Hypergraph Random Walks, Laplacians, and Clustering

Koby Hayashi
Georgia Institute of Technology
Atlanta, (GA), USA
khayashi9@gatech.edu

Sinan G. Aksoy
Pacific Northwest National Laboratory
Richland, (WA), USA
sinan.aksoy@pnnl.gov

Cheong Hee Park
Chungnam National University
Daejeon, Korea
cheonghee@cnu.ac.kr

Haesun Park
Georgia Institute of Technology
Atlanta, (GA), USA
hpark@cc.gatech.edu

ABSTRACT

We propose a flexible framework for clustering hypergraph-structured data based on recently proposed random walks utilizing edge-dependent vertex weights. When incorporating edge-dependent vertex weights (EDVW), a weight is associated with each vertex-hyperedge pair, yielding a weighted incidence matrix of the hypergraph. Such weightings have been utilized in term-document representations of text data sets. We explain how random walks with EDVW serve to construct different hypergraph Laplacian matrices, and then develop a suite of clustering methods that use these incidence matrices and Laplacians for hypergraph clustering. Using several data sets from real-life applications, we compare the performance of these clustering algorithms experimentally against a variety of existing hypergraph clustering methods. We show that the proposed methods produce high-quality clusters and conclude by highlighting avenues for future work.

CCS CONCEPTS

• **Information Systems**; • **Information Systems Applications**; • **Data mining**; • **Clustering**;

KEYWORDS

hypergraphs; random walks; clustering; Laplacian; Symmetric NMF; Joint NMF; edge-dependent vertex weights

ACM Reference Format:
Koby Hayashi, Sinan G. Aksoy, Cheong Hee Park, and Haesun Park. 2020. Hypergraph Random Walks, Laplacians, and Clustering. In *The 29th ACM International Conference on Information and Knowledge Management (CIKM '20), October 19–23, 2020, Virtual Event, Ireland*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3340531.3412034

CIKM '20, October 19–23, 2020, Virtual Event, Ireland

ACM ISBN 978-1-4503-6859-9/20/10…$15.00
https://doi.org/10.1145/3340531.3412034

1 INTRODUCTION

While graphs serve as a popular tool for modeling a myriad of data analytics tasks, graphs are limited to representing pairwise relationships between objects. However, data sets frequently contain multi-way relationships. For instance, in a term-document matrix that is frequently used to represent text data, multiple terms are related to each other from their appearance in the same document. Multi-way relationships also abound in many contexts, such as when multiple people author a paper, groups of proteins interact, or mutations in multiple genes are associated with a disease. These multi-way relationships are different from multiple binary relationships. More generally, k-way relationships occur whenever information naturally presents as set-valued, bipartite or tabular. In such cases, hypergraphs – generalizations of graphs in which edges may link any number of vertices – are more appropriate.

While hypergraph-structured data is widely prevalent, utilizing a hypergraph model to perform analytics tasks is often challenging. First and foremost, a primary difficulty concerns how to best *represent* a hypergraph for key analytics tasks such as clustering. A number of fundamental graph representations such as the adjacency matrix or Laplacian matrix, have no obvious or canonical analog in the hypergraph setting. In particular, developing such representations is especially challenging for non-uniform hypergraphs, which appear most often in real applications. Furthermore, work by Agarwal [2] has shown that many hypergraph Laplacian matrices are in fact directly related to various graph expansions of a hypergraph, and in this sense, do not fully capture the higher-order relationships modeled by the hypergraph. Secondly, another difficulty concerns devising *analytic methods* that can effectively utilize these hypergraph representations. Indeed, hypergraph representations such as tensors for uniform hypergraphs, where the orders of all hyperedges are same, while faithful in capturing higher-order relationships, may be limited to special cases and difficult or prohibitively expensive to adopt and analyze in practice, due to their large dimensionality or otherwise complicated properties.

One promising approach for addressing these challenges is rooted in the study of random walks on hypergraphs. Much of the work on random walks on hypergraphs has limited applicability to real data because it only considers uniform hypergraphs [11, 25, 26]. Other work has considered non-uniform hypergraph random walks, but analyzes simple random walks, in which vertices are chosen uniformly at random from a hyperedge. However, these random walks have been shown [8] to be equivalent to a random walk on the

graph clique expansion of the hypergraph. Recent work by Chitra and Raphael [8] has shown incorporating so-called edge-dependent vertex weights (EDVW) into the random walk is a necessary condition to circumvent this equivalence, therefore to better capture the higher-order properties of hypergraphs. Such vertex weightings, associated with each vertex-hyperedge pair, have appeared in a number of different contexts, such as in term-document matrix represented via tf-idf (term frequency and inverse document frequency), weighting based on the significance in the author order in research paper data, or in general, whenever incidence structures have weighted (rather than binary) cells.

In this work, we use EDVW random walks to develop a diverse and flexible framework for clustering hypergraph data. We explain how to construct several different hypergraph representations in incidence matrices and Laplacians based on EDVW random walks, as well as how one can apply a number of different clustering algorithms to these representations. In addition, we experimentally compare the performance of these EDVW random walk-based clustering approaches to existing hypergraph clustering approaches.

We organize our work as follows: in Section 2, we provide the necessary preliminaries and briefly review random walks on hypergraphs. In Section 3, we explain how the probability transition matrix of EDVW random walks may be utilized to construct a number of different hypergraph Laplacians, and survey appropriate possibilities from the literature. In Section 4, we define clustering methodologies that may be used in conjunction with the aforementioned hypergraph representations. In Section 5, we review other approaches from the literature, consisting of both different hypergraph representations as well as different clustering methodologies. In Section 6 we compare these approaches to ours experimentally: we describe our test datasets, experimental setup, clustering performance evaluation metrics, and report our findings.

2 HYPERGRAPHS AND RANDOM WALKS

2.1 Preliminaries

Hypergraphs are generalizations of graphs in which edges can connect any number of vertices. More formally, a hypergraph $H = (V, E)$ is a set of vertices $V = \{v_1, \ldots, v_n\}$ and a family of edges $E = (e_1, \ldots, e_m)$ where $e_i \subseteq V$ for $i = 1, \ldots, m$. A graph is a uniform hypergraph of edge order 2, i.e., every edge e in a graph has $|e| = 2$. Throughout, we assume the hypergraph has no isolated vertices, i.e. $\bigcup_{e \in E} e = V$, and no empty edges. A hypergraph may be represented by its (unweighted) incidence matrix $\mathbf{X} \in \{0, 1\}^{|E| \times |V|}$, where

$$\mathbf{X}_{ev} = \begin{cases} 1 & \text{if vertex } v \text{ belongs to hyperedge } e, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

(Note that this incidence matrix is sometimes transposed to denote vertex by hyperedge relationships. However, we will use the above notation to make it consistent with the notations in the closely related papers.) The dual of a hypergraph, denoted H^*, is the hypergraph associated with $\mathbf{X}^\mathsf{T}$.

In practice, it is common that hypergraphs are transformed to graphs. One such popular transformation replaces each hyperedge with a clique, and is thus called the *clique expansion*. More precisely, the clique expansion of a hypergraph $H = (V, E)$ is a graph on the same vertex set, with edge set $\{\{u, v\} \mid u, v \in e \text{ for some } e \in E\}$. The clique expansion has weighted adjacency matrix given by $\mathbf{X}^\mathsf{T}\mathbf{X}$, where the (i, j) entry denotes the number of shared hyperedges to which vertices i and j belong.

Although such transformations are convenient in that they facilitate the application of graph-theoretic methods, they also have several drawbacks. First, the clique expansion is *lossy* in the sense that non-isomorphic hypergraphs may have identical clique expansions. In fact, recent work by Kirkland [19] confirms this information loss persists even when hypergraph duality is considered: that is, the *pair* of matrices, $\mathbf{X}^\mathsf{T}\mathbf{X}$ and $\mathbf{X}\mathbf{X}^\mathsf{T}$, corresponding to the weighted clique expansion of a hypergraph and its dual, still does not uniquely identify a hypergraph up to isomorphism. This information loss is a primary reason why clique expansion based hypergraph representations are sometimes criticized.

In addition to information loss, another drawback of clique expansions is their density. In particular, since each hyperedge of size k contributes $\binom{k}{2}$ edges in the clique expansion, a hypergraph with large maximum edge size will have a clique expansion that may be prohibitively dense to analyze or even hold in computer memory. Nonetheless, as will soon be clear, clique expansions are a useful reference point for understanding hypergraph random walks, as well their associated Laplacians.

