moreover, we note that $n\le mk$ for a uniform hypergraph. Hence, the total cost for one iteration of Algorithm 1 is about $\mathcal{O}(mk^2)$ in a word.

3.3. Convergence analysis. If Algorithm 1 terminates finitely, i.e., there exists an iteration c such that $\mathbf{g}(\mathbf{y}_c)=0$, then $f(\mathbf{y}_c)$ is a Z-eigenvalue of the compact Laplacian tensor $\mathcal{L}_C$ and the associated Z-eigenvector is $\mathbf{y}_c$, according to Lemma 3.2. Now, we consider the case that Algorithm 1 generates an infinite sequence of iterates. A weakly convergence theorem of Algorithm 1 is verifiable.

THEOREM 3.5. *Suppose that Algorithm* 1 *generates an infinite sequence of iterates* $\{\mathbf{y}_c\}$. *Then, we get*

$$\liminf_{c\to+\infty}\|\mathbf{g}(\mathbf{y}_c)\|=0.$$

Proof. See Appendix A. ∎

Theorem 3.5 implies that there exists a subsequence of iterates $\{\mathbf{y}_{c_j}\}$ such that $\mathbf{g}(\mathbf{y}_{c_j})\to 0$ as $j\to\infty$. Since the feasible region $\mathbb{S}^{n-2}$ is compact, $\{\mathbf{y}_{c_j}\}$ has at least one accumulation point $\mathbf{y}_*$. Owing to the continuity of $\mathbf{g}(\cdot)$, we have $\mathbf{g}(\mathbf{y}_*)=0$. That is to say, there exists a subsequence of iterates $\{\mathbf{y}_{c_j}\}$ which converges to a first-order stationary point $\mathbf{y}_*$.

THEOREM 3.6. *Let* $\{\mathbf{y}_c\}$ *be an infinite sequence of iterates generated by Algorithm* 1. *Then, we have*

$$\sum_{c=1}^{+\infty}\|\mathbf{y}_{c+1}-\mathbf{y}_c\|<+\infty.$$

Proof. See Appendix B. ∎

Theorem 3.6 means that the sequence of iterates generated by Algorithm 1 is a convergent Cauchy sequence. Hence, the total sequence of iterates $\{\mathbf{y}_c\}$ converges to a stationary point $\mathbf{y}_*$. Since $f(\cdot)$ is continuously differentiable, we have

$$\lim_{c\to\infty} f(\mathbf{y}_c) = f(\mathbf{y}_*) \qquad \text{and} \qquad \lim_{c\to\infty} \|\mathbf{g}(\mathbf{y}_c)\| = 0.$$

Hence, $f(\mathbf{y}_*)$ is a Z-eigenvalue of the compact Laplacian tensor $\mathcal{L}_C$ and $\mathbf{y}_*$ is the associated Z-eigenvector.

3.4. A probabilistic approach for the Fiedler vector. To obtain the Fiedler vector of an even-uniform hypergraph, we need to get the smallest Z-eigenvalue of the compact Laplacian tensor. For this purpose, we first start the trust region algorithm from plenty of random initial points sampled from the unit sphere uniformly. Then, we regard the minimum of resulting Z-eigenvalues as the smallest Z-eigenvalue of the compact Laplacian tensor, i.e., the algebraic connectivity of the hypergraph. Finally, the Fiedler vector of the hypergraph is easy to be calculated from the Z-eigenvector of the compact Laplacian tensor corresponding to the algebraic connectivity. The following theorem estimates the probability of obtaining the Fiedler vector of the hypergraph via this global strategy.

THEOREM 3.7. *Suppose that G is a given even-uniform hypergraph and $\mathcal{L}_C$ is its compact Laplacian tensor. We first start Algorithm 1 from N initial points that are sampled from $\mathbb{S}^{n-2}$ uniformly. Second, we pick up the resulting smallest Z-eigenvalue λ_* of $\mathcal{L}_C$ as well as the associated Z-eigenvector $\mathbf{y}_*$. Finally, we compute the estimated Fiedler vector of G by $\mathbf{x}_* = Q\mathbf{y}_*$. Then, this vector $\mathbf{x}_*$ is the true Fiedler vector of G with a probability of*

$$1 - (1-\varrho)^N, \tag{36}$$

where $\varrho \in (0, 1]$ is a constant. Therefore, if the number of samples N is large enough, we obtain the Fiedler vector of G with a high probability.

Proof. Let $\mathbf{y}^* \in \mathbb{S}^{n-2}$ be the Z-eigenvector of $\mathcal{L}_C$ corresponding to the smallest Z-eigenvalue. From Theorem B.1 and Lemma B.2, there is a neighborhood $\mathscr{B}(\mathbf{y}^*, r)$ such that a sequence of iterates $\{\mathbf{y}_c\}$ generated by Algorithm 1 converges to $\mathbf{y}^*$ if the random initial point $\mathbf{y}_1 \in \mathscr{B}(\mathbf{y}^*, r) \cap \mathbb{S}^{n-2}$. Since $\mathscr{B}(\mathbf{y}^*, r) \cap \mathbb{S}^{n-2}$ is nonempty and $\mathbb{S}^{n-2}$ is compact, there exists a constant $\varrho \in (0, 1]$ such that $\mathbf{y}_1$ happens to be sampled from $\mathscr{B}(\mathbf{y}^*, r) \cap \mathbb{S}^{n-2}$ with probability ϱ. If so, we obtain the Z-eigenvector $\mathbf{y}^*$ of $\mathcal{L}_C$ and the true Fiedler vector $\mathbf{x}^* = Q\mathbf{y}^*$ of G. In fact, we get the true Fiedler vector of G if $\{\mathbf{y}_c\} \cap \mathscr{B}(\mathbf{y}^*, r) \neq \emptyset$.

By the binomial distribution with parameters N and ϱ, we obtain the probabilistic estimation (36) straightforwardly. ∎

4. Numerical experiments. At the beginning, we report parameters for the trust region method proposed in Algorithm 1: $\zeta = 0.01, \eta_1 = 0.1, \eta_2 = 0.9, \gamma_1 = 0.25, \gamma_2 = 0.5, \gamma_3 = 2, \Delta_1 = 1$, and $\overline{\Delta} = 10$. To sample a random initial point $\mathbf{y}_1$ from $\mathbb{S}^{n-2}$ uniformly, we first generate a random vector whose components have a standard Gaussian distribution; then we normalize this vector and obtain $\mathbf{y}_1$. Algorithm 1 terminates if the gradient is tiny ($\|\mathbf{g}(\mathbf{y}_c)\| < 10^{-6}$) or if the number of iterations is large enough ($c > 1000$).

