Higher Order Learning with Graphs

Sameer Agarwal   SAGARWAL@CS.UCSD.EDU
Kristin Branson   KBRANSON@CS.UCSD.EDU
Serge Belongie   SJB@CS.UCSD.EDU
Department of Computer Science and Engineering, University of California San Diego, La Jolla, CA 92093, USA

Abstract

Recently there has been considerable interest in learning with higher order relations (i.e., three-way or higher) in the unsupervised and semi-supervised settings. Hypergraphs and tensors have been proposed as the natural way of representing these relations and their corresponding algebra as the natural tools for operating on them. In this paper we argue that hypergraphs are not a natural representation for higher order relations, indeed pairwise as well as higher order relations can be handled using graphs. We show that various formulations of the semi-supervised and the unsupervised learning problem on hypergraphs result in the same graph theoretic problem and can be analyzed using existing tools.

1. Introduction

Given a data set, it is common practice to represent the similarity relation between its elements using a weighted graph. A number of machine learning methods for unsupervised and semi-supervised learning can then be formulated in terms of operations on this graph. In some cases like spectral clustering, the relation between the structural and the spectral properties of the graph can be exploited to construct matrix theoretic methods that are also graph theoretic. The most commonly used matrix in these methods is the Laplacian of the graph (Chung, 1997). In the same manner that the Laplace-Beltrami operator is used to analyze the geometry of continuous manifolds, the Laplacian of a graph is used to study the structure of the graph and functions defined on it.

A fundamental constraint in this formulation is the

assumption that it is possible to measure similarity between *pairs* of points. Consider a k-lines algorithm which clusters points in $\mathbb{R}^d$ into k clusters where elements in each cluster are well-approximated by a line. As every pair of data points trivially define a line, there is no useful measure of similarity between pairs of points for this problem. However, it is possible to define measures of similarity over triplets of points that indicate how close they are to being collinear. This analogy can be extended to any model-based clustering task where the fitting error of a set of points to a model can be considered a measure of the dissimilarity among them. We refer to similarity/dissimilarity measured over triples or more of points as higher order relations.

A number of questions that have been addressed in domains with pairwise relations can now be asked for the case of higher order relations. How does one perform clustering in such a domain? How can one formulate and solve the semi-supervised and unsupervised learning problems in this setting?

Hypergraphs are a generalization of graphs in which the edges are arbitrary non-empty subsets of the vertex set. Instead of having edges between pairs of vertices, hypergraphs have edges that connect sets of two or more vertices. While our understanding of hypergraph spectral methods relative to that of graphs is very limited, a number of authors have considered extensions of spectral graph theoretic methods to hypergraphs.

Another possible representation of higher order relations is a tensor. Tensors are a generalization of matrices to higher dimensional arrays, and they can be analyzed with multilinear algebra.

Recently a number of authors have considered the problem of unsupervised and semi-supervised learning in domains with higher order relations (Agarwal et al., 2005; Govindu, 2005; Shashua & Hazan, 2005; Shashua et al., 2006; Zhou et al., 2005). The success of graph and matrix theoretic representations have prompted researchers to extend these representations to the case

of higher order relations.

In this paper we focus on spectral graph and hypergraph theoretic methods for learning with higher order relations. We survey a number of approaches from machine learning, VLSI CAD and graph theory that have been proposed for analyzing the structure of hypergraphs. We show that despite significant differences in how previous authors have approached the problem, there are two basic graph constructions that underlie all these studies. Furthermore, we show that these constructions are essentially the same when viewed through the lens of the normalized Laplacian.

The paper is organized as follows. Section 2 defines our notation. Section 3 reviews the properties of the graph Laplacian from a machine learning perspective. Section 4 considers the algebraic generalization of Laplacian to higher order structures and shows why it is not useful for machine learning tasks. Section 5 presents a survey of graph constructions and linear operators related to hypergraphs that various studies have used for analyzing the structure of hypergraphs and for unsupervised and semi-supervised learning. In Section 6 we show how all these constructions can be reduced to two graph constructions and their associated Laplacians. Finally, in Section 7 we conclude with a summary and discussion of key results.