2.2 Random walks

A random walk on a hypergraph $H = (V, E)$ is a discrete-time Markov chain $X_1, X_2, \ldots$, on state space V defined by given transition probabilities. Letting $\omega : E \to \mathbb{R}_+$ denote any function that assigns positive weights to the hyperedges of a hypergraph, a standard formulation for a hypergraph random walk may be given as follows: if at time t, the current state is $X_t = v_t$, then

(1) Select a hyperedge $e \ni v_t$ with probability proportional to $\omega(e)$.
(2) Select a vertex $v \in e$ uniformly at random, and set $X_{t+1} = v$.

In this random walk, a vertex is chosen uniformly at random from a hyperedge, and we refer to it as a *simple random walk*. We note that a number of hypergraph random walks studied in the literature follow this form; see [4, 11, 33].

For the special case of graphs, describing the random walk as a two-step process is generally redundant: since an edge in a graph can only connect *two* vertices, the selection of an incident edge uniquely determines the next state in the chain. For more on graph random walks, see [3]. In contrast, a hyperedge may connect any number of vertices, any of which could be chosen for the next state. Accordingly, the second step above is the key for generalizing random walks on graphs to hypergraphs. Focusing on this step, Chitra and Raphael [8] suggest choosing a vertex from a hyperedge using vertex weightings *specific to that hyperedge*. More formally, for $e \in E$, letting $\gamma_e : e \to \mathbb{R}_+$ denote the weighting function for a hyperedge e, we have if $X_t = v_t$, then

(1) Select a hyperedge $e \ni v_t$ with probability proportional to $\omega(e)$.
(2) Select a vertex $v \in e$ with probability proportional to $\gamma_e(v)$, and set $X_{t+1} = v$.

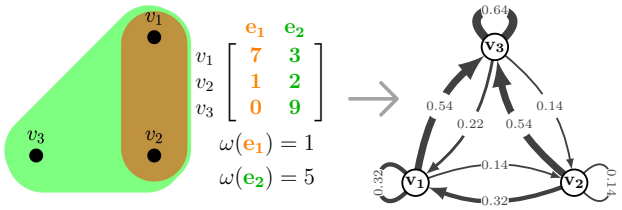

Figure 1: A hypergraph with EDVW and hyperedge weights (left) and the representative digraph of the corresponding EDVW random walk (right).

The collection of functions $\{\gamma_e\}_{e\in E}$ is called an *edge-independent* vertex weighting (EIVW) of H if for every vertex, $\gamma_e(v) = \gamma_{e'}(v)$ for all pairs of hyperedges e, e' containing v [8]. Otherwise, the vertex-weighting is *edge-dependent*. Edge-dependent vertex weights (EDVW) may also be represented by a weighted incidence matrix $\mathbf{R} \in \mathbb{R}_{\geq 0}^{|E|\times|V|}$,

$$\mathbf{R}_{ev} = \begin{cases} \gamma_e(v) & \text{if vertex } v \text{ belongs to hyperedge } e, \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

Stated equivalently, the edge-dependent condition is that there is a column in $\mathbf{R}$ in which the nonzero entries are not all equal. As we will see in Section 6, real data can be naturally represented with EDVW. A good example is the term-document matrix for text data sets, where term frequency-inverse document frequency (tf-idf) matrices are natural analogies for EDVW, which we view as a weighted hypergraph incidence matrix (more precisely, as $\mathbf{R}^{\mathsf{T}}$ with terms as vertices and documents as hyperedges). EDVW also naturally arise for other types of data, such as author-position for author-paper networks, and association scores for gene-disease data. In general, whenever vertex-weights are edge-dependent, we will call the random walk described above an *EDVW random walk*.

In a simple random walk on a hypergraph without vertex weights (or with trivial edge-independent vertex weights), a random walk is *always* equivalent [1] to a random walk on the clique expansion graph of the hypergraph, under some edge weighting of the clique expansion as shown in [8]. They show EDVW are necessary for a random walk on a hypergraph not to be characterized simply as a random walk on the clique expansion graph. In fact, EDVW random walks may be *non-reversible*[2] Markov chains, which implies that they cannot be represented as random walks on *any* undirected graph. In summary, one potential avenue for capturing higher-order properties of hypergraphs is through EDVW random walks [8]. Next, we describe how to utilize EDVW random walks to construct various Laplacian matrices for hypergraph clustering.

3 FROM RANDOM WALKS TO HYPERGRAPH LAPLACIANS

In graph theory, random walks serve as an implicit or explicit foundation for constructing a number of Laplacian matrices. For instance, when $\mathbf{P}$ is the transition probability matrix of the random walk, the random walk Laplacian is $\mathbf{I} - \mathbf{P}$. The normalized Laplacian matrix popularized by Chung [9] also has eigenvalues that are related to those of $\mathbf{P}$ by elementary shifts and scalings. Consequently, it is unsurprising the study of Laplacians is deeply intertwined with that of random walks; for more, see the monograph [9].

Here, we will explain how one may similarly construct various Laplacian matrices for hypergraphs using the transition probability matrix $\mathbf{P}$ of an EDVW random walk [8]. To formally define the transition probability matrix, let $\mathbf{R}$ denote the $|E|\times|V|$ vertex-weight matrix, with $\mathbf{R}_{ev} = \gamma_e(v)$ if $v \in e$ and 0 otherwise. Note that $\mathbf{R}^T$ is a weighted incidence matrix where the weight of each vertex is dependent on the hyperedge it is incident to. Similarly, let $\mathbf{W}$ denote the $|V| \times |E|$ hyperedge-weight matrix, with $\mathbf{W}_{ve} = \omega(e)$ if $v \in e$ and 0 otherwise. Finally, let $\mathbf{D}_V = \text{diag}(\mathbf{We})$ and $\mathbf{D}_E = \text{diag}(\mathbf{Re})$ denote the diagonal vertex degree and hyperedge weight matrices, where $\mathbf{e}$ denotes the vector of an appropriate dimension with all its components ones. The transition probability matrix for the EDVW random walk is given by

$$\mathbf{P} = \mathbf{D}_V^{-1}\mathbf{W}\mathbf{D}_E^{-1}\mathbf{R} \tag{3}$$

As we explain next, this matrix will be used explicitly to construct a number of different hypergraph Laplacians via its interpretation as an edge-weighted directed graph.

3.1 Representative digraph

Recall the transition matrix $\mathbf{P}$ completely defines the random walk on a hypergraph H. One may also represent $\mathbf{P}$, and hence the random walk, as a directed graph (digraph) on vertex set V and edge set $E = \{(i, j) \,|\, \mathbf{P}_{ij} > 0\}$, where the edge weight of (i, j) is simply the transition probability $\mathbf{P}_{ij}$. We call this the *representative digraph* of the random walk. Figure 1 illustrates an example of a hypergraph with hyperedge weights, EDVW weights presented as $\mathbf{R}^{\mathsf{T}}$, and the representative digraph of the associated random walk.

When derived from hypergraph random walks, representative digraphs have several notable properties. First, they do not contain any source or sink vertices since $\mathbf{P}_{ij}$ is nonzero if and only if $\mathbf{P}_{ji}$ is nonzero as well. Furthermore, the representative digraph of a hypergraph random walk is also strongly connected if and only if the hypergraph is connected. Consequently, just as any hypergraph may be written as the vertex and hyperedge-disjoint union of connected hypergraphs, its representative digraph may also be represented as the vertex and edge-disjoint union of strongly connected components – a property that doesn't necessarily hold for directed graphs in general. This means one can apply our proposed clustering methodologies to cluster any hypergraph on a per connected component basis, analogous to how graph clustering methodologies are sometimes performed separately on each connected component of the graph. Lastly, since representative digraphs always contains loop edges of the form (i, i), this guarantees hypergraph random walks are always *aperiodic*. Therefore, a random walk on any connected hypergraph is *ergodic*, which guarantees convergence to the stationary distribution.

3.2 Laplacians based on EDVW random walks

Via the representative digraph, Laplacians for edge-weighted directed graphs naturally serve to construct random-walk based

[1] Two random walks on the same state space are equivalent if they have the same transition probability between each pair of states.
[2] A random walk with probability transition matrix $\mathbf{P}$ and stationary distribution $\boldsymbol{\pi}$ is reversible if $\boldsymbol{\pi}_i\mathbf{P}_{ij} = \boldsymbol{\pi}_j\mathbf{P}_{ji}$ for all pairs of states i, j.