A conjugate gradient (CG) method [43, 54] is employed to solve the trust region subproblem (23). The first CG iteration generates exactly the Cauchy point $\mathbf{s}_c^C$. To

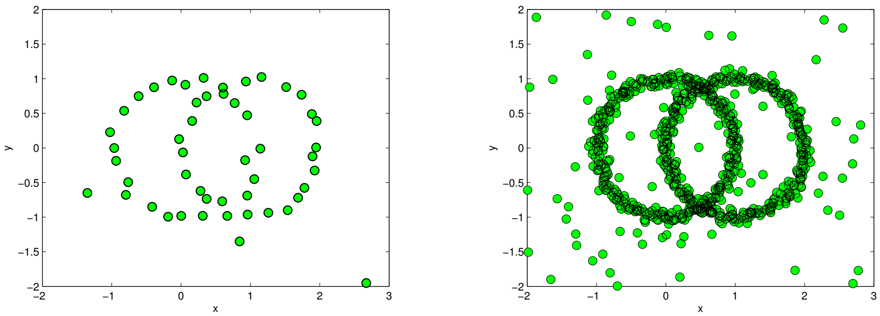


Fig. 4. *Synthetic points for circle partitioning with size $n = 50$ (left) and $n = 800$ (right).*

ensure (25), we may modify the trust region radius Δ_c to $10\|\mathbf{s}_c^C\|$ if the latter is smaller than Δ_c. Whereafter, CG iterations are going on until the CG path reaches the trust region boundary or the residual error is rather small: $\|H(\mathbf{y}_c)\mathbf{s}_c + \mathbf{g}(\mathbf{y}_c)\| < 0.01\|\mathbf{g}(\mathbf{y}_c)\|$.

In the following experiments, we focus on a fundamental problem in hypergraph partitioning, which is to divide vertices of a hypergraph into two groups. If several groups of vertices are divided, we could partition the resulting subhypergraphs and compute the corresponding Fiedler vectors hierarchically. In addition, we do not concern ourselves with the detection of outlier vertices in the process of hypergraph partitioning.

4.1. Circle partitioning. Circle partitioning is a spectral case of subspace clustering. In Figure 1, we try to estimate centers and radiuses of two crossing circles on which some points are known. The synthetic points are produced as follows. We set centers of true circles at $(0, 0)$ and $(1, 0)$. The common radius of two circles is one. Let $[\cdot]$ be the rounding function. For a given $n = |V|$, we choose $[n \times 45\%]$ points from each circle and then corrupt coordinates of these points with Gaussian white noise whose standard deviation is 0.05. The reminder $[n \times 10\%]$ points are outliers. Figure 4 illustrates typical locations of sampled points. These points serve as vertices of a k-uniform hypergraph, where $k \geq 4$ is even. We note that a complete hypergraph contains $\binom{n}{k}$ edges. This number maybe huge, e.g., $\binom{800}{6} \approx 3.57 \times 10^{14}$. Here, the complete hypergraph is approximated by a k-uniform random hypergraph which is generated by a three-step approach. First, we construct a complete graph which is connected. Second, for each edge $\{i, j\}$ of the complete graph, we add $k - 2$ random vertices which are sampled uniformly from vertex set $V \backslash \{i, j\}$. Finally, we repeat the second step $k - 1$ times and obtain $m = (k - 1)\binom{n}{2}$ edges. Hence, the resulting hypergraph is connected. In this way, a 6-uniform random hypergraph with 800 vertices is only equipped with 1,598,000 edges, which is significantly smaller than the number of edges of a complete hypergraph. So the resulting hypergraph is sparse and easy to manipulate. For each edge e_p of the hypergraph, we fit locations of points in e_p onto a circle and denote the fitting error as r_p. Let ς be the sample standard deviation of fitting errors $\{r_p\}_{p=1,\ldots,m}$. Then, the weight for an edge e_p is $w_p = \exp(-r_p/\varsigma)$.

For these hypergraphs, we test three kinds of partitioning methods.

- CE. Clique expansion of a hypergraph means that each edge of a hypergraph is expanded to a clique. Then, we partition the resulting graph using the normalized cut [3].

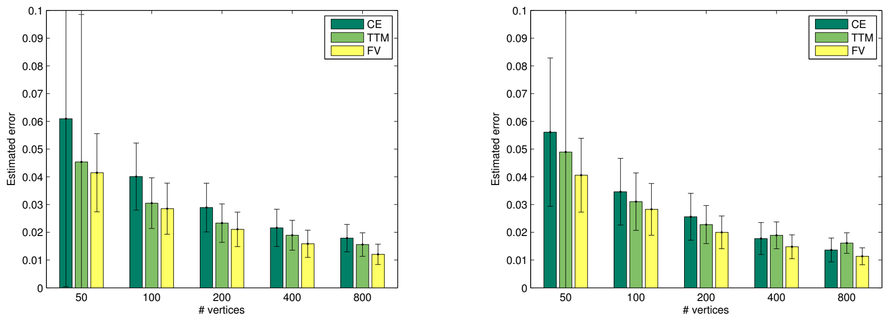


Fig. 5. *Estimated errors of two circles. Results for* 4*-uniform (left) and* 6*-uniform (right) hypergraphs.*

– TTM. The tensor trace maximization method based on the adjacency tensor of a hypergraph was studied in statistics [4, 23].
– FV. The new approach explores the Fiedler vector of an even-uniform hypergraph, which is a Z-eigenvector of the normalized Laplacian tensor corresponding to the second smallest Z-eigenvalue.

When the partition of vertices is obtained, we estimate two circles using points from these two clusters, respectively. For the appearance of outliers, we employ ℓ_1-norm fitting. The estimated error of a circle is defined as the distance between true and estimated centers plus the difference between true and estimated radiuses.

We test 4-uniform and 6-uniform hypergraphs and the number of vertices ranges from 50 to 800. For three partitioning methods, we run 100 tests in each case. The mean and standard deviation of estimated errors of two circles are reported in Figure 5. We see that all methods work well for a sparse hypergraph with some randomly selected edges. As the number of vertices increases, the estimated error of all methods decreases because more information on circles is available. Compared with the matrix-based method CE for 4-uniform hypergraphs, the Fiedler vector based on a normalized Laplacian tensor improves the estimated error by 29.8% on average.

4.2. Face clustering. The set of images of an object with a Lambertian reflectance function forms an illumination cone whose dimension is approximately three [5]. Regarding each image as a vertex, this phenomenon motivates us to establish 4-uniform hypergraphs. For every four images, the dissimilarity is defined as $s_4^2/(s_1^2+\cdots+s_4^2)$, where s_i is the ith singular value of a matrix whose columns are the vectorization of these images.