2. Notation

Let $G(V, E)$ denote a hypergraph with vertex set V and edge set E. The edges are arbitrary subsets of V with weight $w(e)$ associated with edge e. The degree $d(v)$ of a vertex is $d(v) = \sum_{e \in E | v \in e} w(e)$. The degree of an edge e is denoted by $\delta(e) = |e|$. For k-uniform hypergraphs, the degrees of each edge are the same, $\delta(e) = k$. In particular, for the case of ordinary graphs or "2-graphs," $\delta(e) = 2$. The vertex-edge incidence matrix H is $|V| \times |E|$ where the entry $h(v, e)$ is 1 if $v \in e$ and 0 otherwise. By these definitions, we have:

$$d(v) = \sum_{e \in E} w(e)h(v, e) \quad \text{and} \quad \delta(e) = \sum_{v \in V} h(v, e) \quad (1)$$

D_e and D_v are the diagonal matrices consisting of edge and vertex degrees, respectively. W is the diagonal matrix of edge weights, $w(\cdot)$. A number of different symbols have been used in the literature to denote the Laplacian of graph. We follow the convention in (Chung, 1997) and use L for the combinatorial Laplacian and $\mathcal{L}$ for the normalized Laplacian. L is also known as the unnormalized Laplacian of a graph and is usually written as

$$L = D_v - S \quad (2)$$

where S is the $|V| \times |V|$ adjacency matrix with entry (u, v) equal to the weight of the edge (u, v) if they are connected, 0 otherwise. An important variant is the normalized Laplacian,

$$\mathcal{L} = I - D_v^{-1/2} S D_v^{-1/2} \quad (3)$$

For future reference it is useful to rewrite the above expressions in terms of the vertex-edge incidence relation

$$L = 2D_v - HWH^\top \quad (4)$$

$$\mathcal{L} = I - \frac{1}{2} D_v^{-1/2} HWH^\top D_v^{-1/2} \quad (5)$$

3. The Graph Laplacian

The graph Laplacian is the discrete analog of the Laplace-Beltrami operator on compact Riemannian manifolds (Belkin & Niyogi, 2003; Rosenberg, 1997; Chung, 1997). It has been used extensively in machine learning, initially for the unsupervised case and then recently for semi-supervised learning (Zhu, 2005). In this section we highlight some of the properties of the Laplacian from a machine learning perspective, and motivate the search for similar operators on hypergraphs.

Perhaps the earliest use of the graph Laplacian was the development of spectral clustering algorithms which considered continuous relaxations of graph partitioning problems (Alpert et al., 1999; Shi & Malik, 2000; Ng et al., 2002). The relaxation converted the optimization problem into a generalized eigenvalue problems involving the Laplacian matrix of the graph.

In (Zhou & Schölkopf, 2005), the authors develop a discrete calculus on 2-graphs by treating them as discrete analogs of compact Riemannian manifolds. As one of the consequences of this development they argue that, in analogy to the continuous case, the graph Laplacian be defined as an operator $L : \mathcal{H}(V) \to \mathcal{H}(V)$,

$$Lf := \frac{1}{2} \operatorname{div}(\nabla f) \quad (6)$$

Zhou et al. also argue that there exists a family of regularization operators on the 2-graphs, the Laplacian being one of them, that can be used for transduction, i.e., given a partial labeling of the graph vertices y, use the geometric structure of the graph to induce a labeling f on the unlabeled vertices. The vertex label $y(v)$ is $+1, -1$ for positive and negative valued examples, respectively, and 0 if no information is available about the label. They consider the regularized least squares problem

$$\arg \min_f \left(\langle f, Lf \rangle + \mu \|f - y\|_2^2 \right) \quad (7)$$

While the discrete version of the problem where $f(v) \in \{+1, -1\}$ is a hard combinatorial problem, relaxing the range of f to the real line $\mathbb{R}$ results in a simple linear least squares problem, solved as $f = \mu(\mu I + \Delta)^{-1}y$. A similar formulation is considered by (Belkin & Niyogi, 2003). In the absence of any labels, the problem reduces to that of clustering, and the eigenmodes of the Laplacian are used to label the vertices (Shi & Malik, 2000; Ng et al., 2002).

More generally, in (Smola & Kondor, 2003), the authors prove that just as the continuous Laplacian is the unique linear second order self adjoint operator invariant under the action of rotation operators, the same is true for the Laplacian and the unnormalized Laplacian with the group of rotations replaced by group of permutations.

A number of successful regularizers in the continuous domain can be written as $\langle f, r(L)f \rangle$ where L is the continuous Laplacian, f is the model and r is non-decreasing scalar function that operates on the spectrum of Δ. Smola and Kondor show that the same can be shown for a variety of regularization operators on graphs.

4. Higher Order Laplacians

In light of the previous section, it is interesting to consider generalizations of the Laplacian to higher order structures. We now present a brief look at the algebro-geometric view of the Laplacian, and how it leads to the generalization of the combinatorial Laplacian for hypergraphs. For simplicity of exposition, we will consider the unweighted case. For a more formal presentation of the material in this section, we refer the reader to (Munkres, 1984; Chung, 1993; Chung, 1997; Forman, 2003).

