

# Hypergraph $p$ -Laplacian: A Differential Geometry View

**Shota Saito**

The University of Tokyo  
ssaito@sat.t.u-tokyo.ac.jp

**Danilo P Mandic**

Imperial College London  
d.mandic@imperial.ac.uk

**Hideyuki Suzuki**

Osaka University  
hideyuki@ist.osaka-u.ac.jp

## Abstract

The graph Laplacian plays key roles in information processing of relational data, and has analogies with the Laplacian in differential geometry. In this paper, we generalize the analogy between graph Laplacian and differential geometry to the hypergraph setting, and propose a novel hypergraph  $p$ -Laplacian. Unlike the existing two-node graph Laplacians, this generalization makes it possible to analyze hypergraphs, where the edges are allowed to connect any number of nodes. Moreover, we propose a semi-supervised learning method based on the proposed hypergraph  $p$ -Laplacian, and formalize them as the analogue to the Dirichlet problem, which often appears in physics. We further explore theoretical connections to normalized hypergraph cut on a hypergraph, and propose normalized cut corresponding to hypergraph  $p$ -Laplacian. The proposed  $p$ -Laplacian is shown to outperform standard hypergraph Laplacians in the experiment on a hypergraph semi-supervised learning and normalized cut setting.

## Introduction

Graphs are a standard way to represent pairwise relationship data on both regular and irregular domains. One of the most important operators characterizing a graph is the graph Laplacian, which can be explained in several ways. For the example of spectral clustering (von Luxburg 2007), we consider normalized graph cut (Shi and Malik 1997; Yu and Shi 2003), random walks (Meila and Shi 2001; Grady 2006), and analogues to differential geometry of graphs (Branin 1966; Grady and Schwartz 2003; Zhou and Schölkopf 2006; Bougleux, Elmoataz, and Melkemi 2007).

Hypergraphs are a natural generalization of graphs, where the edges are allowed to connect more than two nodes (Berge 1984). The data representation with a hypergraph is used in a variety of applications (Huang, Liu, and Metaxas 2009; Liu, Latecki, and Yan 2010; Klamt, Haus, and Theis 2009; Tan et al. 2014). This natural generalization of graphs motivates us to consider a natural generalization of Laplacian to hypergraphs, which can be applied to hypergraph clustering problems. However, there is no straightforward approach to generalize the graph Laplacian to a hypergraph Laplacian. One way is to model a hypergraph as a tensor, for which we can define Laplacian (Cooper and Dutle 2012; Hu and Qi

2015) and construct hypergraph cut algorithms (Bulò and Pelillo 2009; Ghoshdastidar and Dukkipati 2014). However, this requires the hypergraph to obey a strict condition of a  $k$ -uniform hypergraph, where each edge connects exactly  $k$  nodes. The second approach is to construct a weighted graph, which can deal with arbitrary hypergraphs. Rodriguez's approach defines Laplacian of arbitrary hypergraph as an adjacency matrix of weighted graph (Rodriguez 2002). Zhou's approach defines a hypergraph from the normalized cut approach, and outperforms Rodriguez's Laplacian on a clustering problem (Zhou, Huang, and Schölkopf 2006). However, Rodriguez's Laplacian does not consider how many nodes are connected by each edge, and Zhou's Laplacian is not consistent with the graph Laplacian. Although all of previous studies consider the analogue to graph Laplacian, none of them considers the analogue to the Laplacian from differential geometry. This allows us to further extend to more general hypergraph  $p$ -Laplacian, which is not extensively studied unlike in the case of graph  $p$ -Laplacian (Bühler and Hein 2009; Zhou and Schölkopf 2006).

In this paper, we generalize the analogy between graph Laplacian and differential geometry to the hypergraph setting, and propose a novel hypergraph  $p$ -Laplacian, which is consistent with the graph Laplacian. We define gradient of the function over hypergraph, and induce the divergence and Laplacian as formulated in differential geometry. Taking advantage of this formulation, we extend our hypergraph Laplacian to a hypergraph  $p$ -Laplacian, which allows us to better capture hypergraph characteristics. We also propose a semi-supervised machine learning method based upon this  $p$ -Laplacian. Our experiment on hypergraph semi-supervised clustering problem shows that our hypergraph  $p$ -Laplacian outperforms the current hypergraph Laplacians.

The versatility of differential geometry allows us to introduce several rigorous interpretations of our hypergraph Laplacian. A normalized cut formulation is shown to yield the proposed hypergraph Laplacian in the same manner as in standard graphs. We further propose a normalized cut corresponding to our  $p$ -Laplacian, which shows better performance than the ones corresponding to current Laplacians in the experiments. We also explore the physical interpretation of hypergraph Laplacian, by considering the analogue to the continuous  $p$ -Dirichlet problem, which is widely used in Physics. *All proofs are in the supplemental material.*