Laplacians for hypergraphs. A number of different directed graph Laplacians have been proposed, and they could be utilized in this context. Perhaps most natural for our purposes are the combinatorial and normalized digraph Laplacians matrices proposed by Chung [10]. Indeed, we note Chitra and Raphael adopt Chung's combinatorial digraph Laplacian as their hypergraph Laplacian in [8]. To define these matrices, recall the stationary distribution $\boldsymbol{\pi}$ of a random walk is the all-positive dominant left eigenvector of the transition probability matrix $\mathbf{P}$,

$$\boldsymbol{\pi}^{\mathsf{T}}\mathbf{P} = \boldsymbol{\pi}^{\mathsf{T}}, \tag{4}$$

scaled to have unit 1-norm. By the Perron Frobenius theorem, the stationary distribution $\boldsymbol{\pi}$ exists if the matrix $\mathbf{P}$ is irreducible, which, in turn, occurs precisely when the representative digraph is strongly connected. Letting $\boldsymbol{\Phi} = \text{diag}(\boldsymbol{\pi})$, Chung defines the directed combinatorial Laplacian $\mathbf{L}$ and normalized Laplacian $\mathcal{L}$ as follows:

$$\mathbf{L} = \boldsymbol{\Phi} - \frac{\boldsymbol{\Phi}\mathbf{P} + \mathbf{P}^{\mathsf{T}}\boldsymbol{\Phi}}{2} \tag{5}$$

$$\mathcal{L} = \boldsymbol{\Phi}^{-\frac{1}{2}}\mathbf{L}\boldsymbol{\Phi}^{-\frac{1}{2}} = \mathbf{I} - \frac{\boldsymbol{\Phi}^{\frac{1}{2}}\mathbf{P}\boldsymbol{\Phi}^{-\frac{1}{2}} + \boldsymbol{\Phi}^{-\frac{1}{2}}\mathbf{P}^{\mathsf{T}}\boldsymbol{\Phi}^{\frac{1}{2}}}{2} \tag{6}$$

Clearly, both of the above matrices are symmetric. When applied to graphs, $\mathbf{L}$ and $\mathcal{L}$ above are equal to the combinatorial and normalized graph Laplacians, respectively.

Given their explicit basis in random walks and applicability to any irreducible transition matrix $\mathbf{P}$, Chung's Laplacians are natural choices for deriving hypergraph Laplacians based in EDVW random walks. Nonetheless, there are other directed Laplacians proposed and studied in the literature that could serve, some of which are asymmetric. Bauer [5] studies an asymmetric digraph Laplacian that, for digraphs without source vertices, is defined as $\mathbf{I} - \mathbf{P}$. Li and Zhang [24] study the asymmetric digraph Laplacian $\boldsymbol{\Gamma} = \boldsymbol{\Phi}^{1/2}(\mathbf{I} - \mathbf{P})\boldsymbol{\Phi}^{-1/2}$, which is related to Chung's normalized Laplacian $\mathcal{L}$ above by $\mathcal{L} = \frac{\boldsymbol{\Gamma}+\boldsymbol{\Gamma}^{\mathsf{T}}}{2}$. Like Chung's normalized digraph Laplacian $\mathcal{L}$, $\boldsymbol{\Gamma}$ is also a generalization of the normalized graph Laplacian, which it reduces to in the graph case.

Lastly, one final class of digraph matrices and Laplacians, which have recently received attention in the literature, utilize *complex values*. Mohar and Guo [17] propose a Hermitian digraph adjacency matrix which encodes adjacency using the imaginary unit i, or, as proposed more recently [28], the sixth root of unity. While such matrices have varied algebraic properties that capture combinatorial properties of the directed graph, their applicability and effectiveness as representations for clustering has yet to be established.

One exception, however, is recent work by Cucuringu [12]. Utilizing a variant of Mohar and Guo's matrix, Cucuringu proposes a simple digraph spectral clustering algorithm and justifies its effectiveness by analyzing its performance in recovering planted clusters from the Directed Stochastic Block Model, a generalization of the classical SBM [18]. Adapted to our setting, Cucuringu's matrix is

$$\mathbf{B} = i \cdot \left(\mathbf{P} - \mathbf{P}^{\mathsf{T}}\right), \tag{7}$$

where i is the unit imaginary number. We note $\mathbf{B}$ is both Hermitian and skew-symmetric. This skew-symmetrization of the transition matrix may be thought of as constructing an *oriented* digraph (i.e. a digraph without reciprocal edges) from the representative digraph, in which the edge weight between i and j is the difference in their probability transitions, and directionality is encoded by sign. For clustering, Cucuringu suggests normalizing this matrix by the diagonal matrix with $\mathbf{S}_{ii} = \sum_j |\mathbf{B}_{ij}|$, i.e., forming $\mathbf{S}^{-1}\mathbf{B}$.

3.3 Relationship with the clique expansion

Lastly we clarify how, when applied to the representative digraph of the EDVW hypergraph random walk, the above matrices are related to Laplacians of the hypergraph's clique expansion graph. This question was considered by Agarwal [2], who showed a number of other hypergraph Laplacians are equivalent to the graph Laplacians of the clique expansion.

As we've noted, an EDVW hypergraph random walk may be non-reversible, in which case there is no graph (including the clique expansion) with probability transition matrix equal to that of the hypergraph, as all graph random walks are necessarily reversible [3]. Consequently, it immediately follows that the hypergraph Laplacian matrix $\mathbf{I} - \mathbf{P}$ cannot be characterized as $\mathbf{I} - \mathbf{Q}$, where $\mathbf{Q}$ is the probability transition matrix of a random walk on a graph. Furthermore, we prove an analogous statement holds for Li and Zhang's digraph Laplacian matrix, $\boldsymbol{\Phi}^{1/2}(\mathbf{I} - \mathbf{P})\boldsymbol{\Phi}^{-1/2}$.

Proposition 3.1. *Let $\mathbf{P}$ denote the probability transition matrix of an EDVW random walk on a connected hypergraph. If the Markov chain given by $\mathbf{P}$ is non-reversible, there does not exist any edge-weighted graph G such that*

$$\boldsymbol{\Phi}^{1/2}(\mathbf{I} - \mathbf{P})\boldsymbol{\Phi}^{-1/2} = \boldsymbol{\Pi}^{1/2}(\mathbf{I} - \mathbf{Q})\boldsymbol{\Pi}^{-1/2}, \tag{8}$$

where $\mathbf{Q}$ denotes the probability transition matrix of a random walk on G, and $\boldsymbol{\Phi}, \boldsymbol{\Pi}$ denote diagonal matrices with the stationary distributions of $\mathbf{P}, \mathbf{Q}$, respectively, on the diagonal.

Proof. Assume such a graph exists. Then

$$\mathbf{P}_{ij} = \left(\sqrt{\phi_j\pi_i}/\sqrt{\phi_i\pi_j}\right)\mathbf{Q}_{ij}, \tag{9}$$

where $\phi_i = \boldsymbol{\Phi}_{ii}$ and $\pi_i = \boldsymbol{\Pi}_{ii}$. Since the random walk given by $\mathbf{P}$ is ergodic, Eqn. (9) implies random walk given by $\mathbf{Q}$ is ergodic as well. Furthermore, since all random walks on graphs are time reversible, the random walk given by $\mathbf{Q}$ is time reversible. Ergodic, time-reversible random walks satisfy Kolmogorov's criterion; applied to $\mathbf{Q}$, Kolmogorov's criterion states that for any set of vertices, $\{v_1, \ldots, v_n\}$, we have
$\mathbf{Q}_{v_1v_2}\mathbf{Q}_{v_2v_3}\cdots\mathbf{Q}_{v_{n-1}v_n}\mathbf{Q}_{v_nv_1} = \mathbf{Q}_{v_1v_n}\mathbf{Q}_{v_nv_{n-1}}\cdots\mathbf{Q}_{v_3v_2}\mathbf{Q}_{v_2v_1}$.
Now,

$$\begin{aligned}\mathbf{P}_{v_1v_2}\mathbf{P}_{v_2v_3}\ldots\mathbf{P}_{v_nv_1} &= \mathbf{Q}_{v_1v_2}\mathbf{Q}_{v_2v_3}\cdots\mathbf{Q}_{v_nv_1}\\ &= \mathbf{Q}_{v_1v_n}\mathbf{Q}_{v_nv_{n-1}}\cdots\mathbf{Q}_{v_2v_1} = \mathbf{P}_{v_1v_n}\mathbf{P}_{v_nv_{n-1}}\cdots\mathbf{P}_{v_2v_1}\end{aligned}$$

due to Eqn. (9), which implies $\mathbf{P}$ is the transition matrix of a time-reversible Markov chain, a contradiction. □

The above result means the EDVW random walked based hypergraph Laplacians $\mathbf{I} - \mathbf{P}$ and $\boldsymbol{\Phi}^{1/2}(\mathbf{I} - \mathbf{P})\boldsymbol{\Phi}^{-1/2}$ cannot be characterized as Laplacians of graphs, and hence are not subject to Agarwal's criticism of hypergraph Laplacians in [2]. However, Chung's digraph Laplacians symmetrize the probability transition matrix and, as has been previously noted in [16, 24], *can* be understood as Laplacians of weighted undirected graphs. In our context, this observation may be stated more precisely as follows:

Proposition 3.2. *Let* $\mathbf{L}$ *and* $\mathcal{L}$ *denote the hypergraph Laplacians obtained from applying Chung's digraph Laplacians defined in Eqns* (5) *and* (6)*, to the representative digraph of a random walk on a connected hypergraph* H*. The matrix*

$$\mathbf{A} = \frac{\mathbf{\Phi P} + \mathbf{P}^{\mathsf{T}}\mathbf{\Phi}}{2} \tag{10}$$

is the adjacency matrix of the clique expansion graph of H*, under an edge-weighting. Furthermore, the edge-weighted graph given by* $\mathbf{A}$ *has combinatorial and normalized graph Laplacian matrices equal to* $\mathbf{L}$ *and* $\mathcal{L}$*, respectively.*

Proof. To see $\mathbf{A}$ is the adjcency matrix of the clique expansion graph of H, observe that, by definition of an EDVW random walk on a hypergraph, $\mathbf{P}_{ij}$ (and hence $\mathbf{A}_{ij}$) is nonzero if and only if vertices $i, j \in e$, for some edge e in the hypergraph H. This is precisely the edge condition in the clique expansion definition. The combinatorial and normalized Laplacians of $\mathbf{A}$ are the same as $\mathbf{L}$ and $\mathcal{L}$, respectively, since $\mathbf{Ae} = \frac{1}{2}(\mathbf{\Phi}(\mathbf{Pe}) + \mathbf{P}^{\mathsf{T}}(\mathbf{\Phi e})) = \frac{1}{2}(\mathbf{\Phi e} + \mathbf{P}^{\mathsf{T}}\boldsymbol{\pi}) = \boldsymbol{\pi}$ and hence the weighted graph described by $\mathbf{A}$ has diagonal degree matrix $\mathbf{D} = \mathbf{\Phi}$. Substituting $\mathbf{D}$ and $\mathbf{A}$ into $\mathbf{D} - \mathbf{A}$ and $\mathbf{I} - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$ yields the result. □

In this sense, Agarwal's criticism in [2] also applies to hypergraph Laplacians derived from Chung's digraph Laplacians. However, as we will highlight further in the next section, Chung's digraph Laplacian matrices still preserve key information about the random walk important for clustering and hence serve as effective representations for our approach.

Lastly, we note Proposition 3.2 also answers a question of Chitra [8, Section 5.1] on whether there exist edge weights on the clique expansion such that its combinatorial Laplacian is "close" to the hypergraph Laplacian obtained from Chung's digraph combinatorial Laplacian. Indeed, the proposition shows there exists edge weights such that *equality* holds. Nonetheless, it remains to be seen whether a more explicit formula for these edge weights may be obtained solely in terms of the hypergraph's hyperedge and vertex weights, rather than invoking the stationary distribution, as in Eqn. (10).

4 PROPOSED CLUSTERING METHODS

Given a hypergraph $H = (V, E)$ with EDVW, and a desired number of clusters k, our goal is to partition V into disjoint subsets $S_1, \ldots, S_k$, such that a cluster quality objective function is optimized. Recall that in *graph* clustering, one such well known function is the normalized cut (Ncut), which measures the weight between each cluster S to its complement S^c relative to the cluster "volume". More precisely

$$\text{Ncut}(S_1, \ldots, S_k) = \frac{1}{2}\sum_{i=1}^{k} \frac{\text{vol}(\partial S_i)}{\text{vol}(S_i)}, \tag{11}$$

where $\text{vol}(S_i) = \sum_{u \in S_i}(\mathbf{Ae})_u$, $\text{vol}(\partial S_i) = \sum_{u \in S_i, v \in S_i^c} \mathbf{A}_{uv}$ and $\mathbf{A}$ is the adjacency matrix of the graph. However, in our case we are utilizing the representative digraph of the EDVW hypergraph random walk (and its associated Laplacians) as our basis for clustering. Thus, we require a notion of Ncut for directed graphs, which, in turn, requires directed notions of $\text{vol}(S)$ and $\text{vol}(\partial S)$. Chung proposed

Algorithm 1: Representative Digraph Clustering-Spec (RDC-Spec)

Input: A connected hypergraph $H = (V, E)$ with hyperedge weights and edge-dependent vertex weights, and desired number of clusters $k \geq 2$.

1 Construct $\mathbf{P}$ as in Eqn. (3).
2 Construct $\mathbf{\Phi} = \text{diag}(\boldsymbol{\pi})$ such that $\boldsymbol{\pi}^{\mathsf{T}}\mathbf{P} = \boldsymbol{\pi}$ and $\|\boldsymbol{\pi}\|_1 = 1, \boldsymbol{\pi} > 0$
3 Construct $\mathbf{T}$ as in Eqn. (15).
4 Compute the k eigenvectors paired with the k largest eigenvalues of $\mathbf{T}$ and collect them into the columns of the matrix $\mathbf{U} \in \mathbb{R}^{|V| \times k}$.
5 Normalize the rows of $\mathbf{U}$ with respect to the L_2 norm.
6 Cluster the rows of the matrix $\mathbf{U}$ using k-means.

Output: k vertex clusters

such digraph analogs of volume, which are based in the probability transition matrix $\mathbf{P}$ and stationary distribution $\boldsymbol{\pi}$:

$$\text{vol}(S_i) = \sum_{u \in S_i} \boldsymbol{\pi}_u, \tag{12}$$

$$\text{vol}(\partial S_i) = \sum_{u \in S_i, v \in S_i^c} \boldsymbol{\pi}_u \mathbf{P}_{uv}. \tag{13}$$

As shown in [16], we note $\text{vol}(\partial S) = \text{vol}(\partial S^c)$ and that the function $F_{\boldsymbol{\pi}} = \boldsymbol{\pi}_u \mathbf{P}_{uv}$ is an example of a *circulation* function, a general type of flow on the directed graph (see [10] for more).

We also note these directed notions of volume yield an elegant and intuitive random walk interpretation of Ncut, which was observed in the graph case by Meila and Shi [27]. In particular, if we let $\Pr(S^c|S)$ denote the probability of transitioning to a vertex in S^c given the current state is a vertex in S, then it is straightforward to show $\Pr(S^c|S) = \frac{\text{vol}(\partial S)}{\text{vol}(S)}$. By definition of Ncut, we thus have:

$$\text{Ncut}(S, S^c) = \Pr(S^c|S) + \Pr(S|S^c). \tag{14}$$

Now, returning our attention to Chung's digraph Laplacian matrices, recall Chung's digraph Laplacians are equivalent to the graph Laplacians associated with a particular edge-weighted graph derived from the digraph – that is, the graph with edge-weighted adjacency matrix defined in Eqn. (10). Further, Gleich showed $\text{vol}(S)$ and $\text{vol}(\partial S)$ of this graph are equal to their directed analogs (defined above) of the associated digraph [16, p. 7].

This means that any graph clustering algorithm which minimizes graph Ncut will, when applied to Chung's digraph Laplacians, minimize the directed analog of Ncut obtained by using the directed volume definitions in Eqns. (12) – (13) in the Ncut definition in Eqn. (11). We test two algorithms to obtain clusterings on Chung's Laplacian. The spectral method is motivated by Zhou et al.'s [32] algorithm which heuristically minimizes the Ncut on Chung's Laplacian and another method proposed by Ng, Jordan, and Weiss [29] which also heuristically minimizes the Ncut. Applying a similar algorithm to Chung's normalized digraph Laplacian yields our suggested hypergraph spectral clustering method, Algorithm 1. Additionally, we propose using a Symmetric Non-negative Matrix factorization based algorithm for graph clustering based on the framework from Kuang et al. [20, 21].

Algorithm 2: Representative Digraph Clustering-SymNMF (RDC-Sym)

Input: A connected hypergraph $H = (V, E)$ with hyperedge weights and edge-dependent vertex weights, and desired number of clusters $k \geq 2$.

1 Construct $\mathbf{P}$ as in Eqn. (3).
2 Construct $\mathbf{\Phi} = \text{diag}(\boldsymbol{\pi})$ such that $\boldsymbol{\pi}^{\mathsf{T}}\mathbf{P} = \boldsymbol{\pi}$ and $\|\boldsymbol{\pi}\|_1 = 1, \boldsymbol{\pi} > 0$
3 Construct $\mathbf{T}$ as in Eqn. (15).
4 Compute a rank k Symmetric NMF of $\mathbf{T}$ and collect the factors into the columns of the matrix $\mathbf{U} \in \mathbb{R}_{\geq 0}^{|V| \times k}$.
5 Assign each row of $\mathbf{U}$ to the column index of its max element. Use these as cluster assignments.