Let us assume that a graph represents points in some abstract space with the edges representing lines connecting these points and the weights on the edge having an inverse relation to the length of the line. The Laplacian then measures how smoothly a function defined on these points (vertices) changes with respect to their relative arrangement. As we saw earlier, the quadratic form $f^\top Lf$ does this for the vertex function f. This view of a graph and its Laplacian can be generalized to hypergraphs. A hypergraph represents points in some abstract space where each hyperedge corresponds to a simplex in that space with the vertices of the hyperedge as its corners. The weight on the hyperedge is inversely related to the *size* of the simplex. Now we are not restricted to define functions on just vertices, we can define functions on sets of vertices, corresponding to

lines, triangles, etc. Algebraic topologists refer to these functions as p-chains, where p is size of the simplices on which they are defined. Thus vertex functions are 0-chains, edge functions are 1-chains, and so on. In each case one can ask the question, how does one measure the variation in these functions with respect to the geometry of the hypergraph or its corresponding simplex?

Let us take a second look at the graph Laplacian. As the graph Laplacian is a positive semidefinite operator, it can be written as

$$L = BB^\top \qquad (8)$$

Here, B is a $|V| \times |E|$ matrix such that (u, v)th column contains $+1$ and -1 in rows u and v, respectively. The exact ordering does not matter. B is called the boundary operator ∂_1 that maps on 1-chains (edges) to 0-chains and $B^\top$ is the co-boundary operator that maps 0-chains to 1-chains. Note that B is different from H; although H is also a vertex-edge incidence matrix, all of its entries are non-negative. We can rewrite

$$f^\top Lf = f^\top BB^\top f = \|B^\top f\|_2^2. \qquad (9)$$

Thus $f^\top Lf$ is the squared norm of a vector of size $|E|$, whose entries are the change in the vertex function or the 0-chain along an edge. This is a particular case of the general definition of the pth Laplacian operator on p-chains, given by

$$L_p = \partial_{p+1}\partial_{p+1}^\top + \partial_p^\top \partial_p \qquad (10)$$

Symbolically, this is exactly the same as the Laplace operator on p-forms on a Riemannian manifold (Rosenberg, 1997). For the case of hypergraphs or simplicial complexes, we interpret this as the operator that measures variations on functions defined on p-sized subsets of the vertex set (p-chains). It does so by considering the change in the chain with respect to simplices of size $p+1$ and $p-1$. For the case of the ordinary graph, we only consider the first term in the above expression since vertex functions are 0-chains, and there are no -1 sized simplices. It is however possible to consider 1-chains or functions defined on edges of the graphs and measure their variation using the edge Laplacian, given by $L_1 = B^\top B$. In light of this, the usual Laplacian on the graph is the L_0 or vertex Laplacian. In (Chung, 1993) the Laplacian for the particular case of the k-uniform hypergraph is presented. A more elaborate discussion of the construction of various kinds of Laplacians on simplicial complexes and their uses is described in (Forman, 2003).

Unfortunately, while geometrically and algebraically these constructions extend the graph Laplacian to hypergraphs, it is not clear how one can use them in

machine learning. The fundamental object we are interested in is a vertex function or 0-chain, thus the linear operator we are looking for should operate on 0-chains. Notice, however, that a pth order Laplacian only considers p-chains, and the structure of the Laplacian depends on the incidence relations between $p-1$, p and $p+1$ simplices. To operate on vertex functions, one needs a vertex Laplacian, which unfortunately only considers the incidence of 0-chains with 1-chains. Thus the vertex Laplacian for a k-uniform hypergraph will not consider any hyperedges, rendering it useless for the purposes of studying vertex functions. Indeed the Laplacian on a 3-uniform graph operates on 2-chains, functions defined on all pairs of vertices (Chung, 1993).

5. Hypergraph Learning Algorithms

A number of existing methods for learning from a hypergraph representation of data first construct a graph representation using the structure of the initial hypergraph. Then, they project the data onto the eigenvectors of the combinatorial or normalized graph Laplacian. Other methods define a hypergraph "Laplacian" using analogies from the graph Laplacian. These methods show that the eigenvectors of their Laplacians are useful for learning, and that there is a relationship between their hypergraph Laplacians and the structure of the hypergraph. In this section, we review these methods. In the next section, we compare these methods analytically.