## Differential Geometry on Hypergraphs

### Preliminary Definition of Hypergraph

In this section, we review standard definitions and notations from hypergraph theory. We refer to the literature (Berge 1984) for a more comprehensive study. A *hypergraph*  $G$  is a pair  $(V, E)$ , where  $E \subset \bigcup_{k=1}^{|V|} \cup_{\{v_1, \dots, v_k\} \subset V} \{[v_{\sigma(1)}, \dots, v_{\sigma(k)}] \mid \sigma \in \mathcal{S}_k\}$ , and  $\mathcal{S}_k$  denotes the set of permutations  $\sigma$  on  $\{1, \dots, k\}$ . An element of  $V$  is called a *vertex* or *node*, and an element of  $E$  is referred to as an *edge* or *hyperedge* of the hypergraph. A hypergraph is *connected* if the intersection graph of the edges is connected. In what follows, we assume that the hypergraph  $G$  is connected. A hypergraph is *undirected* when the set of edges are symmetric, and we denote a set of undirected edges as  $E_{un} = E/\mathcal{S}$ , where  $\mathcal{S} = \bigcup_{k=1}^{|V|} \mathcal{S}_k$ . In other words, edges  $[v_1, v_2, \dots, v_k] \in E$  and  $[v_{\sigma(1)}, v_{\sigma(2)}, \dots, v_{\sigma(k)}] \in E$  are not distinguished in  $E_{un}$  for any  $\sigma \in \mathcal{S}_k$ , where  $k$  is the number of nodes in the edge. A hypergraph is *weighted* when it is associated with a function  $w: E \rightarrow \mathbb{R}^+$ . For an undirected hypergraph it holds that  $w([v_1, v_2, \dots, v_k]) = w([v_{\sigma(1)}, v_{\sigma(2)}, \dots, v_{\sigma(k)}])$ . We define the *degree* of a node  $v \in V$  as  $d(v) = \sum_{e \in E: v \in e} w(e)$ , while the degree of an edge  $e \in E$  is defined as  $\delta(e) = |e|$ . To simplify the notation we write  $\delta_e$  instead of  $\delta(e)$ .

We define  $\mathcal{H}(V)$  as a Hilbert space of real-valued functions endowed with the usual inner product

$$\langle f, g \rangle_{\mathcal{H}(V)} := \sum_{v \in V} f(v)g(v) \quad (1)$$

for all  $f, g \in \mathcal{H}(V)$ . Accordingly, the Hilbert space  $\mathcal{H}(E)$  is defined with the inner product

$$\langle f, g \rangle_{\mathcal{H}(E)} := \sum_{e \in E} \frac{1}{\delta_e!} f(e)g(e). \quad (2)$$

Note that  $f$  and  $g$  are defined for directed edges.

### Hypergraph Gradient and Divergence Operators

We shall now extend standard graph gradient and divergence operators studied in (Zhou and Schölkopf 2006) to hypergraphs, which can be considered as hypergraph analogues in both of discrete and continuous case. First, we propose to define hypergraph gradient as follows.

**Definition 1.** *The hypergraph gradient is an operator  $\nabla: \mathcal{H}(V) \rightarrow \mathcal{H}(E)$  defined by*

$$(\nabla\psi)([v_1, \dots, v_{\delta_e}] = e) := \frac{\sqrt{w(e)}}{\sqrt{\delta_e - 1}} \sum_{i=1}^{\delta_e} \left( \frac{\psi(v_i)}{\sqrt{d(v_i)}} - \frac{\psi(v_1)}{\sqrt{d(v_1)}} \right) \quad (3)$$

The gradient is defined as a sum of a pairwise smoothness term between  $e_{[1]}$  node and the others. Since the coefficient of graph gradient is defined as a square root of the weight  $w(e)$  (Zhou and Schölkopf 2006), we derive the coefficient for hypergraph by considering the average among the pairs between  $e_{[1]}$  and the other node,  $w(e)$  divided by  $\delta_e - 1$ , in order to normalize the effect of weight. For an undirected hypergraph, we define a gradient for an edge  $e \in E_{un}$  and vertex  $v$ , i.e.  $(\nabla\psi)(e, v)$ . Using the gradient defined for each

edge, we can define the gradient at each node  $v$  as  $\nabla\psi(v) := \{(\nabla\psi)(e)/\delta_e! \mid e_{[1]} = v, e \in E\}$ , where  $e_{[1]}$  denotes the first element of edge  $e$ . Then, the norm of the gradient  $\nabla\psi$  at node  $v$  is defined by

$$\|\nabla\psi(v)\| := \left( \sum_{e \in E: e_{[1]} = v} \frac{(\nabla\psi)^2(e)}{\delta_e!} \right)^{\frac{1}{2}}. \quad (4)$$

It then follows that the definition of this norm satisfies the conditions of a norm in a metric space. The *p-Dirichlet sum* of the function  $\psi$  is given by

$$S_p(\psi) := \sum_{v \in V} \|\nabla\psi(v)\|^p. \quad (5)$$

Loosely speaking, the norm of the gradient on a node of a hypergraph measures local smoothness of the function around the node, and the Dirichlet sum measures total roughness over the hypergraph. Remark that  $\|\nabla\psi\|$  is defined in the space  $\mathcal{H}(E)$  as  $\|\nabla\psi\| = \langle \nabla\psi, \nabla\psi \rangle_{\mathcal{H}(E)}^{1/2}$ , and satisfies  $S_2 = \|\nabla\psi\|^2$ .

**Definition 2.** *The hypergraph divergence is an operator  $\text{div}: \mathcal{H}(E) \rightarrow \mathcal{H}(V)$  which satisfies  $\forall \psi \in \mathcal{H}(V), \forall \phi \in \mathcal{H}(E)$*

$$\langle \nabla\psi, \phi \rangle_{\mathcal{H}(E)} = \langle \psi, -\text{div}\phi \rangle_{\mathcal{H}(V)}. \quad (6)$$

Notice that Eq. (6) can be regarded as a hypergraph analogue of Stokes' Theorem on manifolds. The divergence can now be written in a closed form as follows:

**Proposition 3.**

$$\begin{aligned} \text{div}\phi(v) = & - \sum_{e \in E: v \in e} \frac{\sqrt{w(e)}}{\delta_e! \sqrt{\delta_e - 1} \sqrt{d(v)}} \phi(e) \\ & + \sum_{e \in E: e_{[1]} = v} \frac{\delta_e \sqrt{w(e)}}{\delta_e! \sqrt{\delta_e - 1} \sqrt{d(v)}} \phi(e). \end{aligned} \quad (7)$$

Intuitively, Eq. (7) measures the net flows at the vertex  $v$ ; the first term counts the outflows from the originator  $v$  and the second term measures the inflow towards  $v$ . Note that this allows us to use Eq. (7) as a definition of the divergence; it satisfies Eq. (6), analogously to Stokes' theorem in the continuous case. Note also that divergence is always 0 if  $\phi$  is undirected i.e  $\phi(v_1, v_2, \dots, v_k) = \phi(v_{\sigma(1)}, v_{\sigma(2)}, \dots, v_{\sigma(k)})$ .