In this experiment, we consider the extended Yale face database B [22, 38]. 600 face images of 30 persons under 20 lighting conditions are chosen. To improve the efficiency of computation, we resize each image as 48×42 pixels. In one test, we select 40 face images of two persons with or without 10 outlier images. See Figure 6 for a typical selection. Hence, there are $n=40$ or $n=50$ vertices in a 4-uniform hypergraph. Then, the three-step approach introduced in circle partitioning is employed to construct random hypergraphs. Particularly, we equip a random hypergraph with more edges by repeating the second step $[\binom{n}{4}\cdot 10\%/\binom{n}{2}]$ times. In fact, we repeat the second step 12 and 19 times when $n=40$ and $n=50$, respectively. For four images in an edge e_p, we compute the dissimilarity d_p. Let ς be the sample standard deviation of $\{d_p\}$. Then, the weight of an edge e_p is $w_p=\exp(-d_p/\varsigma)$. In this way, we produce a 4-uniform hypergraph.

Fig. 6. *Sampled images from the extended Yale face dataset B* [22, 38]. *Face images of two persons (lines* 1–4*) and* 10 *outliers (line* 5*).*

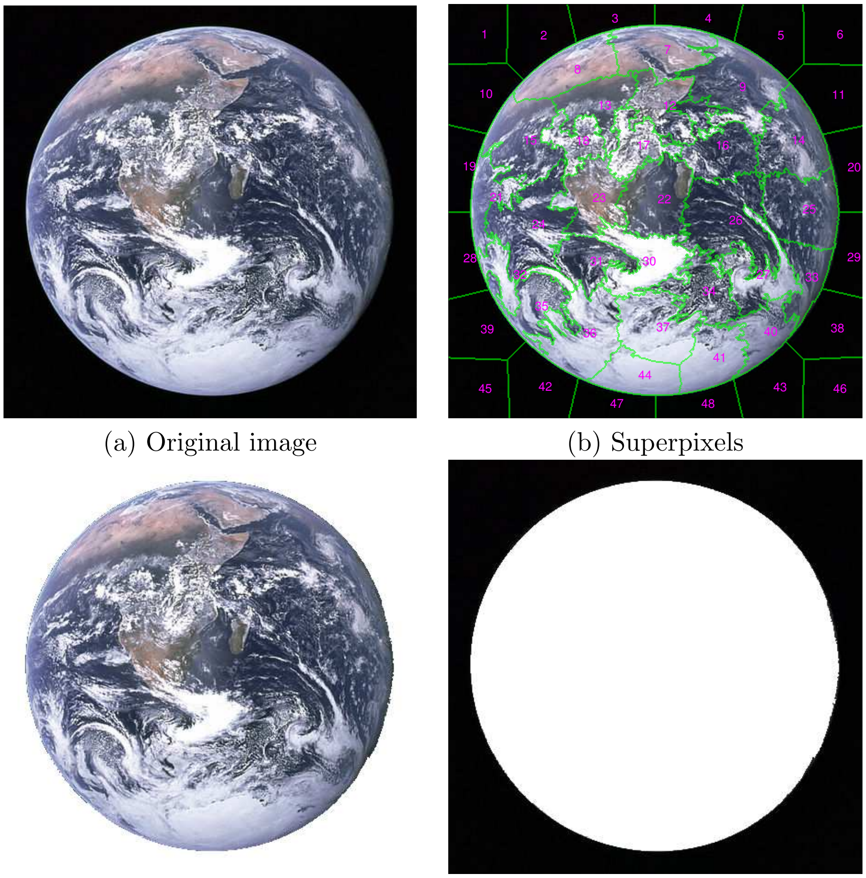
Table 2
Results for face clustering (mean ± standard deviation).

# outliers	GT	TD	TTM	FV
0	0.380 ± 0.081	0.203 ± 0.100	0.098 ± 0.190	0.002 ± 0.007
10	0.402 ± 0.071	0.214 ± 0.138	0.028 ± 0.109	0.009 ± 0.018

Our target is to divide 40 images of two persons into two groups such that images of one person under different lighting conditions are collected in one group. We do not care about outlier images that could be in any group. To divide vertices of these hypergraphs, we compare four kinds of methods for hypergraph partitioning. First, a game-theoretic (GT) method [51] modeled the hypergraph clustering problem in terms of a noncooperative multiplayer clustering game. Second, a tensor decomposition (TD) approach was applied for the adjacency tensor of a uniform hypergraph [25, 11]. The other two approaches are TTM and our FV. For each method, we run 100 tests and report results on mean and standard deviation of partition errors of two persons in Table 2. Compared with GT and TD, we find that FV and TTM work well for a hypergraph with a small quantity of edges. To partition a 4-uniform hypergraph generated from face clustering, the Fiedler vector heuristics based on a normalized Laplacian tensor performs better.

4.3. Image segmentation. The Fiedler vector of a hypergraph could be applied for image segmentation. For example, we consider the earth image[1] shown in Figure 7(a) and try to extract the earth image from a dark background. Using the technique of superpixels [49], we divide the original image into dozens of superpixels in advance. Here, we employ the SLIC technique [2] and obtain $n = 48$ superpixels which are reported in Figure 7(b). Then, we construct a k-uniform hypergraph, whose vertices are these superpixels. The three-step approach introduced in circle partitioning is applied for generating a random hypergraph, where the second step is repeated $(k-1)^2$

[1]Download from https://en.wikipedia.org/wiki/Earth.



(a) Original image (b) Superpixels

(c) Earth (d) Dark

Fig. 7. *The earth image is segmented by Fiedler vectors of even-uniform hypergraphs.*

times. Hence, the resulting k-uniform hypergraph has $m = (k-1)^2\binom{n}{2}$ edges. If $k = 2$, we obtain a complete graph. In each edge e_p, the weight w_p is proportional to the similarity of color distributions of superpixels color_p and is inversely proportional to distance among superpixels dist_p. Given an image, we first transform the image from the RGB color space to the HSV color space since the latter is more perceptually relevant. Second, we compute the HSV color distribution hsv_i for superpixel $i = 1, \ldots, n$. Here, HSV stands for hue, saturation, and value. The domain of hue is divided into 12 intervals. Domains of saturation and value are divided into four intervals, respectively. That is to say, the HSV color space is divided into 192 areas. Since a superpixel may contain plenty of pixels, we count the number of pixels in these areas and then normalize to obtain the HSV color distribution $\mathrm{hsv}_i \in \mathbb{R}^{192}_+$. The similarity of color distributions of superpixels in an edge $e_p = \{i_1, i_2, \ldots, i_k\}$ is defined as

$$\mathrm{color}_p = \frac{\mathrm{hsv}_{i_1}^\top(\mathrm{hsv}_{i_2} * \cdots * \mathrm{hsv}_{i_k})}{\|\mathrm{hsv}_{i_1}\|_k \cdots \|\mathrm{hsv}_{i_k}\|_k},$$

where $*$ is the componentwise Hadamard product. Third, we find centers of superpixels cent_i for $i = 1, \ldots, n$. Let $\overline{\mathrm{cent}}_p = \frac{1}{k}\sum_{j \in e_p} \mathrm{cent}_j$ be an imaginary center of e_p.