5.1. Clique Expansion

The clique expansion algorithm constructs a graph $G^x(V, E^x \subseteq V^2)$ from the original hypergraph $G(V, E)$ by replacing each hyperedge $e = (u_1, ..., u_{\delta(e)}) \in E$ with an edge for each pair of vertices in the hyperedge (Zien et al., 1999): $E^x = \{(u, v) : u, v \in e, e \in E\}$.

Note that the vertices in hyperedge e form a clique in the graph G^x. The edge weight $w^x(u, v)$ minimizes the difference between the weight of the graph edge and the weight of each hyperedge e that contains both u and v:

$$w^x(u,v) = \arg\min_{w^x(u,v)} \sum_{e \in E: u,v \in e} (w^x(u,v) - w(e))^2 \quad (11)$$

Thus, clique expansion uses the discriminative model that every edge in the clique of G^x associated with hyperedge e has weight $w(e)$. The minimizer of this criterion is simply

$$w^x(u,v) = \mu \sum_{e \in E: u,v \in e} w(e) = \mu \sum_e h(u,e)h(v,e)w(e). \quad (12)$$

Here μ is a fixed scalar. The combinatorial or normalized Laplacian of the constructed graph G^x is then used to partition the vertices.

5.2. Star Expansion

The star expansion algorithm constructs a graph $G^*(V^*, E^*)$ from hypergraph $G(V, E)$ by introducing a new vertex for every hyperedge $e \in E$, thus $V^* = V \cup E$ (Zien et al., 1999). It connects the new graph vertex e to each vertex in the hyperedge to it, i.e. $E^* = \{(u, e) : u \in e, e \in E\}$.

Note that each hyperedge in E corresponds to a star in the graph G^* and that G^* is a bi-partite graph. Star expansion assigns the scaled hyperedge weight to each corresponding graph edge:

$$w^*(u,e) = w(e)/\delta(e) \quad (13)$$

The combinatorial or normalized Laplacian of the constructed graph G^x is then used to partition the vertices.

5.3. Bolla's Laplacian

Bolla (Bolla, 1993) defines a Laplacian for an unweighted hypergraph in terms of the diagonal vertex degree matrix D_v, the diagonal edge degree matrix D_e, and the incidence matrix H, defined in Section 2.

$$L^o := D_v - HD_e^{-1}H^\top. \quad (14)$$

The eigenvectors of Bolla's Laplacian L^o define the "best" Euclidean embedding of the hypergraph. Here, the cost for embedding $\phi : V \to \mathbb{R}^k$ of the hypergraph is the total squared distance between pairs of embedded vertices in the same hyperedge

$$\sum_{u,v \in V} \sum_{e \in E: u,v \in e} \|\phi(u) - \phi(v)\|^2 \quad (15)$$

Bolla shows a relationship between the spectral properties of L^o and the minimum cut of the hypergraph.

5.4. Rodriguez's Laplacian

Rodríguez (Rodríguez, 2003; Rodríguez, 2002) constructs a weighted graph $G^r(V, E^r = E^x)$ from an unweighted hypergraph $G(V, E)$. Like clique expansion, each hyperedge is replaced by a clique in the graph G^r. The weight $w^r(u, v)$ of an edge is set to the number of edges containing both u and v:

$$w^r(u,v) = |\{e \in E : u, v \in e\}| \quad (16)$$

Rodríguez expresses the graph Laplacian applied to G^r in terms of the hypergraph structure:

$$L^r(G^r) = D_v^r - HH^\top \quad (17)$$

where D_v^r is the vertex degree matrix of the graph G^r. Like Bolla, Rodriguez shows a relationship between the spectral properties of L^r and the cost of minimum partitions of the hypergraph.

5.5. Zhou's Normalized Laplacian

Zhou et al. (Zhou et al., 2005) generalize their earlier work on regularization on graphs and consider the following regularization on a vertex function f.

$$\langle f, L^z f \rangle = \frac{1}{2} \sum_{e \in E} \frac{1}{\delta(e)} \sum_{\{u,v\} \subseteq e} w(e) \left(\frac{f(u)}{\sqrt{d(u)}} - \frac{f(v)}{\sqrt{d(v)}} \right)^2$$

Note that this regularization term is small if vertices with high affinities have the same label. They show that the operator L^z can then be written as

$$L^z = I - D_v^{-1/2} H W D_e^{-1} H^\top D_v^{-1/2} \qquad (18)$$

In addition, Zhou et al. define a hypergraph normalized cut criterion for a k-partition of the vertices $P_k = \{V_1, ..., V_k\}$:

$$\text{NCut}(P_k) := \sum_{i=1}^{k} \frac{\sum_{e \in E} w(e) |e \cap V_i| |e \cap V_i^c|}{\delta(e) \sum_{v \in V_i} d(v)}. \qquad (19)$$

This criterion is analogous to the normalized cut criterion for graphs. They then show that if minimizing the normalized cut is relaxed to a real-vealed optimization problem, the second smallest eigenvector of L^z is the optimal classification function f. Finally, they also draw a parallel between their hypergraph normalized cut criterion and random walks over the hypergraph.