### Laplace Operators

In this section, we present the *hypergraph p-Laplace operator*, which can be considered as a discrete analogue of the Laplacian in the continuous case.

**Definition 4.** *The hypergraph Laplacian is an operator  $\Delta_p: \mathcal{H}(V) \rightarrow \mathcal{H}(V)$  defined by*

$$\Delta_p\psi := -\text{div}(\|\nabla\psi\|^{p-2}\nabla\psi). \quad (8)$$

This operator is linear for  $p=2$ . For an undirected hypergraph, we get the hypergraph  $p$ -Laplacian as follows;

**Proposition 5.**

$$(\Delta_p\psi)(v) = \sum_{u \in V \setminus \{v\}} \left( d_p(v) \frac{\psi(v)}{\sqrt{d(v)}} - w_p(u, v) \frac{\psi(u)}{\sqrt{d(u)}} \right) \quad (9)$$

denoting  $w_p(v, v) = 0$  and

$$w_p(u, v) = \sum_{e \in E_{un}; u, v \in e} \frac{w(e)}{\delta_e - 1} \times (-\|\nabla\psi_e\|^{p-2} + \|\nabla\psi(u)\|^{p-2} + \|\nabla\psi(v)\|^{p-2}),$$

and

$$d_p(v) = d(v)\|\nabla\psi(v)\|^{p-2} - \sum_{e \in E_{un}; v \in e} \left( \frac{w(e)}{\delta_e - 1} (-\|\nabla\psi_e\|^{p-2} + \|\nabla\psi(v)\|^{p-2}) \right),$$

where  $\|\nabla\psi_e\|^p = \sum_{v' \in e} \|\nabla\psi(v')\|^p / \delta_e$ .

Let  $W_p$  be a matrix whose elements  $w_p(u, v)$ ,  $D_p$  be a diagonal matrix whose elements  $d(u, u) = \sum_{v \in V} d_p(u, v)$ . For  $p = 2$  case, which is a standard setting for hypergraph Laplacian, it becomes

$$(\Delta\psi)(v) = \sum_{u \in V \setminus \{v\}} \left( d(v) \frac{\psi(v)}{\sqrt{d(v)}} - w(u, v) \frac{\psi(u)}{\sqrt{d(u)}} \right), \quad (10)$$

where

$$w(u, v) = \sum_{e \in E_{un}; u, v \in e} \frac{w(e)}{\delta_e - 1}, \quad w(u, u) = 0.$$

We denote  $W_2$  by  $W$  and a diagonal matrix  $D$  whose elements by  $d(u, u) = d(u)$ . Note that  $d_p(u, u) = \sum_{u \in V} w_p(v, u)$ . Using these matrices the Laplacian in (8) can be rewritten as

$$(\Delta_p\psi) = D^{-1/2}(D_p - W_p)D^{-1/2}\psi. \quad (11)$$

We shall denote the matrix associated with the Laplacian by  $L_p = D^{-1/2}(D_p - W_p)D^{-1/2}$ , so that the Dirichlet sum can be rewritten by using  $L_p$  as follows.

**Proposition 6.** *The Dirichlet sum  $S_p(\psi)$  can be rewritten as*

$$S_p(\psi) = \psi^\top L_p \psi. \quad (12)$$

Note that  $L_p$  depends on the function  $\psi$ , while  $L := L_2$  is independent. When the hypergraph degenerates into a standard graph and  $p = 2$ ,  $L$  coincides with the graph Laplacian.

From the above analysis, the following three statements follow straightforwardly.

**Proposition 7.**  $\langle \psi, \Delta_p \psi \rangle_{\mathcal{H}(V)} = S_p(\psi)$

**Corollary 8.** *The Laplacian  $L_p$  is positive semi-definite.*

**Proposition 9.**  $\frac{\partial}{\partial \psi} S_p(\psi) = p \Delta_p \psi$

**Remark 1.** For the case of standard graph in this setting, the discussion in this section reduces to the discrete geometry for standard graphs, as introduced in (Zhou and Schölkopf 2006). This implies that our proposed definition is a natural generalization of discrete geometry for a graph.

## Hypergraph Regularization

### Hypergraph Regularization Algorithm

In this section, we consider the hypergraph regularization problem and propose a novel solution. Given the hypergraph  $H = (V, E)$  and label set  $\mathcal{Y} = \{-1, 1\}$ , and assume that the subset of  $S \subset V$  is labeled, the problem is to classify the vertices in  $V \setminus S$  using the label of  $S$ . To solve this, we formulate hypergraph regularization as follows. The regularization of a given function  $y \in \mathcal{H}(V)$  aims to find a function  $\psi^*$ , which enforces smoothness on all the nodes of the hypergraph, and at the same time closeness to the values of a given function  $y$ , as follows:

$$\psi^* = \operatorname{argmin}_{\psi \in \mathcal{H}(V)} (S_p(\psi) + \mu \|\psi - y\|^2), \quad (13)$$

where  $y(v)$  takes  $-1$  or  $1$  if  $v$  is labeled,  $0$  otherwise. The first energy term represents the smoothness as explained in Eq. (5), while the second term is a regularization term. Let  $\mathcal{E}_p(\psi, y, \mu)$  be the objective function of Eq. (13). Since the positive power of positive convex function is also convex,  $\mathcal{E}_p$  is a convex function for  $\psi$ , meaning that Eq. (13) has a unique solution, satisfying

$$\frac{\partial \mathcal{E}_p}{\partial \psi} \Big|_v = \frac{\partial}{\partial \psi} \|\nabla_v \psi\|^p + 2\mu(\psi(v) - y(v)) = 0, \quad (14)$$

for all  $v \in V$ . Using Proposition 9 we can rewrite the left hand side of Eq. (14) as

$$p(\Delta_p \psi)(v) + 2\mu(\psi(v) - y(v)) = 0, \quad \forall v \in V. \quad (15)$$

