HYPERGRAPHS WITH EDGE-DEPENDENT VERTEX WEIGHTS: SPECTRAL CLUSTERING BASED ON THE 1-LAPLACIAN

Yu Zhu, Boning Li, and Santiago Segarra

Dept. of Electrical and Computer Engineering, Rice University, USA

ABSTRACT

We propose a flexible framework for defining the 1-Laplacian of a hypergraph that incorporates edge-dependent vertex weights. These weights are able to reflect varying importance of vertices within a hyperedge, thus conferring the hypergraph model higher expressivity than homogeneous hypergraphs. We then utilize the eigenvector associated with the second smallest eigenvalue of the hypergraph 1-Laplacian to cluster the vertices. From a theoretical standpoint based on an adequately defined normalized Cheeger cut, this procedure is expected to achieve higher clustering accuracy than that based on the traditional Laplacian. Indeed, we confirm that this is the case using real-world datasets to demonstrate the effectiveness of the proposed spectral clustering approach. Moreover, we show that for a special case within our framework, the corresponding hypergraph 1-Laplacian is equivalent to the 1-Laplacian of a related graph, whose eigenvectors can be computed more efficiently, facilitating the adoption on larger datasets.

***Index Terms*—** Hypergraphs, Laplacian, spectral clustering, edge-dependent vertex weights, submodular hypergraphs

1. INTRODUCTION

Clustering is the task of dividing a set of entities into several groups such that entities in the same group are more similar to each other than to those in other groups. In graph clustering or partitioning, the vertices of a graph correspond to the entities and the edge weights encode pairwise similarities that can also be understood as the cost of assigning the endpoints of an edge to different clusters. In this way, graph clustering can be cast as an optimization problem that aims to minimize the total cost of cutting the edges across clusters [1, 2].

Although graphs are widely used in a myriad of machine learning tasks, they are limited to representing *pairwise* interactions. By contrast, in many real-world applications the entities engage in *higher-order* relations [3–5]. Examples of these multi-way relations include authors collaborating together to generate a manuscript [6], multiple customers who bought the same product [7], or even multiple news articles containing the same key words [8, 9]. Such relations can be modeled by hypergraphs, where the notion of an edge is generalized to a hyperedge that can connect more than two vertices [3, 4]. Classical hypergraph partitioning approaches mimic graph partitioning and consider homogeneous hyperedge cuts — an identical cost is charged if a hyperedge is cut no matter *how* it is cut [10].

To leverage the fact that different subsets of vertices in a hyperedge may have different structural importance, the concept of an inhomogeneous hyperedge cut has been proposed [11]. In this case, each hyperedge e is associated with a function $w_e : 2^e \to \mathbb{R}_{\geq 0}$ that assigns costs to every possible cut of the hyperedge where 2^e denotes the power set of e. The weight $w_e(\mathcal{S})$ indicates the cost of partitioning e into the two subsets $\mathcal{S}$ and $e \setminus \mathcal{S}$. In particular, when w_e is a submodular function, the corresponding model is called a submodular hypergraph which has desirable mathematical properties, making it convenient for theoretical analysis. A series of results in graph spectral theory [12] including p-Laplacians and Cheeger inequalities have been generalized to submodular hypergraphs [13, 14].

Naturally, the choice of the weight function w_e has a large practical effect on the clustering of a hypergraph and its subsequent downstream tasks. Most existing works consider cardinality-based functions [15–17] in which the value of $w_e(\mathcal{S})$ depends only on the cardinalities of $\mathcal{S}$ and $e \setminus \mathcal{S}$ such as $w_e(\mathcal{S}) = |\mathcal{S}| \cdot |e \setminus \mathcal{S}|$. However, cardinality-based functions are limited in the sense that they cannot incorporate information regarding the different contribution of vertices to a hyperedge. To capture such information, we leverage edge-dependent vertex weights (EDVWs): Every vertex v is associated with a weight $\gamma_e(v)$ for each incident hyperedge e that reflects the contribution of v to e [6]. The hypergraph model with EDVWs is highly relevant in practice. Going back to our earlier examples, EDVWs can be used to model the author positions in a co-authored manuscript [6], the quantity of a product bought by a user in an e-commerce system [7], or the relevance of a word to a document in text mining [8, 9]. In this context, we provide a principled way of defining weight functions w_e based on the EDVWs that guarantee submodularity. Lastly, even when given w_e, finding the optimal (balanced) graph cut is NP-hard [18]. A common workaround is to rely on spectral relaxations based on the combinatorial (2-)Laplacian [1]. Here we outperform this standard approach by leveraging the fact that the second eigenvalue of the 1-Laplacian is equal to the Cheeger constant [13, 19–21].