Output: k vertex clusters

When the hypergraph is disconnected, Algorithms 1 and 2 may be applied per connected component. We also note this algorithm utilizes the matrix $\mathbf{T}$,

$$\mathbf{T} = \mathbf{I} - \mathcal{L} = \frac{\mathbf{\Phi}^{\frac{1}{2}}\mathbf{P}\mathbf{\Phi}^{-\frac{1}{2}} + \mathbf{\Phi}^{-\frac{1}{2}}\mathbf{P}^{\mathsf{T}}\mathbf{\Phi}^{\frac{1}{2}}}{2} \tag{15}$$

which is slightly different from Chung's digraph Laplacian, which is $\mathcal{L} = \mathbf{I} - \mathbf{T}$. This modification is made to ensure the input is non-negative. This assumption is not necessary for spectral clustering, but necessary for non-negative matrix factorizations.

Our framework can be applied to multi modal data analysis frameworks that seek to utilize hypergraph information. For example Joint-NMF (JNMF), as proposed by Du et al. [14], is able to utilize multiple sources of information to perform clustering. This is further discussed in Section 6.3.2.

Symmetric Non-negative Matrix Factorization (SymNMF) solves

$$\min_{\mathbf{F} \geq 0} \|\mathbf{S} - \mathbf{F}\mathbf{F}^{\mathsf{T}}\|_F^2 \tag{16}$$

where $\mathbf{S}$ is a symmetric, non-negative matrix and $\mathbf{F} \in \mathbb{R}_{\geq 0}^{|V| \times r}$, r is some positive integer which is usually set to the number of clusters when clustering. It has been shown that SymNMF can achieve state-of-art results on various graph clustering tasks such as image segmentation [20, 21]. Additionally, Kuang et al. [21] show SymNMF and Spectral clustering minimize the same objective function but with different constraints. While SymNMF aims to solve Eqn. (16), spectral clustering aims to solve the same objective but imposes that $\mathbf{F}^{\mathsf{T}}\mathbf{F} = \mathbf{I}$ instead of $\mathbf{F} \geq 0$. While it depends on the algorithms utilized, SymNMF and Spectral Clustering have comparable complexities of (very) roughly $O(|V|^2 k)$ per iteration. SymNMF is dominated by the formation of the normal equations which requires the matrix products $\mathbf{F}^T\mathbf{F}$ and $\mathbf{F}^T\mathbf{S}$, whereas Spectral Clustering is dominated by the cost of computing a truncated spectral decomposition of $\mathbf{S}$. We adapt SymNMF clustering to our framework in Algorithm 2.

Lastly, we conclude this section by acknowledging other approaches that may be taken to cluster hypergraphs via the representative digraph of an EDVW hypergraph random walk. In particular, the aforementioned work by Cucuringu [12] takes a Stochastic Block Model (SBM) approach towards clustering digraphs. The SBMs are probabilistic models that generate random networks with planted communities; for more on SBMs and clustering, see [1, 22]. Cucuringu shows that, under mild assumptions on the parameters of the Directed Stochastic Block Model, the number of vertices misclassified by their algorithm is well-bounded, with high probability. This is shown by applying particular tools from random matrix theory, which critically rely upon both the Hermitian and skew-symmetry properties of the matrix. The complex-values in Cucuringu's input representation serve the purpose of allowing digraph edge-directionality to be encoded in a matrix with these properties. For details interested readers may refer to [12]. Although not explored further in this work, we note their algorithm may be applied to any edge-weighted digraph, and thus could be applied to the representative digraph of an EDVW hypergraph.

5 EXISTING HYPERGRAPH CLUSTERING METHODS

We've outlined an EDVW random-walk based framework for clustering hypergraphs that offers flexibility both in the choice of representation, as well as clustering method. Now, we will survey other clustering approaches that utilize different hypergraph representations or clustering methods than what we have proposed. Then, in Section 6, we will compare our clustering framework against these methods on text-document and other datasets. Before describing the details, it is helpful to take a broader viewpoint of hypergraph clustering approaches and briefly discuss how EDVW-based methods fit within this literature.

Much of the recent work on hypergraph clustering is fundamentally centered around the question of how hyperedges can be cut or alternatively how a vertex contributes to a hyperedge. In the context of the graph expansion-approaches, in which the aforementioned clique expansion is studied in place of the hypergraph, this question is answered by how the edges in the expansion graph are assigned weights. These weights are usually uniform with respect to a single hyperedge. Consequently, each vertex within a hyperedge is treated equally; for example, Zhou's Laplacian and hypergraph clustering formulation [33] is one such example.

Instead of clustering based on how hyperedges are cut, an alternative approach is to consider how network motifs (certain small subgraphs, such as a 3-clique) are cut. This approach was suggested by Benson [6] and further explored in [23]. In the motif-based clustering algorithms proposed in [6] no matter how a motif is cut it incurs a constant penalty. Later, the authors of [23] discussed the idea of inhomogeneous hypergraphs which can be thought of as motif clustering where different cuts in a motif incur different penalties. The cost of some of the different cuts are assumed to be given a priori: for example, if a hyperedge contains vertices a, b, c, d, then a weight for the cost of separating a, b and c, d is given. This information is then used to form each hyperedge into a clique that preserves cut constraints, yielding a clique expansion style method.

EDVW hypergraphs seek to address the same problem but from a different perspective. Instead of having a-priori penalties for edges or relationships between vertices within a hyperedge, for each hyperedge, EDVW give us data-driven values for how much each vertex individually contributes to that hyperedge. This information is then used to describe a random walk, which serves as the basis

for deriving representations utilized by our proposed clustering algorithms. Below, we survey other hypergraph clustering methods.

- Clique-expanded Hypergraph Clustering (CHC): Proposed by Zhou in [33], this algorithm expands each hyperedge in a given hypergraph into a clique and assigns a uniform weight value to each edge formed. Spectral clustering is then run on the resulting weighted graph which has the Laplacian

$$\mathbf{\Delta} = \mathbf{D}_V^{-1/2}\mathbf{X}^{\mathsf{T}}\mathbf{Z}\mathbf{D}_E^{-1}\mathbf{X}\mathbf{D}_V^{-1/2} \tag{17}$$

  where $\mathbf{X} \in \{0,1\}^{|E|\times|V|}$ is the incidence matrix, $\mathbf{Z} \in \mathbb{R}_+^{|E|\times|E|}$ is the diagonal matrix containing the weights of hyperedges, $\mathbf{D}_E = \text{diag}(\mathbf{Xe})$ and $\mathbf{D}_V$ is a diagonal matrix where the (i,i)th entry is $\mathbf{e}^{\mathsf{T}}\mathbf{Z}\mathbf{x}_i$ and $\mathbf{x}_i$ is the ith row of $\mathbf{X}$.
- NMF for Text-Clustering (NMF) [30] solves the problem

$$\min_{(\mathbf{U},\mathbf{M})\geq 0} \|\mathbf{R}^{\mathsf{T}} - \mathbf{U}\mathbf{M}^{\mathsf{T}}\|_F^2,$$

  $\mathbf{U} \in \mathbb{R}_{\geq 0}^{|V|\times k}$ and $\mathbf{M} \in \mathbb{R}_{\geq 0}^{|E|\times k}$, then column normalizes the document factor matrix to unit 2-norm and uses its max row indices to assign documents to clusters.
- K-Means (KM) runs the K-Means clustering algorithm to obtain a document clustering on the tf-idf matrix.
- Clique Random Walk Clustering (CRWC): In [33] Zhou proposed the uniform random walk discussed in Section 2.2. The probability transition matrix for this random walk is $\mathbf{P} = \mathbf{D}_V^{-1}\mathbf{X}^{\mathsf{T}}\mathbf{Z}\mathbf{D}_E^{-1}\mathbf{X}$. This matrix is fed into Algorithm 1 in place of the EDVW stochastic matrix on line 1. The matrices in this equation are the same as for CHC above. This method is included to assess the value of using EDVW vs EIVW.
- Spectral Bi-Clustering (SBC) [13] uses a weighted incidence matrix to obtain both edge and vertex clusterings. Following [13], we use the tf-idf matrix to cluster documents. An SVD is applied to a normalized tf-idf matrix $\mathbf{D}_1^{-1/2}\mathbf{R}\mathbf{D}_2^{-1/2}$ where $\mathbf{D}_1 = \text{diag}(\mathbf{Re})$ and $\mathbf{D}_2 = \text{diag}(\mathbf{R}^{\mathsf{T}}\mathbf{e})$. Then k-means is run on a set of truncated-normalized singular vectors.