The solution to the problem (13) is therefore also solution to (15). Substituting the expression of the Laplacian from Eq. (9) into Eq. (15) yields

$$p \sum_{u \in V} l_p(u, v) \psi(u) + 2\mu(\psi(v) - y(v)) = 0, \quad (16)$$

where  $l_p(u, v)$  is a element of  $L_p$ . To solve Eq. (16) numerically, we shall use the Gauss-Jacobi iterative algorithm, similarly to the discrete case introduced in (Bougleux, Elmoataz, and Melkemi 2007). Let  $\psi^{(t)}$  be the solution obtained at the iteration step  $t$ , the update rule of the corresponding linearized Gauss-Jacobi algorithm is then given by

$$\psi^{(t+1)}(v) = \sum_{u \in V \setminus \{v\}} c^{(t)}(u, v) \psi^{(t)}(u) + m^{(t)}(v) y(v), \quad (17)$$

where

$$c^{(t)}(u, v) = -\frac{pl_p^{(t)}(u, v)}{pl_p^{(t)}(v, v) + 2\mu} \text{ and } m^{(t)}(v) = \frac{2\mu}{pl_p^{(t)}(v, v) + 2\mu},$$

and where  $l_p^{(t)}(u, v)$  is a  $p$ -Laplacian defined by  $\psi^{(t)}$ . We take  $\psi^{(0)} = y$  as an initial condition. The following theorem guarantees the convergence of the update rule for arbitrary  $p$ .

**Theorem 10.** *The update rule Eq. (17) yields a convergent sequence. Moreover, with notations  $\alpha = 1/(1 + \mu)$  and  $\beta = \mu/(1 + \mu)$ , a closed form solution to Eq. (13) for  $p = 2$  is*

$$\psi = \beta(I - \alpha D^{-1/2} W D^{-1/2})^{-1} y. \quad (18)$$

The update rule (17) can be intuitively thought of as an analogue to heat diffusion process, similar to the standard graph case (Zhou and Schölkopf 2006). At each step, every vertex is affected by its neighbors, which is normalized by the relationship among any number of nodes. At the same time, the neighbors also retains some fraction from their effects. The relative amount by which these updates occur is specified by the coefficients defined in Eq. (17).

### Physical Interpretation of Hypergraph Regularization

In standard graph cases, regularization with the graph Laplacian can be explained as an analogue to the continuous Dirichlet problem (Grady and Schwartz 2003), which is widely used in physics, particularly in fluid dynamics. To avoid confusion, the continuous calculus operators are referred to when (c) is superscripted. The Dirichlet integral is defined as

$$S_p^{(c)}(\psi) = \int_{\Omega} \|\nabla^{(c)}\psi\|^p d\Omega, \quad (19)$$

and is minimized when the Laplace equation

$$\Delta^{(c)}(\psi) := \operatorname{div}^{(c)}(\|\nabla^{(c)}\psi\|^{p-2}\nabla^{(c)}\psi) = 0 \quad (20)$$

is satisfied (Courant and Hilbert 1962). The parameter  $p$  is a coefficient for characteristics of viscosity of fluid. The function  $\phi$  satisfying the Laplace equation is called a harmonic function. Solving Eq. (19) with a boundary condition makes it possible to find a plausible interpolation between the boundary points.

From a physical standpoint, finding the shape of an elastic membrane is well approximated by the Dirichlet problem. One may think about a rubber sheet fixed along its boundary, and hung down by gravity. This setting can be written as Dirichlet problem, and the solution would give the most stable form of a rubber sheet, whose characteristics of elasticity is represented by  $p$ . To solve this numerically, we have to discretize this continuous function. With the pairwise effect between the nodes, solving the Dirichlet problem over a standard graph can be thought of as finding a plausible surface over the graph.

In the graph setup, we can say that solving the Dirichlet problem over a standard graph corresponds to finding a plausible surface over the graph which favors boundary condition  $y$ . For the hypergraph setting, solving the hypergraph Dirichlet problem gives a plausible surface with the boundary  $y$ , and  $p$  is a parameter for hypergraph; this is achieved by considering not only the pairwise effects, but also the interactions among any number of nodes. In fact, if we discretize the continuous domain of definition into lattice and consider the effect of the next neighbor, the second-order differential operator is given by  $D - W$  when  $p = 2$ . Interestingly, if we set up the lattice as a hypergraph, which means that we take into account any number of neighbors at the same time, the second-order differential operator is also  $D - W$ .