5.6. Gibson's Dynamical System

In (Gibson et al., 1998) the authors have proposed a dynamical system to cluster categorical data that can be represented using a hypergraph. They consider the following iterative process.

1. $s_{ij}^{n+1} = \sum_{e:i \in e} \sum_{k \neq i \in e} w_e s_{kj}^n$

2. Orthonormalize the vectors s_j^n.

They prove that the above iteration is convergent. We observe that

$$s_{ij}^{n+1} = \sum_{e} h(i, e) \left(\sum_{k} h(k, e) w_e s_{kj}^n - w_e s_{ij}^n \right)$$
$$s_j^{n+1} = (HWH^\top - D_v) s_j^n \qquad (20)$$

Thus, the iterative procedure described above is the power method for calculating the eigenvectors of the adjacency matrix $S = D_v - HWH^\top$.

5.7. Li's Adjacency Matrix

Li et al. (Li & Solé, 1996) formally define properties of a regular, unweighted hypergraph $G(V, E)$ in terms of the star expansion of the hypergraph. In particular, they define the $|V| \times |V|$ adjacency matrix of the hypergraph, $HH^\top$. They show a relationship between the spectral properties of the adjacency matrix of the hypergraph $HH^\top$ and the structure of the hypergraph.

6. Comparing Hypergraph Learning Algorithms

In this section, we compare the algorithms for learning from a hypergraph representation of data described in Section 5. In Section 6.1, we compute the normalized Laplacian for the star expansion graph. In Section 6.2, we compute the combinatorial and normalized Laplacian of the clique expansion graph. In Section 6.3, we show that these Laplacians are nearly equivalent to each other. Finally, in Section 6.4, we show that the various hypergraph Laplacians can be written as the graph Laplacian of the clique expansion graph.

We begin by stating a simple lemma. The proof is trivial.

Lemma 1. *Let,*

$$B = \begin{bmatrix} I & -A \\ -A^\top & I \end{bmatrix}$$

be a block matrix with A rectangular. Consider the eigenvalue problem

$$\begin{bmatrix} I & -A \\ -A^\top & I \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} = \lambda \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix}$$

then the following relation holds

$$AA^\top \mathbf{x} = (1 - \lambda)^2 \mathbf{x}$$

6.1. Star Graph Laplacian

Given a hypergraph $G(V, E)$, consider the star graph $G^*(V^*, E^*)$, i.e. $V^* = V \cup E$, $E^* = \{(u, e) : u \in e, e \in E\}$. Notice that this is a bipartite graph, with vertices corresponding to E on one side and vertices corresponding to V on the other, since there are no edges from V to V or from E to E. Let us also assume that the vertex set V^* has been ordered such that all elements of V come before elements of E.

Let $w^* : V \times E \to \mathbb{R}^+$ be the (as yet unspecified) graph edge weight function. In addition, let S^* be the $(|V| + |E|) \times (|V| + |E|)$ affinity matrix. We can write the affinity matrix in terms of the hypergraph structure

and the weight function w^* as

$$S^* = \begin{bmatrix} 0_{|V|} & HW^* \\ W^*H^\top & 0_{|E|} \end{bmatrix} \tag{21}$$

The degrees of vertices in G^* are then

$$d^*(u) = \sum_{e \in E} h(u,e)w^*(u,e) \quad u \in V \tag{22}$$

$$d^*(e) = \sum_{u \in V} h(u,e)w^*(u,e) \quad e \in E \tag{23}$$

The normalized Laplacian of this graph can now be written in the form

$$\mathcal{L}^* = \begin{bmatrix} I & -A \\ -A^\top & I \end{bmatrix}. \tag{24}$$