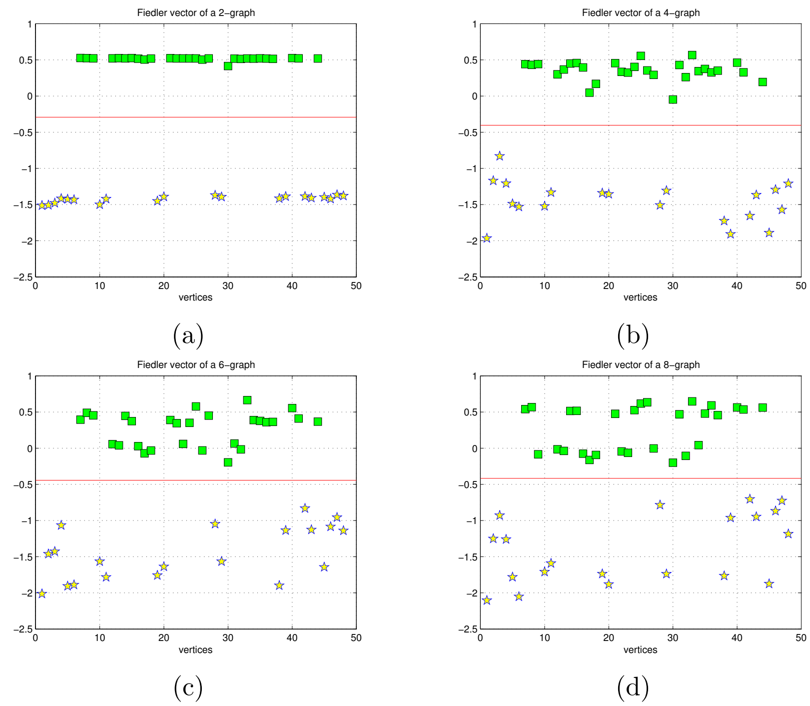


FIG. 8. *Element values of Fiedler vectors of k-uniform hypergraphs.*

The star distance among superpixels in an edge e_p is

$$\mathrm{dist}_p = \sum_{i \in e_p} (\mathrm{cent}_i - \overline{\mathrm{cent}}_p)^k.$$

For even $k = 2$, 4, 6, and 8, Fiedler vectors of k-uniform hypergraphs generate the same partition of the earth image. See Figures 7(c) and (d).

If $k = 2$, the hypergraph reduces to a complete graph whose Fiedler vector is illustrated in Figure 8(a). Components of the Fiedler vector of a graph enjoy two clusters 0.5 and -1.4. Fiedler vectors of 4-, 6-, and 8-uniform hypergraphs are reported in Figures 8(b), (c), and (d), respectively. It is easy to see that with the increase of even k, the distribution of components of the Fiedler vector of the k-uniform hypergraph becomes loose and unconsolidated. This phenomenon is consistent with Theorem 2.8.

5. Conclusion. The Fiedler vector of an even-uniform hypergraph has been proposed and applied for subspace partitioning and face clustering. For an even-uniform hypergraph, the Fiedler vector is the Z-eigenvector corresponding to the second smallest Z-eigenvalue of a normalized Laplacian tensor. When the order of vertices of the hypergraph is permuted, components of the Fiedler vector change consistently. The relationship between the Fiedler vector and the Cheeger constant of the hypergraph was also addressed.

A trust region algorithm with a global strategy was proposed to compute the Fiedler vector of a k-uniform hypergraph, where k is even. The computational complexity of the proposed algorithm is about $\mathcal{O}(|E|k^2)$ multiplications in each iteration. So it is cheap. Theoretically, the probability of obtaining the Fiedler vector of the hypergraph was analyzed, and it is close to one when random initial points on the unit sphere are dense enough. Generally speaking, our Fiedler vector heuristics for partitioning an even-uniform hypergraph is a generalization of the normalized cut for a graph and contains more advantages in applications. The theoretical rationality of the partitions of a hypergraph induced by the tensor Fiedler vector are natural connected clusters. Altogether, the Fiedler vector of an even-uniform hypergraph enriches spectral hypergraph theory.

For partitioning a uniform hypergraph, the design of the cost function, such as the ones presented in (5) and (7), is delicate and crucial. How to design a better cost function for hypergraph partitioning and establish the (equivalence) relationship between the cost function value and the number of cut edges are our further research works.

Appendix A. Proof of Theorem 3.5. Since the objective function of the spherical optimization (20) is twice continuously differentiable and the feasible region is compact, we get the following lemma.

Lemma A.1 (Lemma 3 in [13]). *Suppose that* $f(\mathbf{y})$, $\mathbf{g}(\mathbf{y})$, *and* $H(\mathbf{y})$ *are defined by* (20), (21), *and* (22), *respectively. Then, there exists a constant* $M \geq 1$ *such that*

$$|f(\mathbf{y})| \leq M, \quad \|\mathbf{g}(\mathbf{y})\| \leq M, \quad \text{and} \quad \|H(\mathbf{y})\| \leq M$$

for $\mathbf{y} \in \mathbb{S}^{n-2}$.

The next lemma means that if the gradient is bounded away from zero, the trust region radius cannot tend to zero and we will find a new iterate soon.