## Hypergraph Cut

### Revisiting the Hypergraph Two-class Cut

From the discussion so far, it is to be expected that there exists a relationship between hypergraph spectral theory and the

considered manifold setup. Similarly to the case of standard graph and Zhou's hypergraph Laplacian, we now introduce the hypergraph cut problem that has a connection to our Laplacian, whereby a hypergraph can be partitioned into two disjoint sets,  $A$  and  $B$ ,  $A \cup B = V$ , and  $A \cap B = \emptyset$ . The normalized hypergraph cut can now be formulated as a minimization problem given by

$$Ncut(A, B) = \partial V(A, B) \left( \frac{1}{\operatorname{vol}(A)} + \frac{1}{\operatorname{vol}(B)} \right), \quad (21)$$

where

$$\partial V(A, B) := \sum_{u \in A, v \in B} \sum_{e \in E_{un}: u, v \in e} \frac{w(e)}{\delta_e - 1} = \sum_{u \in A, v \in B} w(u, v), \quad (22)$$

and  $\operatorname{vol}(A) = \sum_{u \in A} d(u)$ . Note that this setting is consistent with the normalized cut problem on a standard graph. Let  $f \in \mathcal{H}(V)$ , be a  $|V|$  dimensional indicator vector function;  $f(v) = a$  if node  $v$  is in  $A$ ,  $f(v) = -b$  otherwise, where  $a = 1/\operatorname{vol}(A)$  and  $b = 1/\operatorname{vol}(B)$ . With these notations the problem (21) can be rewritten as Rayleigh quotient:

$$\min Ncut(A, B) = \frac{f^T D^{-\frac{1}{2}}(D - W)D^{-\frac{1}{2}}f}{f^T f} \quad \text{s.t. } \sqrt{d(v)}f(v) \in \{a, -b\}, f^T D \mathbf{1} = 0, \quad (23)$$

Minimizing  $Ncut$  is NP-hard, but it can be relaxed if we embed this problem in the real domain, and the solution of the relaxed problem is given by the second smallest eigenvalue of  $L$  (Golub and Van Loan 1996; Shi and Malik 1997).

This setting is somewhat different from the work by Zhou, Huang, and Schölkopf (2006) : if we replace the denominator of Eq. (22) from  $(\delta_e - 1)$  to  $\delta_e$ , then it is exactly same as (Zhou, Huang, and Schölkopf 2006). This difference from Zhou's approach allows for the proposed setting to be consistent with standard graphs and standard random walk setting whereas Zhou's setting can be seen as a case of the lazy random walk, as discussed in the supplemental material.

### Hypergraph Multiclass Cut

We shall now extend two-class cut to multiclass cuts and establish the connection between this setting and our proposed Laplacian, similarly to (Zhou, Huang, and Schölkopf 2006). In the standard graph case, multiclass clustering problem corresponds to decomposing  $V$  into  $k$  disjoint sets;  $V = \cup_{i=1}^k V_i$  and  $V_i \cap V_j = \emptyset$  for  $i \neq j$ . We shall denote this multiclass clustering by  $\Gamma_V^k = \{V_1, \dots, V_k\}$ , and formulate this problem as that of minimizing

$$Ncut(\Gamma_V^k) = \sum_{i=1}^k \frac{\partial V(V_i, V \setminus V_i)}{\operatorname{vol}(V_i)}, \quad (24)$$

where  $\partial V(V_i, V \setminus V_i)/\operatorname{vol}(V_i)$  measures the total weights of the links from  $V_i$  to other clusters in  $G$ . We denote the multiclass clustering  $\Gamma_V^k$  by a  $|V| \times k$  matrix  $X$ , where  $x(u, i) = 1$  if node  $u$  belongs to the  $i$ th cluster and 0 otherwise. This al-

lows us to rewrite the problem as

$$\begin{aligned} \min. \quad Ncut(\Gamma_V^k) &= \sum_{i=1}^k \frac{X_i^\top (D - W) X_i}{X_i^\top D X_i} \\ \text{s.t.} \quad X &\in \{1, 0\}^{N \times k}, X \mathbf{1}_k = \mathbf{1}_{|V|}. \end{aligned} \quad (25)$$

To this end, we consider relaxing the constraints  $X$  by minimizing  $Ncut(\Gamma_V^k)$  with constraints  $\tilde{Z}^\top \tilde{Z} = I_k$  where  $\tilde{Z} = D^{1/2} Z$  and  $Z = X(X^\top D X)^{(-1/2)}$ . The optimal solution to this problem is given by the eigenvectors associated with the smallest  $k$  eigenvalues of the Laplacian  $L$ . Similarly to Zhou's Laplacian, the following proposition holds.

**Proposition 11.** *Denote the eigenvalues of Laplacian  $L$  by  $\lambda_1 \leq \dots \leq \lambda_{|V|}$ , and define  $c_k(H) = \min Ncut$  s. t.  $X \in \{1, 0\}^{N \times k}, X \mathbf{1}_k = \mathbf{1}_{|V|}$ . Then  $\sum_{i=1}^k \lambda_i \leq c_k(H)$ .*

As discussed in (Zhou, Huang, and Schölkopf 2006), this result shows that the result of the real-value relaxed optimization problem gives us a lower bound for the original combinatorial optimization problem. However, it is not clear how to use the  $k$  eigenvectors to obtain  $k$  clusters. For a standard graph, applying the  $k$ -means method to the  $k$  eigenvectors heuristically performs well, and this approach can be applied to the hypergraph problem as well.

### Hypergraph $p$ -Normalized Cut

From the above discussion, one might expect that there exists corresponding hypergraph cut induced from hypergraph  $p$ -Laplacian, similarly to the graph  $p$ -Laplacian case (Bühler and Hein 2009). Since  $p$ -Laplace operator is nonlinear, we need to define eigenvalues and eigenvectors.