Contribution. The contributions of this paper can be summarized as follows:
1) We provide a framework to define hyperedge weight functions w_e based on EDVWs that guarantee submodularity.
2) We consider a particular choice of w_e in the proposed framework and show that its corresponding hypergraph 1-Laplacian reduces to the 1-Laplacian of a graph obtained via clique expansion.
3) We leverage the eigenvector associated with the second smallest eigenvalue of the hypergraph 1-Laplacian to cluster the vertices and validate the effectiveness of this method using real-world datasets.

2. PRELIMINARIES

We briefly review some basic mathematical concepts in Section 2.1 and then introduce the submodular hypergraph model and its corresponding 1-Laplacian in Section 2.2. Throughout the paper we assume that the hypergraph is connected.

This work was supported by NSF under award CCF-2008555. E-mails: {yz126, boning.li, segarra}@rice.edu

978-1-6654-0540-9/22/$31.00 ©2022 IEEE

2.1. Mathematical preliminaries

For a finite set $\mathcal{S}$, a set function $F : 2^{\mathcal{S}} \to \mathbb{R}$ is called submodular if $F(\mathcal{S}_1 \cup \{u\}) - F(\mathcal{S}_1) \geq F(\mathcal{S}_2 \cup \{u\}) - F(\mathcal{S}_2)$ for every $\mathcal{S}_1 \subseteq \mathcal{S}_2 \subset \mathcal{S}$ and every $u \in \mathcal{S} \setminus \mathcal{S}_2$. Considering a set function $F : 2^{\mathcal{S}} \to \mathbb{R}$ where $\mathcal{S} = [N] = \{1, 2, \cdots, N\}$, its Lovász extension $f : \mathbb{R}^N \to \mathbb{R}$ is defined as follows. For any $\mathbf{x} \in \mathbb{R}^N$, sort its entries in non-increasing order $x_{i_1} \geq x_{i_2} \geq \cdots \geq x_{i_N}$ and set

$$f(\mathbf{x}) = \textstyle\sum_{j=1}^{N-1} F(\mathcal{S}_j)(x_{i_j} - x_{i_{j+1}}) + F(\mathcal{S})x_{i_N} \tag{1}$$

where $\mathcal{S}_j = \{i_1, \cdots, i_j\}$ for $1 \leq j < N$. The Lovász extension f is convex if and only if F is submodular [22, 23].

2.2. Submodular hypergraphs

Let $\mathcal{H} = (\mathcal{V}, \mathcal{E}, \mu, \{w_e\})$ denote a submodular hypergraph [13] where $\mathcal{V} = [N]$ is the vertex set and $\mathcal{E}$ represents the set of hyperedges. The function $\mu : \mathcal{V} \to \mathbb{R}_+$ assigns positive weights to every vertex. Each hyperedge $e \in \mathcal{E}$ is associated with a submodular function $w_e : 2^e \to \mathbb{R}_{\geq 0}$ that assigns non-negative costs to each possible partition of the hyperedge e. Moreover, w_e is required to satisfy $w_e(\emptyset) = 0$ and be symmetric so that $w_e(\mathcal{S}) = w_e(e \setminus \mathcal{S})$ for any $\mathcal{S} \subseteq e$. The domain of the function w_e can be extended from 2^e to $2^{\mathcal{V}}$ by setting $w_e(\mathcal{S}) = w_e(\mathcal{S} \cap e)$ for any $\mathcal{S} \subseteq \mathcal{V}$, guaranteeing that the submodularity is maintained. Note that if w_e is a constant function, submodular hypergraphs reduce to homogeneous hypergraphs, where the cost of cutting a hyperedge is independent of the specific partition.

A cut is a partition of the vertex set $\mathcal{V}$ into two disjoint, non-empty subsets denoted by $\mathcal{S}$ and its complement $\bar{\mathcal{S}} = \mathcal{V} \setminus \mathcal{S}$. The weight of the cut is defined as the sum of cut costs associated with each hyperedge [13], i.e., $\mathrm{cut}(\mathcal{S}, \bar{\mathcal{S}}) = \sum_{e \in \mathcal{E}} w_e(\mathcal{S})$. The Cheeger constant [12, 13, 21] is defined as

$$h_2 = \min_{\emptyset \subset \mathcal{S} \subset \mathcal{V}} \frac{\mathrm{cut}(\mathcal{S}, \bar{\mathcal{S}})}{\min\{\mathrm{vol}(\mathcal{S}), \mathrm{vol}(\bar{\mathcal{S}})\}}, \tag{2}$$

where $\mathrm{vol}(\mathcal{S}) = \sum_{v \in \mathcal{S}} \mu(v)$ denotes the volume of $\mathcal{S}$. The solution to (2) provides an optimal partitioning in the sense that we obtain two balanced clusters (in terms of their volume) that are only loosely connected, as captured by a small cut weight. Although many alternative clustering formulations exist [1, 2, 24, 25], in this paper we adopt the minimization of (2) as our objective.