Table 1 summarises various attributes in hypergraph clustering algorithms and indicates if each algorithm utilizes them. The attributes are 1) Random Walk, if an algorithm is based on a random walk formulation, 2) Spectral, if an algorithms uses the spectrum of a matrix to cluster, 3) EDVW, if an algorithm uses information from edge-dependant vertex weights, and 4) EIVW if an algorithm uses edge-independent vertex weights. A ✓ indicates a "has" or yes while a × indicates a "has not" or no.

Alg	Random Walk	Spectral	EDVW	EIVW
RDC-Spec	✓	✓	✓	×
RDC-Sym	✓	×	✓	×
CHC	✓	✓	×	✓
NMF	×	×	✓	×
KM	×	✓	✓	×
CRWC	✓	✓	×	✓
SBC	×	✓	✓	×

Table 1: Algorithm characteristics on whether random walks are used (Random Walk), eigenvalues are used to cluster (Spectral), and whether edge-dependent or edge-independent vertex weights are considered (EDVW, EIVW).

6 EXPERIMENTS

We test our proposed methods on a number of datasets. Most of these data sets come with ground truth allowing various metrics to be used to assess the output quality of an algorithm.

6.1 Data Sets and Preprocessing

We experiment on the following four data sets:

- 20-Newsgroups
- United States Patents Data
- Reuter's Corpus Volume 1
- Gene-Disease Data

Data	#-Vertices	#-Hyperedges	NNZ (Ind. Mtrx)	#-Clstrs
G1	1498	22755	.0049	4
G2	1545	19081	.0048	4
G3	1430	19412	.0048	4
G4	1945	20260	.0039	5
A22	835	4496	.0160	15
A42	965	4692	.0140	17
D02	744	4499	.0151	12
B06	688	4106	.0164	12
RCV1	9625	29969	.0023	4
GD	2261	12368	.0041	na

Table 2: Hypergraph data statistics. For text data, statistics are reported after pruning with a sparsity parameter of 0.2. The labels G1-G4, A22-B06, RCV1, and GD refer to 20Newsgroups, US Patents, Reuters Corpus Volume 1, and gene-disease datasets, respectively; see Section 6.3 for more.

For each text data set the documents are taken as vertices and the words are taken as hyperedges. The corresponding EDVWs are the tf-idf values. That is for the matrix $\mathbf{R}$, as in Eqn. (3), rows correspond to words and columns corresponding to documents. The entry in the wth row and dth column of $\mathbf{R}$ is the tf-idf value between the wth word and the dth document. Each hyperedge weight is computed as the standard deviation of a word (row of $\mathbf{R}$) in the tf-idf matrix and encoded in the matrix $\mathbf{W}$ as in Eqn. (3)[8]. A few preprocessing steps are applied to each dataset. Following [13], words that appear in over a certain fraction of the data-sets are removed, this fraction is determined experimentally and the best one is chosen for each algorithm. We refer to this parameter as the sparsity parameter. Similarly, words that only appear in a single document or that appear in no documents are removed. Every data set is checked to make sure it consists of a single connected component. Various information about each data set is listed in Table 2.

6.2 Clustering Metrics

In order to assess clustering quality on the 3 text data sets we compute the Normalized Mutual Information (NMI), average F1

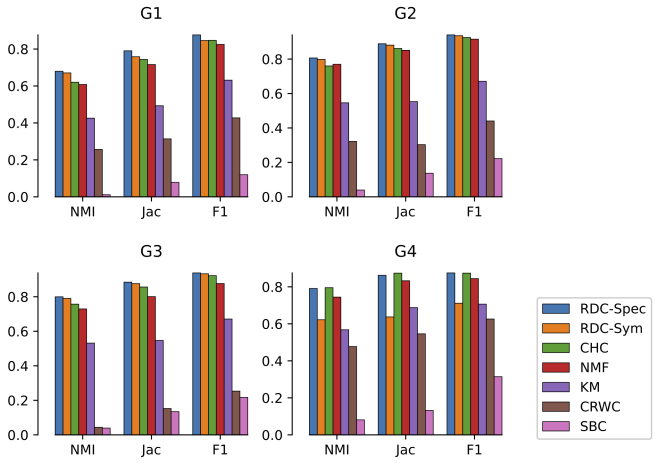

Figure 2: Clustering scores based on NMI, Jaccard index, and F1 score, for 4 subsets of the 20News data set.

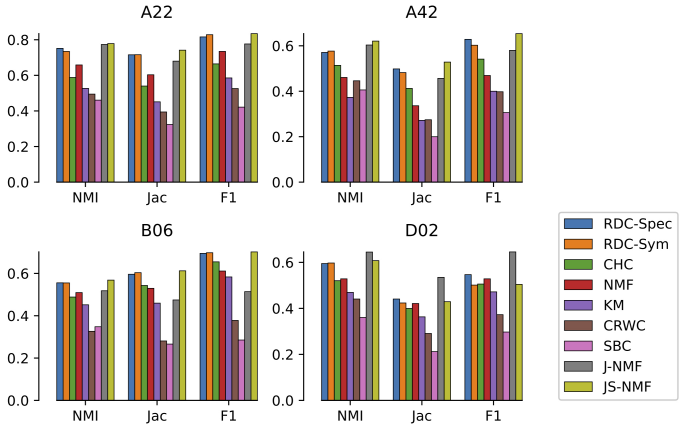

Figure 3: Clustering scores based on NMI, Jaccard index, and F1 score, for four US Patents data sets.

score, and Jaccard index. These metrics are based on comparing two clustering results represented by X and Y, which may be treated as two vectors of integer labels where $X_i = X_j$ only if the ith and jth ($i \neq j$) vertices are assigned to the same cluster. For our purposes one of X and Y is the ground truth cluster labels and the other is the predicted cluster labels from some clustering algorithm. Normalized mutual information (NMI) is given by $\frac{2 \cdot I(X,Y)}{H(X)+H(Y)}$, where $I(X,Y)$ is the mutual information and $H(X)$ is the entropy. The average F1 score is a generalization of the F1 score for multi-clusterings. $|X| \cup |Y|$ is the number of points the two assignments classify the same and $|X| \cap |Y|$ is $|X| + |Y| - |X \cup Y|$. For two clusters, the F1 score (see [31]) is $F_1(X,Y) = 2\frac{|X \cap Y|}{|X|+|Y|}$, and the Jaccard index is $J(X,Y) = \frac{|X \cap Y|}{|X \cup Y|}$. In order to handle multi-clustering, we use the Khun-Munkres algorithm to compute a matching between clusters to maximize the overall score for both the Jaccard and average F1 scores. This approach is similar to that used by Kuang et al. [20].

6.3 Results

6.3.1 20-Newsgroups. The first set of experiments was done on the 20-News[3] data set. This data set consists of 4 major categories each with a varying number of subcategories. We select 4 subsets of the subcategories to cluster on. The subcategories for each experiment are G1) OS Microsoft Windows, automobiles, cryptography, and politics-guns, G2) atheism, computer graphics, medicine, and Christianity G3) Windows X, motorcycles, space, religion-miscellaneous and G4) computer graphics, OS Microsoft Windows, IBM PC hardware, MAC hardware, and Windows X. The first three groups are expected to be well separated while the last is expected to present a more challenging clustering problem. All clusters have between 318 and 398 documents. Some basic statistics from the 20-News data set subsets we use are given in Table 2 for G1-G4. These hypergraph statistics are all from using a pruning parameter of 0.2 which gives representative results for all algorithms. From Figure 2 we observe that our framework gives competitive results for all subsets of the 20-News data.

6.3.2 US-Patents. This data was originally processed by Du et al. [14] from the Patents View[4] website and contains word count and citation information for a number of patents claims for 13 different categories (denoted A22, B06, etc.) each with multiple sub-classes. These sub-classes are used as ground truth and only patents belonging to a single sub-class are used. Finally, only sub-classes with 40 or more patents are kept. For our experiments we selected the categories A22, A42, D02, and B06 as these each have a sufficient number of patents that belong to only a single sub-class.