**Definition 12.** *Hypergraph  $p$ -eigenvalue  $\lambda_p \in \mathbb{R}$  and  $p$ -eigenvector  $\psi \in \mathcal{H}(V)$  of  $\Delta_p$  are defined by*

$$(\Delta_p \psi)(v) = \lambda_p \xi_p(\psi(v)), \text{ where } \xi_p(x) = |x|^{p-1} \text{sgn}(x). \quad (26)$$

To obtain  $p$ -eigenvector and  $p$ -eigenvalue, we consider Rayleigh quotient and the following statements follow:

**Proposition 13.** *Consider the Rayleigh quotient for  $p$ -Laplacian,*

$$R_p(\psi) = \frac{S_p(\psi)}{\|\psi\|_p^p}, \text{ where } \|\psi\|_p = \left( \sum_v \psi^p(v) \right)^{1/p}. \quad (27)$$

*The function  $R_p$  has a critical point at  $\psi$  if and only if  $\psi$  is  $p$ -eigenvector of  $\Delta_p$ . The corresponding  $p$ -eigenvalue  $\lambda_p$  is given as  $\lambda_p = R_p(\psi)$ . Moreover, we have  $R_p(\alpha\psi) = R_p(\psi)$ ,  $\forall \psi \in H(V)$  and  $\alpha \in \mathbb{R}, \alpha \neq 0$ .*

**Corollary 14.** *The smallest  $p$ -eigenvalue  $\lambda_p^{(1)}$  equals to 0, and corresponding  $p$ -eigenvector is  $D^{1/2} \mathbf{1}$ .*

Eq. (27) is analogue to the continuous nonlinear Rayleigh quotient

$$R_p^{(c)}(\psi) = \frac{\int_{\Omega} \|\nabla^{(c)} \psi\|^p d\Omega}{\int_{\Omega} \|\psi\|^p d\Omega}, \quad (28)$$

which relates to nonlinear eigenproblem.

In order to define a hypergraph cut corresponding to hypergraph  $p$ -Laplacian, let us consider  $f \in \mathcal{H}(V)$ , a  $|V|$  dimensional indicator vector function in Eq. (23). Then substituting  $f$  into Eq. (27) gives

$$\begin{aligned} \min p\text{-}Ncut(A, B) &:= \frac{f^\top D^{-\frac{1}{2}} (D_p - W_p) D^{-\frac{1}{2}} f}{\|f\|_p^p} \\ \text{s.t.} \quad \sqrt{d(v)} f(v) &\in \{a, -b\}, f^\top D \mathbf{1} = 0, \end{aligned} \quad (29)$$

which can be seen as the cut corresponding to our hypergraph  $p$ -Laplacian. The problem (29) is NP-hard, and therefore we need to consider a relaxed problem, similarly to the case of  $p=2$ . The constraints in Eq. (29) require the second eigenvector to be orthogonal to the first eigenvector. However, the orthogonal constraint is not suitable for  $p$ -eigenvalue problem, since the  $p$ -Laplacian is nonlinear and therefore eigenvectors are not necessary to be orthogonal to each other.

For  $p = 2$  case, since we see

$$\|\psi\|_2^2 = \|\psi - \frac{\langle \psi, D^{\frac{1}{2}} \mathbf{1} \rangle}{|V|} D^{\frac{1}{2}} \mathbf{1}\| = \min_{c \in \mathbb{R}} \|\psi - c D^{\frac{1}{2}} \mathbf{1}\| \quad (30)$$

by  $\langle \psi, D^{\frac{1}{2}} \mathbf{1} \rangle = 0$  for the second eigenvector, the Rayleigh quotient to get the second eigenvector  $v^{(2)}$  can be written as

$$v^{(2)} = \underset{\psi \in H(V)}{\text{argmin}} \frac{S_p(\psi)}{\min_c \|\psi - c D^{\frac{1}{2}} \mathbf{1}\|^2}. \quad (31)$$

Motivated by this, we here define the Rayleigh quotient for the second smallest  $p$ -eigenvalue as

$$R_p^{(2)}(\psi) = \frac{S_p(\psi)}{\min \|\psi - c D^{\frac{1}{2}} \mathbf{1}\|_p^p}, \quad (32)$$

This quotient is supported by the following theorem.

**Theorem 15.** *The global minimum of Eq. (32) is equal to the second smallest  $p$ -eigenvalue  $\lambda_p^{(2)}$  of  $\Delta_p$ . The corresponding  $p$ -eigenvector  $\psi_p^{(2)}$  can be obtained by  $\psi_p^{(2)} = \psi^* - c^* D^{\frac{1}{2}} \mathbf{1}$ , for any global minimizer  $\psi^*$  of  $R_p^{(2)}$ , where  $c^* = \underset{c \in \mathbb{R}}{\text{argmin}} \|\sum \psi^*(v) - \sqrt{d(v)} c\|^p$ . Moreover, we have  $R_p^{(2)}(t\psi + c) = R_p^{(2)}(\psi)$  where  $t \in \mathbb{R}, t \neq 0$ , and  $\forall c \in \mathbb{R}$ .*

Therefore, for the relaxed problem of Eq.(29), we consider the Rayleigh quotient Eq.(32). We note that, as  $R_p^{(2)}$  is not convex, optimization algorithms might have danger not to achieve the global minimum. However, since the function  $R_p^{(2)}$  is continuous for  $p$ , we can assume that the global minimizer of  $R_{p_1}^{(2)}$  and of  $R_{p_2}^{(2)}$  are close, if  $p_1$  and  $p_2$  are close. Hence, we firstly obtain the second eigenvector for  $p = 2$ , where there exist more stable algorithms to obtain eigenvectors, and use it as the initial condition for optimization algorithms for  $p \neq 2$ .