Optimally solving (2) has been shown to be NP-hard for graphs [2, 18, 26], let alone weighted hypergraphs. Hence, different relaxations have been proposed, a popular one being spectral clustering, a relaxation based on the second eigenvector of the graph (or hypergraph) Laplacian [1, 2, 8]. This approach is also theoretically justified through the Cheeger inequality [6, 12, 13, 21], where h_2 is *upper bounded* by a function of the second smallest eigenvalue of the Laplacian. However, it has been proved that the Cheeger constant h_2 is *equal* to the second smallest eigenvalue λ_2 of the hypergraph 1-Laplacian, an alternative (non-linear) operator that generalizes the classical Laplacian (cf. Theorem 4.1 in [13]). Moreover, the corresponding partitioning can be obtained by thresholding the eigenvector associated with λ_2 (cf. Theorem 4.3 in [13]). The 1-Laplacian $\triangle_1$ of a submodular hypergraph is defined as an operator that, for all $\mathbf{x} \in \mathbb{R}^N$, induces

$$\langle \mathbf{x}, \triangle_1(\mathbf{x}) \rangle = \textstyle\sum_{e \in \mathcal{E}} f_e(\mathbf{x}), \tag{3}$$

where f_e is the Lovász extension of w_e. Notice that $\triangle_1$ can be alternatively defined in terms of the subdifferential of f_e [13], but the inner product definition in (3) is more instrumental to our development. The eigenvector of $\triangle_1$ associated with λ_2 can be obtained by minimizing

$$R_1(\mathbf{x}) = \frac{\langle \mathbf{x}, \triangle_1(\mathbf{x}) \rangle}{\min_{c \in \mathbb{R}} \|\mathbf{x} - c\mathbf{1}\|_{1,\mu}}, \tag{4}$$

where $\|\mathbf{x}\|_{1,\mu} = \sum_{v \in \mathcal{V}} \mu(v)|x_v|$. Since finding a vector $\mathbf{x}$ that minimizes $R_1(\mathbf{x})$ is equivalent to solving (2), this minimization is also NP-hard, stemming from the non-linear nature of $\triangle_1$. However, developing approximate solutions through this route has shown advantages for submodular hypergraphs [13]. In this paper, we show that this is also the case for the more expressive model of hypergraphs with EDVWs.

Remark (Graph 1-Laplacian). An undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{A})$ with adjacency matrix $\mathbf{A}$ can be regarded as a special case of submodular hypergraphs where $w_e(\mathcal{S})$ equals the edge weight if only one endpoint of the edge is contained in $\mathcal{S}$ and equals zero otherwise. Under this setting, the Lovász extension f_e of w_e for edge $e = (u, v)$ turns out to be $f_e(\mathbf{x}) = A_{uv}|x_u - x_v|$ according to (1). Following (3), we thus obtain that the *graph* 1-Laplacian, which we denote by $\triangle_1^{(g)}$ for convenience, is given by

$$\langle \mathbf{x}, \triangle_1^{(g)}(\mathbf{x}) \rangle = \tfrac{1}{2} \textstyle\sum_{u,v \in \mathcal{V}} A_{uv}|x_u - x_v|. \tag{5}$$

This definition is consistent with prior definitions of the 1-Laplacian in the graph domain [19, 20, 26, 27].

3. SPECTRAL CLUSTERING OF HYPERGRAPHS WITH EDVWS BASED ON 1-LAPLACIAN

Let $\mathcal{H} = (\mathcal{V}, \mathcal{E}, \mu, \kappa, \{\gamma_e\})$ be a hypergraph with EDVWs [6] where $\mathcal{V} = [N]$, $\mathcal{E}$, and μ respectively denote the vertex set, the hyperedge set, and positive vertex weights. The function $\kappa : \mathcal{E} \to \mathbb{R}_+$ assigns positive weights to hyperedges, and those weights can reflect the strength of connection. Each hyperedge $e \in \mathcal{E}$ is associated with a function $\gamma_e : \mathcal{V} \to \mathbb{R}_{\geq 0}$ to assign *edge-dependent vertex weights*. For $v \in e$, $\gamma_e(v)$ is positive and measures the importance of the vertex v to the hyperedge e; for $v \notin e$, we set $\gamma_e(v) = 0$.