As this data also contains information on which patents reference each other, this additional citation information may be utilized via Joint-NMF (J-NMF) style algorithms, which solve

$$\min_{\{\mathbf{M},\mathbf{Z},\tilde{\mathbf{M}}\} \geq 0} \|\mathbf{X} - \mathbf{Z}\mathbf{M}^{\mathsf{T}}\|_F^2 + \gamma \|\mathbf{S} - \mathbf{M}\tilde{\mathbf{M}}^{\mathsf{T}}|_F^2 + \beta \|\mathbf{M} - \tilde{\mathbf{M}}\|_F^2 \quad (18)$$

where $\mathbf{X} \in \mathbb{R}_{\geq 0}^{|E| \times |V|}$ (here, $\mathbf{X}$ does not necessarily refer to a 0,1 incidence matrix) and $\mathbf{S} \in \mathbb{R}_{\geq 0}^{|V| \times |V|}$ and $\{\beta, \gamma\} \geq 0$ are some weighting parameters. The matrix $\tilde{\mathbf{M}}$ is then used to obtain clusters similar to the standard NMF clustering procedure. Note that if $\beta = \gamma = 0$ then a standard NMF objective is recovered. For our experiments we set γ and β as recommended by Du et al. For this data set $\mathbf{X}$ is set to the EDVW incidence matrix $\mathbf{R}$ and $\mathbf{S}$ is a symmetric, 0, 1 matrix indicating if two patents cite each other. Additionally, we adapt J-NMF to utilize Chung's Laplacian via the matrix $\mathbf{T}$, eq. (15), and decompose two symmetric matrices. We refer to this as Joint-Symmetric NMF (JS-NMF), the new objective is given in Eqn. (19).

$$\min_{\{\mathbf{M},\hat{\mathbf{M}},\tilde{\mathbf{M}}\} \geq 0} \|\mathbf{C} - \mathbf{M}\hat{\mathbf{M}}^{\mathsf{T}}\|_F^2 + \alpha \|\mathbf{M} - \hat{\mathbf{M}}\|_F^2 + \gamma \|\mathbf{S} - \mathbf{M}\tilde{\mathbf{M}}^{\mathsf{T}}|_F^2 + \beta \|\mathbf{M} - \tilde{\mathbf{M}}\|_F^2 \quad (19)$$

where $\mathbf{C} \in \mathbb{R}_{\geq 0}^{|V| \times |V|}$ is $\mathbf{T}$ and $\mathbf{S}$ is the patent citation data as in Eqn. (18). We note that JS-NMF performs the best out of all algorithms for 3 of the 4 US-Patent groups, as can be seen in Figure 3. Due to the enforcement of symmetry for both norms in Eqn. (19), there is

[3] http://qwone.com/ jason/20Newsgroups/

[4] www.patentsview.org

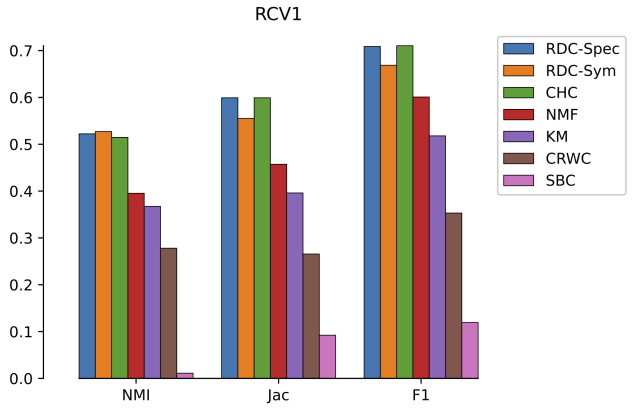

Figure 4: Clustering scores based on the measures of NMI, Jaccard index, and F1 score, for RCV1 Data.

Graph	RDC-Spec	RDC-Sym	CHC
ANC-**T**	0.1470	0.1177	0.1953
ANC-**Δ**	0.2541	0.2218	0.3019
ACo-**T**	0.2999	0.2479	0.3906
ACo-**Δ**	0.5193	0.4654	0.6038

Table 3: Average-Normalized cut (ANC) and Average-Conductance (ACo) values for RDC-Spec, RDC-Sym, and CHC on matrices T and Δ with 15 clusters

Graph	RDC-Spec	RDC-Sym	CHC
ANC-**T**	0.1681	0.1331	0.2059
ANC-**Δ**	0.2879	0.2492	0.3133
ACo-**T**	0.1681	0.1363	0.2059
ACo-**Δ**	0.2879	0.2558	0.3133

Table 4: Average-Normalized cut (ANC) and Average-Conductance (ACo) values for RDC-Spec, RDC-Sym, and CHC on matrices T and Δ with 25 clusters.

an additional parameter α, similar to γ and β, that must be chosen. For fairness we attempt to generalize Du et al.'s recommendation for parameter setting but believe performance could further be improved if more effort were put into selecting these parameters. Results are visualized in Figure 3.

6.3.3 RCV1. The Reuters Corpus Volume 1 (RCV1)[5] data set is a collection of newswire stories. Table 2 lists some basic statistics for this data. This is the largest data set we run on and we observe a similar trend to the two previous experiments. RDC-Spec and RDC-Sym perform competitively with CHC, with scores separated by very slim margins. The scores for each clustering quality metric are visualized in Figure 4.

6.3.4 Gene-Disease Data. This dataset consists of collections of genes associated with human diseases, taken from DisGeNET[6]. This may be modeled as a hypergraph in which vertices are diseases, and genes are hyperedges. Table 2 presents basic statistics for this data. For each disease-gene pair, DisGeNET computes a "Gene Disease Association" (GDA) score between 0 and 1, which is based on the number of and types of sources supporting that disease-gene association. Higher values indicate stronger associations; see DisGeNET's documentation[7] for full details. For our purposes, GDA scores serve naturally as EDVW for the disease-gene hypergraph.

Since this dataset lacks ground truth clusters, we turn to other clustering quality metrics. Du et al. in [15] report the average-Ncut value obtained by various clustering algorithms on a number of data sets. This technique is difficult to apply directly to our work where different algorithms are based on different representations of a hypergraph: different representations assign different edge weights which affect the Ncut values. To this end we run RDC-Spec, RDC-Sym, and CHC on the Gene-Disease hypergraph and compute the average-Ncut and average-conductance, Eqns. (20) and (21), of each clustering over 10 different sparsifications of the hypergraph. These 3 algorithms are chosen as they tend to achieve consistently high scores on the previous data sets. Each sparsified input data set is clustered 10 times and the average of the average-normalized cut and average-conductance is computed. We report the best scores for each algorithm.

$$\text{Av-Ncut}(S_1, \cdots, S_k) = \frac{1}{2k} \sum_{i=1}^{k} \frac{\text{vol}(\partial S_i)}{\text{vol}(S_i)} \tag{20}$$

$$\text{Av-Cond}(S_1, \cdots, S_k) = \frac{1}{2k} \sum_{i=1}^{k} \frac{\text{vol}(\partial S_i)}{\min\{\text{vol}(S_i), \text{vol}(S_i^c)\}} \tag{21}$$

Additionally, for every pair of algorithms that are based on two different hypergraph representations, each algorithm is run on the other algorithm's hypergraph representation, producing two different scores. For example, the clustering yielded by RDC-Spec operating on **T**, Eqn. (15), is taken and its average-Ncut and average-conductance values are computed on the matrix **Δ**, Eqn. (17). The same is done for RDC-Sym and the reverse is done for CHC. Note these matrices have the same non-zero pattern. Therefore differences in cuts values and clusterings are due to edge-weightings. Tables 3 and 4 present the results for 15 and 25 clusters respectively. These cluster sizes were chosen arbitrarily. RDC-Spec and RDC-Sym achieve lower cut and conductance scores than CHC on both graphs.

7 CONCLUSION

We presented a flexible framework for clustering a hypergraph and showed that edge-dependent vertex weights represents the information in a hypergraph well in the context of clustering. As recently proposed [8], these weights may be utilized to define hypergraph random walks, which naturally yield a number of different hypergraph Laplacians via the representative digraph of the random walk. Focusing on Chung's normalized digraph Laplacian, we explained its effectiveness as a representation of a hypergraph for clustering based on its relationship to a normalized cut criterion, and proposed a suite of clustering algorithms that utilize this input.

[5] http://www.daviddlewis.com/resources/testcollections/rcv1/
[6] https://www.disgenet.org/
[7] https://www.disgenet.org/dbinfo

We demonstrated the viability of our frameworks in comparison to other methods through experiments on 3 text datasets with ground truth, and on a gene-disease relation data set via well-known partition quality metrics. We found algorithms utilizing the proposed hypergraph Laplacian performed consistently well and frequently better than other methods.

Many directions remain for future work. First, among the hypergraph Laplacians considered in Section 3.2, we only utilized Chung's normalized digraph Laplacian in our experiments, although other Laplacians can be derived from the representative digraph. It would be interesting to explore whether these Laplacians are effective representations for hypergraph clustering – particularly Li and Zhang's asymmetric digraph Laplacian [24], which we studied in Proposition 3.1, as well as complex-valued digraph matrices utilized in recent clustering work [12]. Additionally, we observed better performance from joint methods that utilize multiple representations and many other combinations here can be explored [14].