Lemma A.2. *Suppose* $\mathbf{g}(\mathbf{y}_c) \neq \mathbf{0}$. *If*

$$\Delta_c \leq \frac{(1-\eta_2)C_1\|\mathbf{g}(\mathbf{y}_c)\|}{M(2+\overline{\Delta}/2)},$$

we have

$$\Delta_{c+1} \geq \Delta_c.$$

Proof. Since $(1-\eta_2)C_1 < 1 < (2+\overline{\Delta}/2)$ and by Lemma A.1, we have $\Delta_c \leq \frac{\|\mathbf{g}(\mathbf{y}_c)\|}{M} \leq \frac{\|\mathbf{g}(\mathbf{y}_c)\|}{\|H(\mathbf{y}_c)\|}$. Hence, by (26), we know

$$m_c(\mathbf{0}) - m_c(\mathbf{s}_c) \geq \frac{C_1}{2}\|\mathbf{g}(\mathbf{y}_c)\|\Delta_c. \tag{37}$$

Because the objective function $f(\mathbf{x})$ is zero-order homogeneous, we get

$$\mathbf{y}^\top \mathbf{g}(\mathbf{y}) = 0. \tag{38}$$

Since $\|\mathbf{y}_c\| = 1$, we have $|\mathbf{s}_c^\top \mathbf{y}_c| \leq \|\mathbf{s}_c\|$. Moreover, $\|\mathbf{s}_c\| \leq \Delta_c \leq \overline{\Delta}$. From (31), (38), and Lemma A.1, we get

$$\begin{aligned}\left|\mathbf{g}(\mathbf{y}_c)^\top(\mathbf{y}_c^+ - \mathbf{y}_c - \mathbf{s}_c)\right| &= \left|\frac{4\mathbf{g}(\mathbf{y}_c)^\top \mathbf{s}_c}{4+\|\mathbf{s}_c\|^2 - (\mathbf{s}_c^\top \mathbf{y}_c)^2} - \mathbf{g}(\mathbf{y}_c)^\top \mathbf{s}_c\right| \\ &= \frac{|\mathbf{g}(\mathbf{y}_c)^\top \mathbf{s}_c|}{4+\|\mathbf{s}_c\|^2 - (\mathbf{s}_c^\top \mathbf{y}_c)^2}(\|\mathbf{s}_c\|^2 - (\mathbf{s}_c^\top \mathbf{y}_c)^2) \\ &\leq \frac{M\overline{\Delta}}{4}\Delta_c^2. \end{aligned} \tag{39}$$

By the mean value theorem, we have

$$f(\mathbf{y}_c^+) = f(\mathbf{y}_c) + \mathbf{g}(\mathbf{y}_c)^\top(\mathbf{y}_c^+ - \mathbf{y}_c) + \frac{1}{2}(\mathbf{y}_c^+ - \mathbf{y}_c)^\top H(\widetilde{\mathbf{y}})(\mathbf{y}_c^+ - \mathbf{y}_c), \tag{40}$$

where $\widetilde{\mathbf{y}} = \theta\mathbf{y}_c^+ + (1-\theta)\mathbf{y}_c$ and $\theta \in (0,1)$. From (32), we know

$$\|\mathbf{y}_c^+ - \mathbf{y}_c\| \leq \|\mathbf{s}_c\|. \tag{41}$$

Moreover, by (33), (23), (37), (39), (40), and (41), we obtain

$$\begin{aligned}
|1-\rho_c| &= \left| \frac{f(\mathbf{y}_c^+) - f(\mathbf{y}_c) - \mathbf{g}(\mathbf{y}_c)^\top\mathbf{s}_c - \frac{1}{2}\mathbf{s}_c^\top H(\mathbf{y}_c)\mathbf{s}_s}{m_c(\mathbf{0}) - m_c(\mathbf{s}_c)} \right| \\
&= \left| \frac{\mathbf{g}(\mathbf{y}_c)^\top(\mathbf{y}_c^+ - \mathbf{y}_c - \mathbf{s}_c) + \frac{1}{2}(\mathbf{y}_c^+ - \mathbf{y}_c)^\top H(\widetilde{\mathbf{y}})(\mathbf{y}_c^+ - \mathbf{y}_c) - \frac{1}{2}\mathbf{s}_c^\top H(\mathbf{y}_c)\mathbf{s}_s}{m_c(\mathbf{0}) - m_c(\mathbf{s}_c)} \right| \\
&\leq \frac{|\mathbf{g}(\mathbf{y}_c)^\top(\mathbf{y}_c^+ - \mathbf{y}_c - \mathbf{s}_c)| + M\|\mathbf{s}_c\|^2}{m_c(\mathbf{0}) - m_c(\mathbf{s}_c)} \\
&\leq \frac{\Delta_c^2 M\overline{\Delta}/4 + \Delta_c^2 M}{C_1\|\mathbf{g}(\mathbf{y}_c)\|\Delta_c/2} \\
&\leq 1-\eta_2.
\end{aligned}$$

Therefore, we have $\rho_c \geq \eta_2$ and hence $\Delta_{c+1} \geq \Delta_c$. ∎

Now, we are ready to prove the weakly convergence theorem for Algorithm 1.

Proof of Theorem 3.5. We proceed by contradiction. Assume that there exists a small constant ϵ such that

$$\|\mathbf{g}(\mathbf{y})\| \geq \epsilon > 0 \qquad \text{for } c = 1,2,\ldots. \tag{42}$$

Let $\mathcal{I} \equiv \{c \in \mathcal{N}_+ : \rho_c \geq \zeta\}$ be an index set of successful iterations. Then, $\mathcal{I}$ contains infinite iterations. Otherwise, let $\overline{c}$ be the largest index in $\mathcal{I}$, which means that $\rho_{\overline{c}+j} < \zeta \leq \eta_1$ for $j = 1,2,\ldots$. Hence, we have $\Delta_{\overline{c}+j+1} \leq \gamma_2\Delta_{\overline{c}+j} \leq \gamma_2^j\Delta_{\overline{c}+1}$ for all $j = 1,2,\ldots$. Therefore, $\Delta_{\overline{c}+j} \to 0$ as $j \to \infty$. Because of (42) and Lemma A.2, $\Delta_{\overline{c}+j+1}$ does not decrease if $\Delta_{\overline{c}+j} \leq \frac{(1-\eta_2)C_1\epsilon}{M(2+\overline{\Delta}/2)}$. This leads to a contradiction.

Since the objective function $f(\mathbf{y})$ has a lower bound $\underline{f}$, according to (33), (26), (42), and Lemma A.1, we obtain

$$\begin{aligned}
f(\mathbf{y}_1) - \underline{f} &\geq \sum_{c\in\mathcal{I}} f(\mathbf{y}_c) - f(\mathbf{y}_{c+1}) \\
&\geq \zeta \sum_{c\in\mathcal{I}} m_c(\mathbf{0}) - m_c(\mathbf{s}_c) \\
&\geq \frac{\zeta C_1}{2}\sum_{c\in\mathcal{I}} \|\mathbf{g}(\mathbf{y}_c)\| \min\left(\Delta_c, \frac{\|\mathbf{g}(\mathbf{y}_c)\|}{\|H(\mathbf{y}_c)\|}\right) \\
&\geq \frac{\zeta C_1}{2}\sum_{c\in\mathcal{I}} \epsilon \min\left(\Delta_c, \frac{\epsilon}{M}\right).
\end{aligned}$$

Hence, we have $\Delta_c \to 0$ as $c \to \infty$ and $c \in \mathcal{I}$. However, by (42) and Lemma A.2, we get that Δ_c cannot tend to zero, which lead to a contradiction. Therefore, Theorem 3.5 is valid. ∎

Appendix B. Proof of Theorem 3.6. Due to the multilinearity of tensors, the objective function $f(\mathbf{y})$ is semialgebraic [6] since its graph

$$\mathrm{Graph} f(\mathbf{y}) \equiv \{(\mathbf{y}, t) : \mathcal{L}_C \mathbf{y}^k = t(\mathbf{y}^\top \mathbf{y})^{k/2}\}$$

is semialgebraic. Hence, $f(\cdot)$ enjoys the following Kurdyka–Łojasiewicz (KL) property.