The introduction of EDVWs enables the hypergraph to model the cases where the vertices in one hyperedge contribute differently to this hyperedge. For example, in a co-authorship network where authors and papers are respectively modeled as vertices and hyperedges, every author generally has a different degree of contribution to a paper, usually reflected by the order of the authors. This information is lost in homogeneous hypergraphs and is hard to be directly described by submodular hypergraphs, but it can be conveniently represented via EDVWs. Thus, deriving a principled framework to treat EDVWs while still leveraging the benefits of submodular hypergraphs (see Section 2.2) is of paramount practical importance.

3.1. Building submodular hypergraphs from EDVWs

We propose a method to define a class of submodular weight functions w_e based on EDVWs. Putting it differently, we propose a way to convert the hypergraph model with EDVWs into a submodular hypergraph by providing a recipe that generates $\{w_e\}$ from κ and $\{\gamma_e\}$. We do this by exploiting the following result.

Proposition 1. *Define a weight function $w_e : 2^{\mathcal{V}} \to \mathbb{R}_{\geq 0}$ associated with hyperedge e as*

$$w_e(\mathcal{S}) = h_e(\kappa(e)) \cdot g_e(\textstyle\sum_{v \in \mathcal{S}} \gamma_e(v)), \tag{6}$$

for all $\mathcal{S} \subseteq \mathcal{V}$. If $h_e : \mathbb{R}_+ \to \mathbb{R}_+$ is an arbitrary function and $g_e : [0, \sum_{v \in e} \gamma_e(v)] \to \mathbb{R}_{\geq 0}$ is concave, symmetric with respect to $\sum_{v \in e} \gamma_e(v)/2$ and satisfies $g_e(0) = 0$, then w_e is a symmetric submodular function satisfying $w_e(\emptyset) = 0$.

Proof. For every $\mathcal{S}_1 \subseteq \mathcal{S}_2 \subset \mathcal{V}$ and every $u \in \mathcal{V} \setminus \mathcal{S}_2$, set $a_1 = \sum_{v \in \mathcal{S}_1} \gamma_e(v)$, $a_2 = \sum_{v \in \mathcal{S}_2} \gamma_e(v)$ and $b = \gamma_e(u)$. Since g_e is concave, it follows from Lemma 1 (see below) that

$$\begin{aligned} &g_e(\textstyle\sum_{v \in \mathcal{S}_1} \gamma_e(v) + \gamma_e(u)) - g_e(\sum_{v \in \mathcal{S}_1} \gamma_e(v)) \geq \\ &\quad g_e(\textstyle\sum_{v \in \mathcal{S}_2} \gamma_e(v) + \gamma_e(u)) - g_e(\sum_{v \in \mathcal{S}_2} \gamma_e(v)). \end{aligned}$$

Multiplying both sides of the above inequality by the positive value $h_e(\kappa(e))$, it immediately follows that

$$w_e(\mathcal{S}_1 \cup \{u\}) - w_e(\mathcal{S}_1) \geq w_e(\mathcal{S}_2 \cup \{u\}) - w_e(\mathcal{S}_2).$$

Hence, $w_e(\mathcal{S})$ defined in (6) is submodular according to the definition given in Section 2.1. Moreover, $g_e(0) = 0$ and the symmetry of g_e respectively guarantee that $w_e(\emptyset) = 0$ and $w_e(\mathcal{S}) = w_e(\mathcal{V} \setminus \mathcal{S})$ for any $\mathcal{S} \subseteq \mathcal{V}$. □

Lemma 1. *If $a_2 \geq a_1$, $b \geq 0$ and g is a concave function, the inequality $g(a_1 + b) - g(a_1) \geq g(a_2 + b) - g(a_2)$ holds.*

Proof. For the cases where $a_1 = a_2$ or $b = 0$, the result is trivial. For the other cases, recall the definition of concave functions. The following inequality holds for any x and y in the domain of g,

$$g(tx + (1-t)y) \geq tg(x) + (1-t)g(y), \quad \forall t \in [0, 1].$$

Set $x = a_1$, $y = a_2 + b$. For $t = \frac{a_2 - a_1}{a_2 - a_1 + b}$ and $t = \frac{b}{a_2 - a_1 + b}$, the above inequality respectively becomes

$$\begin{aligned} g(a_1 + b) &\geq \tfrac{a_2 - a_1}{a_2 - a_1 + b} g(a_1) + \tfrac{b}{a_2 - a_1 + b} g(a_2 + b), \\ g(a_2) &\geq \tfrac{b}{a_2 - a_1 + b} g(a_1) + \tfrac{a_2 - a_1}{a_2 - a_1 + b} g(a_2 + b). \end{aligned}$$

The proof is completed by respectively adding both sides of these two inequalities together. □

Proposition 1 provides a framework for constructing submodular hypergraphs from EDVWs. Based on (the Lovász extension of) the submodular edge weights, we can further define the hypergraph 1-Laplacian [cf. (3)] and apply spectral clustering by minimizing (4). Although not required by Proposition 1, since $\kappa(e)$ reflects the connection strength of hyperedge e, it is often reasonable in practice to select a non-decreasing function for h_e such as $h_e(x) = 1$ and $h_e(x) = x$. Also notice that for trivial EDVWs — i.e., $\gamma_e(v) = 1$ for all vertices v and incident hyperedges e — the functions defined in (6) reduce to cardinality-based functions [17].