Second, rather than utilize EDVW random walks to form Laplacians for clustering, an alternative approach is to use measures associated with the random walk itself, such as hitting and commute times. Such parameters serve as relational measures between pairs of vertices; for instance, hitting times measure the expected number of steps until one vertex is reached from another, and have been utilized in digraph clustering schemes [7]. In the context of EDVW hypergraph random walks, such metrics may be of similar use for hypergraph clustering.

ACKNOWLEDGMENTS

This work was partially funded under the High Performance Data Analytics (HPDA) program at the Department of Energy's Pacific Northwest National Laboratory. PNNL Information Release: PNNL-SA-153213. Pacific Northwest National Laboratory is operated by Battelle Memorial Institute under Contract DE-ACO6-76RL01830. Koby Hayashi acknowledges support from the United States Department of Energy through the Computational Sciences Graduate Fellowship (DOE CSGF) under grant number: DE-SC0020347.

REFERENCES

[1] Emmanuel Abbe. 2017. Community detection and stochastic block models: recent developments. *The Journal of Machine Learning Research* 18, 1 (2017), 6446–6531.

[2] Sameer Agarwal, Kristin Branson, and Serge Belongie. 2006. Higher Order Learning with Graphs. In *Proceedings of the 23rd International Conference on Machine Learning* (Pittsburgh, Pennsylvania, USA) *(ICML '06)*. ACM, New York, NY, USA, 17–24. https://doi.org/10.1145/1143844.1143847

[3] David Aldous and James Fill. 1995. Reversible Markov chains and random walks on graphs.

[4] Chen Avin, Yuval Lando, and Zvi Lotker. 2014. Radio cover time in hyper-graphs. *Ad Hoc Networks* 12 (jan 2014), 278–290. https://doi.org/10.1016/j.adhoc.2012.08.010

[5] Frank Bauer. 2012. Normalized graph Laplacians for directed graphs. *Linear Algebra Appl.* 436, 11 (jun 2012), 4193–4222. https://doi.org/10.1016/j.laa.2012.01.020

[6] Austin R. Benson, David F. Gleich, and Jure Leskovec. 2016. Higher-order organization of complex networks. *CoRR* abs/1612.08447 (2016). arXiv:1612.08447 http://arxiv.org/abs/1612.08447

[7] Mo Chen, Jianzhuang Liu, and Xiaoou Tang. 2008. Clustering via Random Walk Hitting Time on Directed Graphs.. In *AAAI*, Vol. 8. 616–621.

[8] Uthsav Chitra and Benjamin J. Raphael. 2019. Random Walks on Hypergraphs with Edge-Dependent Vertex Weights. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*. 1172–1181. http://proceedings.mlr.press/v97/chitra19a.html

[9] Fan Chung. 1997. *Spectral graph theory*. Number 92. American Mathematical Soc.

[10] F. Chung. 2005. Laplacians and the Cheeger Inequality for Directed Graphs. 9 (2005), 1âĂŞ19. https://doi.org/10.1007/s00026-005-0237-z

[11] Colin Cooper, Alan Frieze, and Tomasz Radzik. 2013. The cover times of random walks on random uniform hypergraphs. *Theoretical Computer Science* 509 (oct 2013), 51–69. https://doi.org/10.1016/j.tcs.2013.01.020

[12] Mihai Cucuringu, Huan Li, He Sun, and Luca Zanetti. [n.d.]. Hermitian matrices for clustering directed graphs: insights and applications. *arXiv:1908.02096* ([n. d.]). arXiv:cs.LG/1908.02096v1

[13] Inderjit Dhillon. 2001. Co-clustering Documents and Words Using Bipartite Spectral Graph Partitioning. *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (05 2001). https://doi.org/10.1145/502512.502550

[14] Rundong Du, Barry L. Drake, and Haesun Park. 2017. Hybrid Clustering based on Content and Connection Structure using Joint Nonnegative Matrix Factorization. *CoRR* abs/1703.09646 (2017). arXiv:1703.09646 http://arxiv.org/abs/1703.09646

[15] Rundong Du, Da Kuang, Barry Drake, and Haesun Park. 2017. Hierarchical Community Detection via Rank-2 Symmetric Nonnegative Matrix Factorization. 4 (12 2017), 1 – 26. https://doi.org/10.1186/s40649-017-0043-5

[16] D. Gleich. 2006. Hierarchical directed spectral graph partitioning. (2006).

[17] Krystal Guo and Bojan Mohar. 2016. Hermitian Adjacency Matrix of Digraphs and Mixed Graphs. *Journal of Graph Theory* 85, 1 (jun 2016), 217–248. https://doi.org/10.1002/jgt.22057

[18] Paul W. Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. 1983. Stochastic blockmodels: First steps. *Social Networks* 5, 2 (jun 1983), 109–137. https://doi.org/10.1016/0378-8733(83)90021-7

[19] Steve Kirkland. 2017. Two-mode networks exhibiting data loss. *Journal of Complex Networks* 6, 2 (aug 2017), 297–316. https://doi.org/10.1093/comnet/cnx039

[20] Da Kuang, Chris Ding, and Haesun Park. [n.d.]. Symmetric Nonnegative Matrix Factorization for Graph Clustering.

[21] Da Kuang, Sangwoon Yun, and Haesun Park. 2015. SymNMF: Nonnegative low-rank approximation of a similarity matrix for graph clustering. *Journal of Global Optimization* 62 (07 2015). https://doi.org/10.1007/s10898-014-0247-2

[22] Jing Lei and Alessandro Rinaldo. 2015. Consistency of spectral clustering in stochastic block models. *The Annals of Statistics* 43, 1 (feb 2015), 215–237. https://doi.org/10.1214/14-aos1274

[23] Pan Li and Olgica Milenkovic. 2017. Inhomogeneous Hypergraph Clustering with Applications. arXiv:cs.LG/1709.01249

[24] Yanhua Li and Zhi-Li Zhang. 2010. Random walks on digraphs, the generalized digraph laplacian and the degree of asymmetry. In *International Workshop on Algorithms and Models for the Web-Graph*. Springer, 74–85.

[25] Ying Liu, Jiabin Yuan, Bojia Duan, and Dan Li. 2018. Quantum walks on regular uniform hypergraphs. *Scientific Reports* 8, 1 (jun 2018). https://doi.org/10.1038/s41598-018-27825-z

[26] Linyuan Lu and Xing Peng. 2013. High-Order Random Walks and Generalized Laplacians on Hypergraphs. *Internet Mathematics* 9, 1 (jan 2013), 3–32. https://doi.org/10.1080/15427951.2012.678151

[27] Marina Maila and Jianbo Shi. 2001. A Random Walks View of Spectral Segmentation. In *Proceedings of AI and STATISTICS (AISTATS) 2001.*

[28] Bojan Mohar. 2020. A new kind of Hermitian matrices for digraphs. *Linear Algebra Appl.* 584 (jan 2020), 343–352. https://doi.org/10.1016/j.laa.2019.09.024

[29] Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. 2001. On Spectral Clustering: Analysis and an Algorithm. In *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic* (Vancouver, British Columbia, Canada) *(NIPSâĂŹ01)*. MIT Press, Cambridge, MA, USA, 849âĂŞ856.

[30] Wei Xu, Xin Liu, and Yihong Gong. 2003. Document Clustering Based on Non-Negative Matrix Factorization. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval* (Toronto, Canada) *(SIGIR âĂŹ03)*. Association for Computing Machinery, New York, NY, USA, 267âĂŞ273. https://doi.org/10.1145/860435.860485

[31] Jaewon Yang and Jure Leskovec. 2013. Overlapping Community Detection at Scale: A Nonnegative Matrix Factorization Approach. In *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining* (Rome, Italy) *(WSDM âĂŹ13)*. Association for Computing Machinery, New York, NY, USA, 587âĂŞ596. https://doi.org/10.1145/2433396.2433471

[32] Dengyong Zhou, Jiayuan Huang, and Bernhard Schoelkopf. 2005. Learning from Labeled and Unlabeled Data on a Directed Graph. *Framework* (01 2005). https://doi.org/10.1145/1102351.1102482

[33] Dengyong Zhou, Jiayuan Huang, and Bernhard Schölkopf. 2007. Learning with Hypergraphs: Clustering, Classification, and Embedding. In *Advances in Neural Information Processing Systems 19*, B. Schölkopf, J. C. Platt, and T. Hoffman (Eds.). MIT Press, 1601–1608. http://papers.nips.cc/paper/3128-learning-with-hypergraphs-clustering-classification-and-embedding.pdf