THEOREM B.1 (KL property [6, 1]). *Suppose that* $\mathbf{y}_*$ *is a stationary point of the objective function* $f(\mathbf{y})$. *Then, there exist a neighborhood* $\mathscr{X}(\mathbf{y}_*)$, *an exponent* $\theta \in [0, 1)$, *and a positive constant* C_{KL} *such that for all* $\mathbf{y} \in \mathscr{X}(\mathbf{y}_*)$, *the following inequality holds:*

$$|f(\mathbf{y}) - f(\mathbf{y}_*)|^\theta \le C_{KL} \|\mathbf{g}(\mathbf{y})\|. \tag{43}$$

Here, we define $0^0 \equiv 0$.

LEMMA B.2. *Let* $\mathscr{B}(\mathbf{y}_*, r) \equiv \{\mathbf{y} \in \mathbb{R}^{n-1} : \|\mathbf{y} - \mathbf{y}_*\| < r\} \subseteq \mathscr{X}(\mathbf{y}_*)$ *be a neighborhood of* $\mathbf{y}_*$. *Suppose that* $\mathbf{y}_1 \in \mathbb{S}^{n-2}$ *is an initial point satisfying*

$$r > \rho(\mathbf{y}_1) \equiv \frac{2C_2 C_{KL}}{\zeta C_1 (1-\theta)} |f(\mathbf{y}_1) - f(\mathbf{y}_*)|^{1-\theta} + \|\mathbf{y}_1 - \mathbf{y}_*\|.$$

Then, the following two assertions hold:

$$\mathbf{y}_c \in \mathscr{B}(\mathbf{y}_*, r) \qquad \text{for } c = 1, 2, \dots \tag{44}$$

and

$$\sum_{c=1}^{\infty} \|\mathbf{y}_{c+1} - \mathbf{y}_c\| \le \frac{2C_2 C_{KL}}{\zeta C_1 (1-\theta)} |f(\mathbf{y}_1) - f(\mathbf{y}_*)|^{1-\theta}. \tag{45}$$

Proof. We proceed by induction. Obviously, we have $\mathbf{y}_1 \in \mathscr{B}(\mathbf{y}_*, r)$.

Now, we assume that $\mathbf{y}_i \in \mathscr{B}(\mathbf{y}_*, r)$ for $i = 1, \dots, c$. Hence, the KL property holds in these points. Let

$$\phi(t) \equiv \frac{C_{KL}}{1-\theta} |t - f(\mathbf{y}_*)|^{1-\theta}.$$

Then, $\phi(t)$ is a concave function for $t > f(\mathbf{y}_*)$. Therefore, for successful iterations $i \in \{1, \dots, c\} \cap \mathcal{I}$, we have

$$\begin{aligned}
\phi(f(\mathbf{y}_i)) - \phi(f(\mathbf{y}_{i+1})) &\ge \phi'(f(\mathbf{y}_i))(f(\mathbf{y}_i) - f(\mathbf{y}_{i+1})) \\
&= C_{KL} |f(\mathbf{y}_i) - f(\mathbf{y}_*)|^{-\theta} (f(\mathbf{y}_i) - f(\mathbf{y}_{i+1})) \\
[\text{KL property}] &\ge \frac{f(\mathbf{y}_i) - f(\mathbf{y}_{i+1})}{\|\mathbf{g}(\mathbf{y}_i)\|} \\
&\ge \frac{\zeta(m_i(\mathbf{0}) - m_i(\mathbf{s}_i))}{\|\mathbf{g}(\mathbf{y}_i)\|} \\
[\text{for (27)}] &\ge \frac{\zeta C_1}{2C_2} \|\mathbf{s}_i\| \\
[\text{for (41)}] &\ge \frac{\zeta C_1}{2C_2} \|\mathbf{y}_{i+1} - \mathbf{y}_i\|.
\end{aligned}$$

On the other hand, for $i \in \{1, \dots, c\} \backslash \mathcal{I}$, we know $\mathbf{y}_{i+1} = \mathbf{y}_i$. Therefore, we obtain

$$\|\mathbf{y}_{i+1} - \mathbf{y}_i\| \le \frac{2C_2}{\zeta C_1} [\phi(f(\mathbf{y}_i)) - \phi(f(\mathbf{y}_{i+1}))] \qquad \text{for } i = 1, \dots, c.$$

Then,

$$\begin{aligned}
\|\mathbf{y}_{c+1} - \mathbf{y}_*\| &\le \sum_{i=1}^{c} \|\mathbf{y}_{i+1} - \mathbf{y}_i\| + \|\mathbf{y}_1 - \mathbf{y}_*\| \\
&\le \frac{2C_2}{\zeta C_1} \sum_{i=1}^{c} \phi(f(\mathbf{y}_i)) - \phi(f(\mathbf{y}_{i+1})) + \|\mathbf{y}_1 - \mathbf{y}_*\| \\
&\le \frac{2C_2}{\zeta C_1} \phi(f(\mathbf{y}_1)) + \|\mathbf{y}_1 - \mathbf{y}_*\| \\
&= \rho(\mathbf{y}_1) < r.
\end{aligned}$$

The above inequality means $\mathbf{y}_{c+1} \in \mathscr{B}(\mathbf{y}_*, r)$, and hence (44) holds. Moreover,

$$\sum_{c=1}^{\infty} \|\mathbf{y}_{c+1} - \mathbf{y}_c\| \le \frac{2C_2}{\zeta C_1} \sum_{c=1}^{\infty} \phi(f(\mathbf{y}_c)) - \phi(f(\mathbf{y}_{c+1})) \le \frac{2C_2}{\zeta C_1} \phi(f(\mathbf{y}_1)).$$

The inequality (45) also holds. ∎

Proof of Theorem 3.6. From Theorem 3.5 and the compactness of $\mathbb{S}^{n-2}$, there exists a subsequence of iterates $\{\mathbf{y}_{c_j}\}$ that converges to a stationary point $\mathbf{y}_* \in \mathbb{S}^{n-2}$.