As mentioned in Section 2.2, we can threshold the second eigenvector of the 1-Laplacian to get a 2-partition that achieves the minimum Cheeger cut h_2. More precisely, given a vector $\mathbf{x} \in \mathbb{R}^N$ and a threshold t, a partitioning can be defined as $\mathcal{S} = \{v \in \mathcal{V} \,|\, x_v > t\}$ and its complement. The optimal threshold t can be determined as the one that minimizes the Cheeger cut in (2). An approximation to the computation of the eigenvector [cf. (4)] can be obtained using the inverse power method (IPM). The IPM involves an inner optimization problem that has faster solvers for graphs than general submodular hypergraphs [13, 27]. From a scalability standpoint, this raises the question of whether there exist submodular functions in the proposed framework whose corresponding *hypergraph* 1-Laplacian is identical to some *graph* 1-Laplacian. In the next section, we give one example that meets this condition.

3.2. A reducible case

Given a hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E}, \mu, \kappa, \{\gamma_e\})$ with EDVWs, we define its corresponding clique expansion graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}^{(g)}, \mu, \mathbf{A})$ as an undirected graph endowed with a vertex weight function μ, which is constructed as follows. The vertex set $\mathcal{V}$ and vertex weights μ are identical to those of $\mathcal{H}$. Each hyperedge in $\mathcal{H}$ is replaced by a clique so that $\mathcal{E}^{(g)} = \{(u, v) \,|\, u, v \in e, u \neq v, e \in \mathcal{E}\}$. Lastly, the edge weights contained in the adjacency matrix are defined as $A_{uv} = \sum_{e \in \mathcal{E}} h_e(\kappa(e)) \gamma_e(u) \gamma_e(v)$, where h_e is defined as in Proposition 1. With this construction in place, the following result holds.

Proposition 2. *For any submodular hypergraph constructed from EDVWs following* (6), *if for every hyperedge $e \in \mathcal{E}$ we select the following concave function*

$$g_e(x) = x \cdot (\textstyle\sum_{v \in e} \gamma_e(v) - x), \tag{7}$$

then the resulting hypergraph 1*-Laplacian is identical to the* 1*-Laplacian of its corresponding clique expansion graph.*

Proof. For any $\mathbf{x} \in \mathbb{R}^N$, sort the subset of its entries indexed by the vertices incident to hyperedge e in non-increasing order $x_{i_1^{(e)}} \geq x_{i_2^{(e)}} \geq \cdots \geq x_{i_{|e|}^{(e)}}$ and define a series of sets $\mathcal{S}_j^{(e)} = \{i_1^{(e)}, \cdots, i_j^{(e)}\}$ for $1 \leq j < |e|$. It follows from (1) that the Lovász extension of the defined w_e can be written as

$$\begin{aligned} f_e(\mathbf{x}) &\overset{(a)}{=} \textstyle\sum_{j=1}^{|e|-1} w_e(\mathcal{S}_j^{(e)})(x_{i_j^{(e)}} - x_{i_{j+1}^{(e)}}), \\ &\overset{(b)}{=} \textstyle\sum_{j=1}^{|e|-1} h_e(\kappa(e)) g_e(\sum_{v \in \mathcal{S}_j^{(e)}} \gamma_e(v))(x_{i_j^{(e)}} - x_{i_{j+1}^{(e)}}), \\ &\overset{(c)}{=} h_e(\kappa(e)) \textstyle\sum_{j=1}^{|e|-1} \sum_{u \in \mathcal{S}_j^{(e)}, v \in e \setminus \mathcal{S}_j^{(e)}} \gamma_e(u) \gamma_e(v)(x_{i_j^{(e)}} - x_{i_{j+1}^{(e)}}), \\ &= \tfrac{h_e(\kappa(e))}{2} \textstyle\sum_{u, v \in e} \gamma_e(u) \gamma_e(v) |x_u - x_v|, \end{aligned}$$

where in (a) we leveraged the fact that $w_e(e) = 0$, and (b), (c) respectively followed (6) and (7). Then according to (3), the hypergraph 1-Laplacian $\triangle_1(\mathbf{x})$ should satisfy

$$\langle \mathbf{x}, \triangle_1(\mathbf{x}) \rangle = \textstyle\sum_{e \in \mathcal{E}} \frac{h_e(\kappa(e))}{2} \sum_{u, v \in e} \gamma_e(u) \gamma_e(v) |x_u - x_v|. \tag{8}$$