### Comparison to Existing Hypergraph Laplacians and Related Regularizer

We now compare our Laplacian with other two standard ones. Zhou, Huang, and Schölkopf (2006) have proposed the Laplacian  $L_Z = I - D_v^{-\frac{1}{2}} H W_e D_e^{-1} H^\top D_v^{-\frac{1}{2}}$  based on a

normalized cut and lazy random walk view, where the degree matrices  $D_v$  and  $D_e$  stand respectively for diagonal matrices, containing degree of nodes and edges,  $W_e$  is a diagonal matrix containing the weights of edges, and indices matrix  $H$  is a  $|V| \times |E_{un}|$  matrix whose element  $h(v, e)=1$  if node  $v$  is connected to the edge  $e$ , and 0 otherwise. In this setting the hypergraph is represented by the matrix  $HW_e D_e^{-1} H^\top$ , where weights  $W_e$  is normalized by degree of edges  $D_e$ . This Laplacian gives the same Laplacian if we consider the standard graph, except for the coefficient 1/2. This difference comes from the consistency of a lazy random walk view as explained in the supplement. Agarwal, Branson, and Belongie (2006) shows that Zhou’s Laplacian is equivalent to hypergraph star expansion in (Zien, Schlag, and Chan 1999) and (Li and Solé 1996).

Another Laplacian has been proposed under the unweighted setting by Rodriguez (2002) and is referred to as Simple Graph Method in (Zhou, Huang, and Schölkopf 2006). The hypergraph is represented by a matrix  $HW_e H^\top - D_v$  and Laplacian is defined as  $L_R = I - D_R^{-1/2} HW_e H^\top D_R^{-1/2}$ , where  $D_R$  is a diagonal matrix whose elements are  $d_R(u, u) = \sum_{v \in V} w_R(u, v)$  and  $w_R(u, v) = \sum_{e \in E_{un}: u, v \in e} w(e)$ . This view is consistent with the standard graph, but it does not consider the difference of edge degree  $\delta_e$ . Rodriguez Laplacian is theoretically equivalent to hypergraph clique expansion in (Zien, Schlag, and Chan 1999), (Bolla 1993), and (Gibson, Kleinberg, and Raghavan 2000) as shown in (Agarwal, Branson, and Belongie 2006).

Our Laplacian can be regarded as a family of Rodriguez’s Laplacian, but we normalize the weight by the edge degree  $\delta_e - 1$  when constructing Laplacian, whose interpretation is in the definition of gradient. If we consider the clique constructed by  $w(e)$ , and also to normalize  $w(e)$  by  $\delta_e - 1$ , we obtain our 2-Laplacian. Moreover, from the viewpoint of differential geometry, we obtain Rodriguez’s Laplacian by changing the denominator in definition of gradient (Def. 1) from  $\sqrt{\delta_e - 1}$  to 1.

Note that Zhou’s, Rodriguez’s, and our Laplacian can be seen to reduce a hypergraph to an ordinary graph, whose adjacency matrix  $HW_e D_e^{-1} H^\top$ ,  $HW_e H^\top - D_v$ , and  $W$  respectively. Moreover, Rodriguez’s and our Laplacian can be constructed from the graph gradient in (Zhou and Schölkopf 2006) using the graph reduced from hypergraph. However, because our hypergraph gradient is different from graph gradient, we cannot construct our hypergraph  $p$ -Laplacian from graph  $p$ -Laplacian in (Zhou and Schölkopf 2006) or another definition of graph  $p$ -Laplacian in (Bühler and Hein 2009). For example, consider an undirected hypergraph  $G$  where  $V = \{v_1, v_2, v_3\}$ ,  $E = \{e = \{v_1, v_2, v_3\}\}$  and  $w(e)=1$ ,  $p=1$  and the function  $\psi(v_1)=1$ ,  $\psi(v_2)=0$ , and  $\psi(v_3)=0$ . In this setting we get  $\Delta_1(v_1)=4/\sqrt{6}$ , and if we consider a graph reduced from hypergraph Zhou’s for  $v_1$  is  $1/2+1/2\sqrt{2}$  and unnormalized and normalized Bühler’s for  $v_1$  are both  $1/2$ , while our and Zhou’s 2-Laplacians give the same values. Details are in the supplemental material.

Hein et al. (2013) proposed a semi-supervised clustering using a  $p$ -regularizer  $S_p^{(h)} = \sum_e w(e) (\max_{v \in V} \psi(v) -$

$\min_{u \in V} \psi(u))^p$ , induced from a total variation on hypergraph, which is Lovász expansion of a hypergraph cut in Eq. (22). Moreover, they use total variation  $S_1^{(h)}$  for hypergraph cut, which favors balance while ours and the others favor to be attracted by larger hypergraph weights. This regularizer is reduced to the same one composed from graph  $p$ -Laplacian in (Bühler and Hein 2009; Tudisco and Hein 2016) when we consider a standard graph.

## Experiments

We compare the proposed hypergraph Laplacian with other hypergraph Laplacians, Zhou’s and Rodriguez’s, and Hein’s regularizer, on categorical data, where for each instance one or more attributes are given. Each attribute has small number of attribute values, corresponding to a specific category. We summarize the benchmark datasets we used in Table 1. As in (Zhou, Huang, and Schölkopf 2006), we constructed a hypergraph for each dataset, where each category is represented as one hyperedge, and set the weight of all the edges as 1. We used semi-supervised learning and clustering to classify the objects of a dataset. For fair comparison, we evaluate the performance with error rate, which is used in the previous studies on hypergraph clustering (Zhou, Huang, and Schölkopf 2006; Hein et al. 2013).