For this $\mathbf{y}_*$, there exists a neighborhood $\mathscr{X}(\mathbf{y}_*)$ such that the KL inequality (43) holds. Since $\mathbf{y}_{c_j} \to \mathbf{y}_*$ as $j \to \infty$, there exists an index J such that $\rho(\mathbf{y}_{c_J}) < r$, where $\mathscr{B}(\mathbf{y}_*, r) \subseteq \mathscr{X}(\mathbf{y}_*)$. Then, by Lemma B.2, we have

$$\sum_{c=1}^{+\infty} \|\mathbf{y}_{c+1} - \mathbf{y}_c\| \le \sum_{c=1}^{c_J - 1} \|\mathbf{y}_{c+1} - \mathbf{y}_c\| + \frac{2C_2 C_{KL}}{\zeta C_1 (1-\theta)} |f(\mathbf{y}_{c_J}) - f(\mathbf{y}_*)|^{1-\theta} < +\infty.$$

Hence, $\{\mathbf{y}_c\}$ is a Cauchy sequence and converges to the stationary point $\mathbf{y}_*$. ∎

Acknowledgments. The authors are grateful to the associate editor and the anonymous referees for their comments which helped us to improve our manuscript essentially.

REFERENCES

[1] P.-A. Absil, R. Mahony, and B. Andrews, *Convergence of the iterates of descent methods for analytic cost functions*, SIAM J. Optim., 16 (2005), pp. 531–547, https://doi.org/10.1137/040605266.

[2] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, *SLIC superpixels compared to state-of-the-art superpixel methods*, IEEE Trans. Pattern Anal. Mach. Intell., 34 (2012), pp. 2274–2281.

[3] S. Agarwal, K. Branson, and S. Belongie, *Higher order learning with graphs*, in Proceedings of the 23rd International ACM Conference on Machine Learning, 2006, pp. 17–24.

[4] E. Arias-Castro, G. Chen, and G. Lerman, *Spectral clustering based on local linear approximations*, Electron. J. Stat., 5 (2011), pp. 1537–1587.

[5] P. N. Belhumeur and D. J. Kriegman, *What is the set of images of an object under all possible illumination conditions?*, Int. J. Comput. Vis., 28 (1998), pp. 245–260.

[6] J. Bolte, A. Daniilidis, and A. Lewis, *The Lojasiewicz inequality for nonsmooth subanalytic functions with applications to subgradient dynamical systems*, SIAM J. Optim., 17 (2007), pp. 1205–1223, https://doi.org/10.1137/050644641.

[7] C. Bu, Y. Fan, and J. Zhou, *Laplacian and signless Laplacian Z-eigenvalues of uniform hypergraphs*, Front. Math. China, 11 (2016), pp. 511–520.

[8] Ü. V. Çatalyürek and C. Aykanat, *Hypergraph-partitioning-based decomposition for parallel sparse-matrix vector multiplication*, IEEE Trans. Parallel Distrib. Systems, 10 (1999), pp. 673–693.

[9] Ü. V. Çatalyürek, C. Aykanat, and B. Uçar, *On two-dimensional sparse matrix partitioning: Models, methods, and a recipe*, SIAM J. Sci. Comput., 32 (2010), pp. 656–683, https://doi.org/10.1137/080737770.

[10] J. Chang, Y. Chen, and L. Qi, *Computing eigenvalues of large scale sparse tensors arising from a hypergraph*, SIAM J. Sci. Comput., 38 (2016), pp. A3618–A3643, https://doi.org/10.1137/16M1060224.

[11] G. Chen and G. Lerman, *Spectral curvature clustering (SCC)*, Int. J. Comput. Vis., 81 (2009), pp. 317–330.

[12] L. Chen, L. Han, and L. Zhou, *Computing tensor eigenvalues via homotopy methods*, SIAM J. Matrix Anal. Appl., 37 (2016), pp. 290–319, https://doi.org/10.1137/15M1010725.

[13] Y. Chen, L. Qi, and Q. Wang, *Computing extreme eigenvalues of large scale Hankel tensors*, J. Sci. Comput., 68 (2016), pp. 716–738.

[14] F. Chung, *The Laplacian of a hypergraph*, in Expanding Graphs, DIMACS Ser. Discrete Math. Theoret. Comput. Sci. 10, AMS, Providence, RI, 1993, pp. 21–36.

[15] F. Chung, *Spectral Graph Theory*, AMS, Providence, RI, 1997.

[16] F. Chung and L. Lu, *Complex Graphs and Networks*, AMS, Providence, RI, 2006.

[17] J. Cooper and A. Dutle, *Spectra of uniform hypergraphs*, Linear Algebra Appl., 436 (2012), pp. 3268–3292.

[18] C.-F. Cui, Y.-H. Dai, and J. Nie, *All real eigenvalues of symmetric tensors*, SIAM J. Matrix Anal. Appl., 35 (2014), pp. 1582–1601, https://doi.org/10.1137/140962292.

[19] O. Duchenne, F. Bach, I.-S. Kweon, and J. Ponce, *A tensor-based algorithm for high-order graph matching*, IEEE Trans. Pattern Anal. Mach. Intell., 33 (2011), pp. 2383–2395.

[20] M. Fiedler, *Algebraic connectivity of graphs*, Czechoslovak Math. J., 23 (1973), pp. 298–305.

[21] M. Fiedler, *A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory*, Czechoslovak Math. J., 25 (1975), pp. 619–633.

[22] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman, *From few to many: Illumination cone models for face recognition under variable lighting and pose*, IEEE Trans. Pattern Anal. Mach. Intell., 23 (2001), pp. 643–660.

[23] D. Ghoshdastidar and A. Dukkipati, *Uniform hypergraph partitioning: Provable tensor methods and sampling techniques*, J. Mach. Learn. Res., 18 (2017), pp. 1–41.

[24] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 4th ed., The Johns Hopkins University Press, Baltimore, MD, 2013.

[25] V. M. Govindu, *A tensor decomposition for geometric grouping and segmentation*, in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), Vol. 1, 2005, pp. 1150–1157.

[26] L. Grigori, E. G. Boman, S. Donfack, and T. A. Davis, *Hypergraph-based unsymmetric nested dissection ordering for sparse LU factorization*, SIAM J. Sci. Comput., 32 (2010), pp. 3426–3446, https://doi.org/10.1137/080720395.