It follows from (5) that the 1-Laplacian $\triangle_1^{(g)}(\mathbf{x})$ of the clique expansion graph $\mathcal{G}$ must satisfy

$$\begin{aligned} \langle \mathbf{x}, \triangle_1^{(g)}(\mathbf{x}) \rangle &= \tfrac{1}{2} \textstyle\sum_{u, v \in \mathcal{V}} \sum_{e \in \mathcal{E}} h_e(\kappa(e)) \gamma_e(u) \gamma_e(v) |x_u - x_v|, \\ &\overset{(d)}{=} \textstyle\sum_{e \in \mathcal{E}} \frac{h_e(\kappa(e))}{2} \sum_{u, v \in e} \gamma_e(u) \gamma_e(v) |x_u - x_v|, \end{aligned} \tag{9}$$

where in (d) we exchanged the order of the two summations and used the fact that $\gamma_e(v) = 0$ for $v \notin e$. Since $\mathbf{x}$ has been chosen arbitrarily, a direct comparison of (8) and (9) reveals that $\triangle_1(\mathbf{x})$ and $\triangle_1^{(g)}(\mathbf{x})$ coincide. □

Proposition 2 enables us to transform an eigenproblem in a hypergraph to an equivalent eigenproblem in a carefully constructed graph with the same number of vertices. This allows us to leverage efficient IPM solvers [13, 27] when approximating the sought eigenvector of the 1-Laplacian. It should be noted that the choice of g_e in (7) is not only computationally convenient but also meaningful in practice. Indeed, the hyperedge cut cost achieves the maximum value when the hyperedge is partitioned into two parts that have the same sum of EDVWs. Putting it differently, cutting a hyperedge by only shaving off an unimportant node (corresponding to a low EDVW) is lightly penalized compared to cutting the hyperedge in half, where this half is defined by the relative weights of the EDVWs.

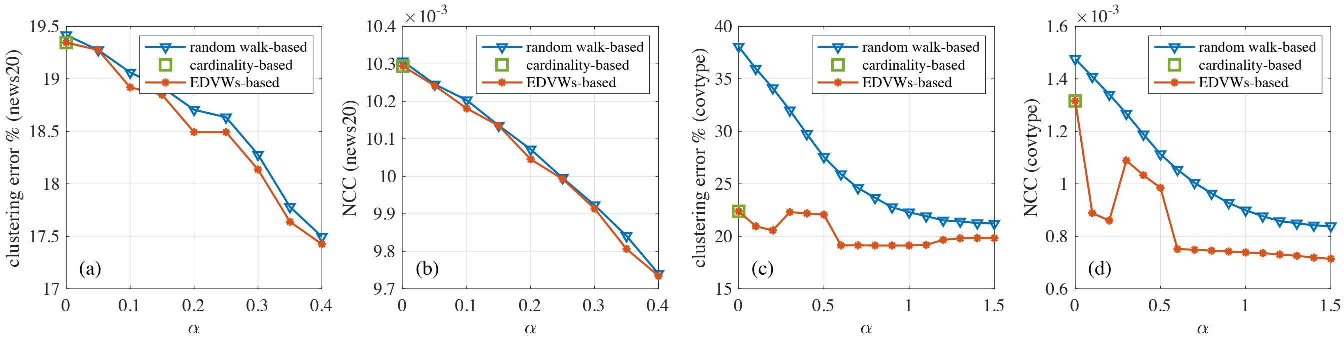

Fig. 1. Clustering performance in two real-world datasets as a function of the parameter α, which defines the EDVWs. (a-b) Clustering error and NCC in the 20 Newsgroups dataset. (c-d) Clustering error and NCC in the Covertype dataset.

4. NUMERICAL EXPERIMENTS

We evaluate the performance of the proposed spectral clustering method for hypergraphs with EDVWs by focusing on the 2-partition case.[1] We compare the proposed method with the more standard spectral clustering approach based on the classical Laplacian and a 1-Laplacian approach that ignores the EDVWs.

We summarize the proposed method (termed as EDVWs-based) as follows: (i) We build a submodular hypergraph from EDVWs according to Proposition 1 where we select $h_e(x) = x$ and g_e as in (7). For each vertex v, we set its weight $\mu(v)$ to its degree defined as $\deg(v) = \sum_{e\in\mathcal{E}:v\in e}\vartheta_e$ where $\vartheta_e = \max_{\mathcal{S}\subseteq e} w_e(\mathcal{S})$. For this choice of vertex weights, the objective function in (2) is called normalized Cheeger cut (NCC) [2]. (ii) We project the constructed submodular hypergraph onto its corresponding clique expansion graph as explained in Section 3.2. (iii) We compute the second eigenvector of the corresponding graph 1-Laplacian using the IPM [27]. (iv) We threshold the obtained eigenvector using the optimal threshold value (see the last paragraph in Section 3.1), which achieves the minimum NCC.