Here, A is the $|V| \times |E|$ matrix

$$A = D_v^{*-1/2} H W D_v^{*-1/2}$$

with entry (u,e)

$$A_{ue} = \frac{h(u,e)w^*(u,e)}{\sqrt{d^*(u)}\sqrt{d^*(e)}}. \tag{25}$$

Any $|V|+|E|$ eigenvector $\mathbf{x}^\top = [\mathbf{x}_v^\top, \ \mathbf{x}_e^\top]$ of $\mathcal{L}^*$ satisfies $\mathcal{L}^*\mathbf{x} = \lambda\mathbf{x}$. Then by Lemma 1, we know that

$$AA^\top \mathbf{x}_v = (\lambda - 1)^2 \mathbf{x}_v. \tag{26}$$

Thus, the $|V|$ elements of the eigenvectors of the normalized Laplacian $\mathcal{L}^*$ corresponding to vertices $V \subseteq V^*$ are the eigenvectors of the $|V| \times |V|$ matrix $AA^\top$. Element (u,v) of $AA^\top$ is

$$[AA^\top]_{uv} = \sum_{e \in E} \frac{h(u,e)h(v,e)w^*(u,e)w^*(v,e)}{\sqrt{d^*(u)}d^*(e)\sqrt{d^*(v)}}. \tag{27}$$

For the standard star expansion weighting function, $w^*(u,e) = w(e)/\delta(e)$, so the vertex degrees are

$$d^*(u) = \sum_{e \in E} h(u,e)w(e)/\delta(e) \quad u \in V \tag{28}$$

$$d^*(e) = \sum_{u \in e} w(e)/\delta(e) = w(e) \quad e \in E \tag{29}$$

Thus, we can write

$$[AA^\top]_{uv}^* = \ = \sum_{e \in E} \frac{h(u,e)h(v,e)w(e)/\delta(e)^2}{\sqrt{d^*(u)}\sqrt{d^*(v)}} \tag{30}$$

6.2. Clique Graph Laplacian

Given a hypergraph $G(V,E)$, consider the graph $G^c(V, E^c = E^x)$ with the same structure as the clique expansion graph, i.e. $E^c = \{(u,v) : u,v \in e, e \in E.$

Let $w^c : V \times E \to \mathbb{R}^+$ be the (as yet unspecified) hypergraph edge weight. We can write the normalized Laplacian of G^c in terms of the hypergraph structure and the weight function w^c as $\mathcal{L}^c := I - C$. If there is no hyperedge $e \in E$ such that $u,v \in E$ then $C_{uv} = 0$. Otherwise,

$$[C]_{uv} = \frac{w^c(u,v)}{\sqrt{d^c(u)}\sqrt{d^c(v)}} \tag{31}$$

where

$$d^c(u) = \sum_{e \in E} h(u,e) \sum_{v \in e \setminus \{u\}} w^c(u,v) \tag{32}$$

is the vertex degree. For the standard clique expansion construction,

$$w^c(u,v) = w^x(u,v) = \sum_{e \in E: u,v \in e} w(e). \tag{33}$$

so the vertex degrees are

$$d^c(u) = d^x(u) = \sum_{e \in E} h(u,e)(\delta(e) - 1)w(e) \tag{34}$$

6.3. Unifying Star and Clique Expansion

To show the relationship between star and clique expansion, consider the star expansion graph $G_c^*(V^*, E^*)$ with weighting function

$$w_c^*(u,e) := w(e)(\delta(e) - 1) \tag{35}$$

Note that this is $(\delta(e) - 1)\delta(e)$ times the standard star expansion weighting function $w^*(u,e)$ (Eq. (13)). Plugging this value into Equations (22) and (23), we get that the degrees of vertices in G^* are

$$d_c^*(u) = \sum_{e \in E} h(u,e)w(e)(\delta(e) - 1) = d^x(u) \tag{36}$$

$$d_c^*(e) = w(e)\delta(e)(\delta(e) - 1) \tag{37}$$

where $d^x(u)$ is the vertex degree for the standard clique expansion graph G_x^*. Thus,

$$[A_c^* A_c^{*\top}]_{uv} = \sum_{e \in E} \left[\frac{\delta(e) - 1}{\delta(e)} \right] \frac{h(u,e)h(v,e)w(e)}{\sqrt{d^x(u)}\sqrt{d^x(v)}} \tag{38}$$

Similarly, suppose we choose the clique expansion weighting function

$$w_*^c(u,v) := \frac{\sum_{e \in E} h(u,e)h(v,e)w(e)}{\delta(e)(1 - \delta(e))} \tag{39}$$

Then we can show that the vertex degree is

$$d_*^c(u) = \sum_{e \in E} h(u,e)w(e)/\delta(e) = d^*(u) \qquad (40)$$

where $d^*(u)$ is the vertex degree function for standard star expansion. We can then write

$$[C_*]_{uv} = \sum_{e \in E} \frac{1}{\delta(e)\delta(e-1)} \frac{h(u,e)h(v,e)w(e)}{\sqrt{d^*(u)}\sqrt{d^*(v)}} \qquad (41)$$