[27] D. J. Higham, G. Kalna, and M. Kibble, *Spectral clustering and its use in bioinformatics*, J. Comput. Appl. Math., 204 (2007), pp. 25–37.

[28] S. Hu and L. Qi, *Algebraic connectivity of an even uniform hypergraph*, J. Comb. Optim., 24 (2012), pp. 564–579.

[29] S. Hu and L. Qi, *The Laplacian of a uniform hypergraph*, J. Comb. Optim., 29 (2015), pp. 331–366.

[30] Y. Huang, Q. Liu, F. Lv, Y. Gong, and D. N. Metaxas, *Unsupervised image categorization by hypergraph partition*, IEEE Trans. Pattern Anal. Mach. Intell., 33 (2011), pp. 1266–1273.

[31] E. Ihler, D. Wagner, and F. Wagner, *Modeling hypergraphs by graphs with the same mincut properties*, Inform. Process. Lett., 45 (1993), pp. 171–175.

[32] B. Jiang and Y.-H. Dai, *A framework of constraint preserving update schemes for optimization on Stiefel manifold*, Math. Program., 153 (2015), pp. 535–575.

[33] S. Joly and G. Le Calvé, *Three-way distances*, J. Classification, 12 (1995), pp. 191–205.

[34] L. Kang, V. Nikiforov, and X. Yuan, *The p-spectral radius of k-partite and k-chromatic uniform hypergraphs*, Linear Algebra Appl., 478 (2015), pp. 81–107.

[35] G. Karypis, R. Aggarwal, V. Kumar, and S. Shekhar, *Multilevel hypergraph partitioning: Applications in VLSI domain*, IEEE Trans. Very Large Scale Integr. (VLSI) Syst., 7 (1999), pp. 69–79.

[36] E. Kayaaslan, A. Pinar, Ü. Çatalyürek, and C. Aykanat, *Partitioning hypergraphs in scientific computing applications through vertex separators on graphs*, SIAM J. Sci. Comput., 34 (2012), pp. A970–A992, https://doi.org/10.1137/100810022.

[37] P. Keevash, J. Lenz, and D. Mubayi, *Spectral extremal problems for hypergraphs*, SIAM J. Discrete Math., 28 (2014), pp. 1838–1854, https://doi.org/10.1137/130929370.

[38] K. C. Lee, J. Ho, and D. J. Kriegman, *Acquiring linear subspaces for face recognition under*

variable lighting, IEEE Trans. Pattern Anal. Mach. Intell., 27 (2005), pp. 684–698.

[39] G. Li, L. Qi, and G. Yu, *The Z-eigenvalues of a symmetric tensor and its application to spectral hypergraph theory*, Numer. Linear Algebra Appl., 20 (2013), pp. 1001–1029.

[40] L.-H. Lim, *Singular values and eigenvalues of tensors: A variational approach*, in Proceedings of the 1st IEEE International Workshop on Computational Advances in Multi-sensor Adaptive Processing, 2005, pp. 129–132.

[41] O. E. Livne and A. Brandt, *Lean algebraic multigrid (LAMG): Fast graph Laplacian linear solver*, SIAM J. Sci. Comput., 34 (2012), pp. B499–B522, https://doi.org/10.1137/110843563.

[42] E. Merkurjev, T. Kostić, and A. L. Bertozzi, *An MBO scheme on graphs for classification and image processing*, SIAM J. Imaging Sci., 6 (2013), pp. 1903–1930, https://doi.org/10.1137/120886935.

[43] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer, New York, 2006.

[44] A. Pothen, H. D. Simon, and K.-P. Liou, *Partitioning sparse matrices with eigenvectors of graphs*, SIAM J. Matrix Anal. Appl., 11 (1990), pp. 430–452, https://doi.org/10.1137/0611030.

[45] L. Qi, *Eigenvalues of a real supersymmetric tensor*, J. Symbolic Comput., 40 (2005), pp. 1302–1324.

[46] L. Qi, *Eigenvalues and invariants of tensors*, J. Math. Anal. Appl., 325 (2007), pp. 1363–1377.

[47] L. Qi, *H^+-eigenvalues of Laplacian and signless Laplacian tensors*, Commun. Math. Sci., 12 (2014), pp. 1045–1064.

[48] L. Qi and Z. Luo, *Tensor Analysis: Spectral Theory and Special Tensors*, SIAM, Philadelphia, 2017.

[49] X. Ren and J. Malik, *Learning a classification model for segmentation*, in Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV'03), 2003, pp. 10–17.

[50] J. A. Rodríguez, *Laplacian eigenvalues and partition problems in hypergraphs*, Appl. Math. Lett., 22 (2009), pp. 916–921.

[51] S. Rota Bulò and M. Pelillo, *A game-theoretic approach to hypergraph clustering*, IEEE Trans. Pattern Anal. Mach. Intell., 35 (2013), pp. 1312–1327.

[52] A. Shashua, R. Zass, and T. Hazan, *Multi-way clustering using super-symmetric nonnegative tensor factorization*, in Proceedings of the European Conference on Computer Vision (ECCV'06), Springer, Berlin, Heidelberg, 2006, pp. 595–608.

[53] J. Shi and J. Malik, *Normalized cuts and image segmentation*, IEEE Trans. Pattern Anal. Mach. Intell., 22 (2000), pp. 888–905.

[54] W. Sun and Y.-X. Yuan, *Optimization Theory and Methods: Nonlinear Programming*, Springer, New York, 2006.

[55] B. Uçar and C. Aykanat, *Revisiting hypergraph models for sparse matrix partitioning*, SIAM Rev., 49 (2007), pp. 595–603, https://doi.org/10.1137/060662459.

[56] J. C. Urschel, J. Xu, X. Hu, and L. T. Zikatanov, *A cascadic multigrid algorithm for computing the Fiedler vector of graph Laplacians*, J. Comput. Math., 33 (2015), pp. 209–226.

[57] U. von Luxburg, *A tutorial on spectral clustering*, Stat. Comput., 17 (2007), pp. 395–416.

[58] Y. Weiss, *Segmentation using eigenvectors: A unifying view*, in Proceedings of the Seventh IEEE International Conference on Computer Vision (ICCV'99), Vol. 2, 1999, pp. 975–982.

[59] J. Xie and A. Chang, *A new type of Laplacian tensor and its Z-eigenvalues of an even uniform hypergraph*, International J. Appl. Math. Stat., 31 (2013), pp. 9–19.

[60] J. Y. Zien, M. D. F. Schlag, and P. K. Chan, *Multilevel spectral hypergraph partitioning with arbitrary vertex sizes*, IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., 18 (1999), pp. 1389–1399.