Datasets. We evaluate our performance on two widely used real-world datasets.

20 *Newsgroups*[2]: This dataset contains documents in different categories. For our 2-partition case, we consider the documents in categories 'rec.motorcycles' and 'rec.sport.hockey' and the 100 most frequent words in the corpus after removing stop words and words appearing in $> 10\%$ and $< 0.2\%$ of the documents. We then remove documents containing less than 5 selected words, leaving us with 1406 documents. A document (vertex) belongs to a word (hyperedge) if the word appears in the document. The EDVWs are taken as the corresponding tf-idf (term frequency-inverse document frequency) values [28] to the power of α, where α is an adjustable parameter.

Covertype[3]: This dataset contains areas of different forest cover types. We consider two cover types (types 4 and 5 in the dataset) and all numerical features. Each numerical feature is first quantized into 20 bins of equal size and then mapped to hyperedges. The resulting hypergraph has 12240 vertices and 196 hyperedges. For each hyperedge (bin), we compute the distance between each feature value in this bin and their median, and then normalize these distances to the range $[0, 1]$. The EDVWs are computed as $\exp(-\alpha \cdot \text{distance})$. In this way, larger EDVWs are given to nodes whose feature values are close to the typical feature value in that corresponding bin (hyperedge).

Following the setting in [8], for both datasets we set the hyperedge weight $\kappa(e)$ to the standard deviation of the EDVWs $\gamma_e(v)$ for all $v \in \mathcal{V}$.

Baselines. We compare our proposed approach with two baseline methods:

Random walk-based: The paper [8] defines a hypergraph Laplacian based on random walks with EDVWs. We compute the second eigenvector of the normalized hypergraph Laplacian proposed in [8] and then threshold it to get the partitioning.

Cardinality-based: In the description of our datasets, when $\alpha = 0$ we are effectively ignoring the EDVWs and the corresponding submodular weight functions w_e become cardinality-based.

Results. The results are shown in Fig. 1, where the clustering error is computed as the fraction of incorrectly clustered samples. Focusing first on the 20 Newsgroups dataset (a-b), we can see that the proposed method achieves better clustering performance than the random-walk method based on the classical Laplacian. This gain can be explained by our method yielding a smaller NCC value. More conspicuously, when EDVWs are ignored ($\alpha = 0$), the clustering performance is severely deteriorated, underscoring the importance of the added modeling flexibility of hypergraphs with EDVWs. Shifting attention to the Covertype dataset (c-d), the same trends are observed. However, in this case there are marked performance differences between our method and the one based on random walks, highlighting the value of considering the non-linear 1-Laplacian in spectral clustering.

5. CONCLUSIONS

We developed a framework for defining submodular weight functions based on EDVWs and provided an example that can be reduced to a clique expansion graph. Numerical results indicated that the combination of the hypergraph 1-Laplacian and EDVWs helps improve clustering performance. Future research avenues include: (i) Studying if there exist other choices in the context of Proposition 1 such that the corresponding hypergraph 1-Laplacian coincides with some graph 1-Laplacian, (ii) Developing efficient computation methods for the hypergraph 1-Laplacian's eigenvectors that are applicable to general submodular weight functions, and (iii) Designing multi-way partitioning algorithms based on non-linear Laplacians [2].

[1] The code needed to replicate the experiments here presented can be found at `https://github.com/yuzhu2019/hg_l1`.

[2] `http://qwone.com/~jason/20Newsgroups/`

[3] `https://archive.ics.uci.edu/ml/datasets/covertype`

6. REFERENCES

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

[2] Thomas Bühler and Matthias Hein, “Spectral clustering based on the graph p-Laplacian,” in *ICML*, 2009, pp. 81–88.

[3] Federico Battiston, Giulia Cencetti, Iacopo Iacopini, Vito Latora, Maxime Lucas, Alice Patania, Jean-Gabriel Young, and Giovanni Petri, “Networks beyond pairwise interactions: Structure and dynamics,” *Physics Reports*, vol. 874, pp. 1–92, 2020.

[4] Michael T Schaub, Yu Zhu, Jean-Baptiste Seby, T Mitchell Roddenberry, and Santiago Segarra, “Signal processing on higher-order networks: Livin’ on the edge... and beyond,” *Signal Processing*, vol. 187, pp. 108149, 2021.