A commonly occuring case is the k-uniform hypergraph. In this case, each hyperedge has exactly the same number of vertices, i.e. $\delta(e) = k$. Then it is easy to see that the bipartite graph matrix $A_c^* A_c^{*\top}$ is a constant scalar times the clique expansion matrix C. Thus, the eigenvectors of the normalized Laplacian for the bipartite graph G_c^* are exactly the eigenvectors of the normalized Laplacian for the standard clique expansion graph G^x. Similarly, the clique matrix C_* is a constant scalars times the standard star expansion matrix $[AA^\top]^*$. Thus, the eigenvectors of the normalized Laplacian for the clique graph G_*^c are exactly the eigenvectors of the normalized Laplacian for standard star expansion. This is a surprising result, since the two graphs are completely different in the number of vertices and the connectivity between these vertices.

For non-uniform hypergraphs (i.e. the hyperedge cardinality varies), the bipartite graph matrix $A_c^* A_c^{*\top}$ while not the same is close to the clique expansion matrix C_*^c. Each term in the sum in Equation (38) has an additional factor $(\delta(e)-1)/\delta(e)$, giving slightly higher weight to hyperedges with a higher degree. This difference however is not large, especially with higher cardinalities. As the bipartite graph matrix $A_c A_c^\top$ is approximately the clique expansion matrix, we conclude that their eigenvectors are similar. A similar relation holds for the clique graph G_*^c and the standard star expansion where the clique graph gives lower weight to larger edges. These observations can be reversed to characterize the behavior of the standard clique expansion and star expansion construction, and we conclude that the clique expansion gives more weight to evidence from larger edges than star expansion.

There is no clear reason why one should give more weight to smaller hyperedges versus larger edges or vice versa. The exact choice will depend on the properties of the affinity function used.

6.4. Unifying Hypergraph Laplacians

In this section we take a second look at the various constructions in Section 5 and show that they all correspond to either clique or star expansion of the original hypergraph with the appropriate weighting function.

For an unweighted hypergraph, Bolla's Laplacian L^o corresponds to the unnormalized Laplacian of the associated clique expansion with the weight matrix of the hypergraph the inverse of the degree matrix D_e:

$$W^o = HD_e^{-1}H^\top \qquad (42)$$

The row sums of this matrix are given by

$$d^o(u) = \sum_v \sum_{e \in E} h(u,e)\frac{1}{\delta(e)}h(v,e) = \sum_{e \in E} h(u,e) \quad (43)$$

which as a diagonal matrix is exactly the vertex degree matrix D_v for an unweighted hypergraph, giving us the unnormalized Laplacian

$$L^o = D_v - HD_e^{-1}H^\top \qquad (44)$$

The Rodríguez Laplacian can similarly be shown to be the unnormalized Laplacian of the clique expansion of an unweighted graph with every hyperedge weight set to 1. Similarly, Gibson's algorithm calculates the eigenvectors of the adjacency matrix for the clique expansion graph.

We now turn our attention to the normalized Laplacian of Zhou et al. Consider the star expansion of the hypergraph with the weight function $w^z(u,e) = w(e)$. Then the adjacency matrix for the resulting bi-partite graph can be written as

$$S^z = \begin{bmatrix} 0 & HW \\ WH^\top & 0 \end{bmatrix} \qquad (45)$$

It is easy to show that the degree matrix for this graph is the diagonal matrix

$$D^z = \begin{bmatrix} D_v & 0 \\ 0 & WD_e \end{bmatrix} \qquad (46)$$

Thus the normalized Laplacian for this bi-partite graph is given by the matrix

$$\begin{bmatrix} I & -D_v^{-1/2}HW^{-1/2}D_e^{-1/2} \\ -D_e^{-1/2}W^{-1/2}H^\top WD_v^{-1/2} & I \end{bmatrix}$$

Now if we consider the eigenvalue problem for this matrix, with eigenvectors $\mathbf{x}^\top = [\ \mathbf{x}_v \quad \mathbf{x}_e\ ]$ then by Lemma 1, we can show that $\mathbf{x}_v$ is given by the following eigenvalue problem.

$$D_v^{-1/2}HWD_e^{-1}H^\top D_v^{-1/2}\mathbf{x}_v = (1-\lambda)^2\mathbf{x}_v \quad (47)$$

$$(I - D_v^{-1/2}HWD_e^{-1}H^\top D_v^{-1/2})\mathbf{x}_v = (1-(1-\lambda)^2)\mathbf{x}_v$$

This is exactly the same eigenvalue problem that Zhou et al. propose for the solution of the clustering problem. Thus Zhou et al.'s Laplacian is equivalent to constructing a star expansion and using the normalized Laplacian defined on it. The following table summarizes this discussion.