[5] T. Mitchell Roddenberry, Nicholas Glaze, and Santiago Segarra, “Principled simplicial neural networks for trajectory prediction,” in *ICML*, 2021, pp. 9020–9029.

[6] Uthsav Chitra and Benjamin Raphael, “Random walks on hypergraphs with edge-dependent vertex weights,” in *ICML*, 2019, pp. 1172–1181.

[7] Jianbo Li, Jingrui He, and Yada Zhu, “E-tail product return prediction via hypergraph-based local graph cut,” in *KDD*, 2018, pp. 519–527.

[8] Koby Hayashi, Sinan G Aksoy, Cheong Hee Park, and Haesun Park, “Hypergraph random walks, Laplacians, and clustering,” in *CIKM*, 2020, pp. 495–504.

[9] Yu Zhu, Boning Li, and Santiago Segarra, “Co-clustering vertices and hyperedges via spectral hypergraph partitioning,” *arXiv preprint arXiv:2102.10169*, 2021.

[10] Matthias Hein, Simon Setzer, Leonardo Jost, and Syama Sundar Rangapuram, “The total variation on hypergraphs - Learning on hypergraphs revisited,” *NIPS*, vol. 26, pp. 2427–2435, 2013.

[11] Pan Li and Olgica Milenkovic, “Inhomogoenous hypergraph clustering with applications,” in *NIPS*, 2017, pp. 2305–2315.

[12] Fan RK Chung, *Spectral Graph Theory*, vol. 92, American Mathematical Soc.

[13] Pan Li and Olgica Milenkovic, “Submodular hypergraphs: p-Laplacians, Cheeger inequalities and spectral clustering,” in *ICML*, 2018, pp. 3014–3023.

[14] Yuichi Yoshida, “Cheeger inequalities for submodular transformations,” in *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, 2019, pp. 2582–2601.

[15] Dengyong Zhou, Jiayuan Huang, and Bernhard Schölkopf, “Learning with hypergraphs: Clustering, classification, and embedding,” *NIPS*, vol. 19, pp. 1601–1608, 2006.

[16] Sameer Agarwal, Kristin Branson, and Serge Belongie, “Higher order learning with graphs,” in *ICML*, 2006, pp. 17–24.

[17] Nate Veldt, Austin R Benson, and Jon Kleinberg, “Hypergraph cuts with general splitting functions,” *arXiv preprint arXiv:2001.02817*, 2020.

[18] Dorothea Wagner and Frank Wagner, “Between min cut and graph bisection,” in *International Symposium on Mathematical Foundations of Computer Science*. Springer, 1993, pp. 744–750.

[19] Kung-Ching Chang, “Spectrum of the 1-Laplacian and Cheeger’s constant on graphs,” *Journal of Graph Theory*, vol. 81, no. 2, pp. 167–207, 2016.

[20] Kung-Ching Chang, Sihong Shao, and Dong Zhang, “Nodal domains of eigenvectors for 1-Laplacian on graphs,” *Advances in Mathematics*, vol. 308, pp. 529–574, 2017.

[21] Francesco Tudisco and Matthias Hein, “A nodal domain theorem and a higher-order Cheeger inequality for the graph p-Laplacian,” *Journal of Spectral Theory*, vol. 8, no. 3, pp. 883–908, 2018.

[22] László Lovász, “Submodular functions and convexity,” in *Mathematical Programming The State of the Art*, pp. 235–257. Springer, 1983.

[23] Francis Bach, “Learning with submodular functions: A convex optimization perspective,” *Foundations and Trends® in Machine Learning*, vol. 6, no. 2–3, pp. 145–373, 2013.

[24] Gunnar Carlsson, Facundo Mmoli, Alejandro Ribeiro, and Santiago Segarra, “Axiomatic construction of hierarchical clustering in asymmetric networks,” in *IEEE Intl. Conf. on Acoustics, Speech and Signal Process. (ICASSP)*, 2013, pp. 5219–5223.

[25] Gunnar Carlsson, Facundo Mmoli, Alejandro Ribeiro, and Santiago Segarra, “Hierarchical clustering of asymmetric networks,” *Adv. Data Anal. Classif.*, vol. 12, pp. 65–105, 2018.

[26] Arthur Szlam and Xavier Bresson, “Total variation and Cheeger cuts,” in *ICML*, 2010, pp. 1039–1046.

[27] Matthias Hein and Thomas Bühler, “An inverse power method for nonlinear eigenproblems with applications in 1-spectral clustering and sparse PCA,” *NIPS*, vol. 23, pp. 847–855, 2010.

[28] Jure Leskovec, Anand Rajaraman, and Jeffrey David Ullman, *Mining of massive datasets*, Cambridge university press, 2020.