Algorithm	Graph	Matrix
Bolla	Clique	Combinatorial Laplacian
Rodríguez	Clique	Combinatorial Laplacian
Zhou	Star	Normalized Laplacian
Gibson	Clique	Adjacency
Li	Star	Adjacency

Table 1. This table summarizes the various hypergraph learning algorithms, their underlying graph construction and the associated matrix used for the spectral analysis.

7. Discussion

In this paper we have examined the use of hypergraphs in learning with higher order relations. We surveyed the various Laplace like operators that have been constructed to analyze the structure of hypergraphs. We showed that all of these methods, despite their very different formulations, can be reduced to two graph constructions – the star expansion and the clique expansion – and the study of their associated Laplacians. We have also shown that for the commonly occurring case of k-uniform graphs these two constructions are identical. This is a surprising and unexpected result as the two graph constructions are completely different in structure. In the case of non-uniform graphs, we showed that the essential difference between the two constructions is how they weigh the evidence from hyperedges of differing sizes.

In summary, while hypergraphs may be an intuitive representation of higher order similarities, it seems (anecdotally at least) that graphs lie at the heart of this problem.

Acknowledgements

References

Agarwal, S., Lim, J., Zelnik-Manor, L., Perona, P., Kriegman, D. J., & Belongie, S. (2005). Beyond pairwise clustering. *CVPR (2)* (pp. 838–845).

Alpert, C. J., Kahng, A. B., & Yao, S.-Z. (1999). Spectral partitioning with multiple eigenvectors. *Discrete Applied Mathematics*, *90*, 3–26.

Belkin, M., & Niyogi, P. (2003). Semi-supervised learning on riemannian manifolds. *Mach. Learn.*, *56*, 209–239.

Bolla, M. (1993). Spectra, euclidean representations and clusterings of hypergraphs. *Discrete Mathematics*, *117*.

Chung, F. R. (1993). The Laplacian of a hypergraph. In J. Friedman (Ed.), *Expanding graphs (DIMACS series)*, 21–36. AMS.

Chung, F. R. K. (1997). *Spectral graph theory*. AMS.

Forman, R. (2003). Bochner's method for cell complexes and combinatorial Ricci curvature. *Discrete & Computational Geometry*, *29*, 323–374.

Gibson, D., Kleinberg, J. M., & Raghavan, P. (1998). Clustering categorical data: An approach based on dynamical systems. *VLDB* (pp. 311–322).

Govindu, V. M. (2005). A tensor decomposition for geometric grouping and segmentation. *CVPR*.

Li, W., & Solé, P. (1996). Spectra of regular graphs and hypergraphs and orthogonal polynomials. *European Journal of Combinatorics*, *17*, 461–477.

Munkres, J. R. (1984). *Elements of algebraic topology*. The Benjamin/Cummings Publishing Company.

Ng, A., Jordan, M., & Weiss, Y. (2002). On spectral clustering: Analysis and an algorithm. *NIPS*.

Rodríguez, J. A. (2002). On the Laplacian eigenvalues and metric parameters of hypergraphs. *Linear and Multilinear Algebra*, *50*, 1–14.

Rodríguez, J. A. (2003). On the Laplacian spectrum and walk-regular hypergraphs. *Linear and Multilinear Algebra*, *51*, 285–297.

Rosenberg, S. (1997). *The Laplacian on a Riemannian manifold*. London Mathematical Society.

Shashua, A., & Hazan, T. (2005). Non-negative tensor factorization with applications to statistics and vision. *ICML*.

Shashua, A., Zass, R., & Hazan, T. (2006). Multi-way clustering using super-symmetric non-negative tensor factorization. *ECCV*.

Shi, J., & Malik, J. (2000). Normalized cuts and image segmentation. *PAMI*, *22*, 888–905.

Smola, A., & Kondor, I. (2003). Kernels and regularization on graphs. *COLT*. Springer.

Zhou, D., Huang, J., & Schölkopf, B. (2005). *Beyond pairwise classification and clustering using hypergraphs* (Technical Report 143). Max Plank Institute for Biological Cybernetics, Tübingen, Germany.

Zhou, D., & Schölkopf, B. (2005). Regularization on discrete spaces. *Pattern Recognition*, 361–368.

Zhu, X. (2005). *Semi-supervised learning literature survey* (Technical Report 1530). Computer Sciences, University of Wisconsin.

Zien, J. Y., Schlag, M. D. F., & Chan, P. K. (1999). Multilevel spectral hypergraph partitioning with arbitrary vertex sizes. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, *18*, 1389–1399.