**Semi-supervised Learning.** We compared our semi-supervised learning method, shown in Eq. (17) with the existing ones using the Laplacians reported by Zhou, Huang, and Schölkopf (2006) and Rodriguez (2002). There are variety of ways to extend two-class clustering to multiclass clustering (Bishop 2007), but in order to keep the comparison simple, we conducted the experiment only on two-class datasets. The parameter  $\mu$  was chosen for all methods from  $10^k$ , where  $k \in \{0, 1, 2, 3, 4\}$  by 5-fold cross validation. We randomly picked up a certain number of labels as known labels, and predicted the remaining ones. We repeated this procedure 10 times for different number of known labels. For our  $p$ -Laplacian, we varied  $p$  from 1 to 3 with the interval of 0.1, and we show the result of  $p=2$  and the result of  $p$  giving the smallest average error for each number of known labeled points. The parameter  $p$  for Hein’s regularizer is fixed at 2, since this is recommended by Hein et al. (2013). The results are shown in Fig. 1 (a)-(d). When  $p=2$  our Laplacian almost consistently outperformed Rodriguez’s Laplacian, and showed almost the same behavior as Zhou’s. This means that normalizing hypergraph weights by the edge degree when constructing Laplacian enhances the performance. When we tuned  $p$ , our Laplacian consistently outperformed other Laplacians, and Hein’s regularizer except the mushroom. The dataset mushroom might fit to the Hein’s balanced cut assumption more than the other Laplacian’s normalized cut assumption. Table 2 shows the values of  $p$  which give the first and the second smallest average error. We can observe that  $p$  giving the optimal error to each number of known label points would give close value to other points. This result implies that  $p$  is a parameter for each hypergraph, rather than a parameter for the number of known label points. This can be seen as the analogue to fluid dynamics, where  $p$  is a coefficient for characteristics of viscosity of each fluid.

Table 1: Dataset summary. All datasets were taken from UCI Machine Learning Repository.

	mushroom	cancer	chess	congress	zoo	20 newsgroups	nursery
# of classes	2	2	2	2	7	4	5
$ V $	8124	699	3196	435	101	16242	12960
$ E $	112	90	73	48	42	100	27
$\sum_{e \in E}  e $	170604	6291	115056	6960	1717	65451	103680

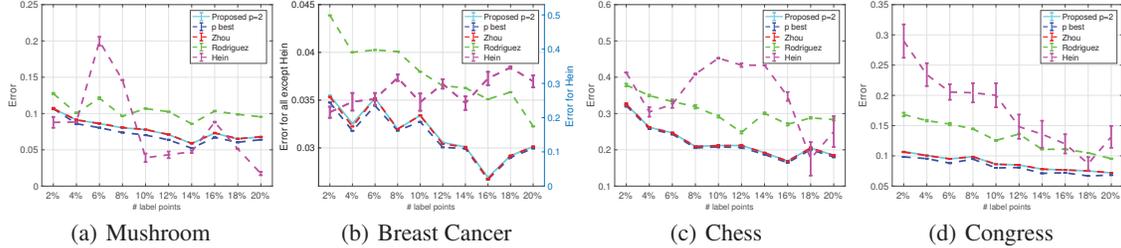


Figure 1: Results of semi-supervised learning from proposed method and the state-of-the-art

Table 2: The value of  $p$  which gives an optimal error.

Dataset	error	The portion of # of known label points									
		2%	4%	6%	8%	10%	12%	14%	16%	18%	20%
Mushroom	smallest	1.6	1.6	2.3	1.5	1.3	1.3	1.7	1.7	1.0	1.4
	2nd smallest	2.0	2.1	2.4	1.6	1.4	1.4	1.6	1.4	1.1	1.7
Breast Cancer	smallest	2.7	2.4	2.7	2.7	2.7	2.7	2.5	2.5	2.6	2.6
	2nd smallest	2.6	2.1	2.4	2.6	2.6	2.6	2.6	2.6	2.7	2.7
Chess	smallest	1.2	1.0	1.0	1.0	1.0	1.1	1.0	1.9	1.0	1.0
	2nd smallest	1.1	1.1	1.1	1.5	1.1	1.2	1.2	2.0	2.4	1.1
Congress	smallest	1.1	1.6	1.3	1.5	1.1	1.1	1.1	2.9	1.1	1.1
	2nd smallest	1.2	1.7	1.4	1.6	1.2	1.2	1.2	3.0	1.2	1.2

**Clustering.** This experiment aimed to evaluate the proposed Laplacian on a clustering task. We performed two-class and multiclass clustering tasks by solving the normalized cut eigenvalue problem of the Laplacian  $L$  for  $p = 2$ . For our  $p$ -Laplacian, we obtain the second eigenvector of Eq.(27) by varying  $p$  from 1 to 3 with the interval 0.1, and showed the optimal result. In multiclass task experiments, we used the  $k$ -means method for the obtained  $k$  eigenvectors from  $L$ , and the true number of clusters as the number of clusters. To keep the comparison simple, we conducted experiments for  $p$ -Laplacian only on twoclass problem for the same reason as semi-supervised problem. For comparison, we present the results obtained from the Zhou’s and Rodriguez’s Laplacian and Hein’s regularizer for normalized cut, and compared the error rates of the clustering results, as summarized in Table 3. Among the Laplacians, we can observe that our  $p$ -Laplacian consistently outperformed Rodriguez’s and Zhou’s Laplacian, while our 2-Laplacian showed slightly better or similar results than Rodriguez’s and Zhou’s ones. For mushroom, Hein’s is significantly better than others. This might be for the same reason in the semi-supervised learning experiment.

## Conclusion

We have proposed a hypergraph  $p$ -Laplacian from the perspective of differential geometry, and have used it to de-

velop a semi-supervised learning method in a clustering setting, and formalize them as the analogue to the Dirichlet problem. We have further explored a theoretical connection with the normalized cut, and propose a normalized cut corresponding to our  $p$ -Laplacian. Our proposed  $p$ -Laplacian has consistently outperformed the current hypergraph Laplacians on the semi-supervised clustering and the clustering tasks. There are several future directions. A fruitful future direction would be to explore extentions, such as algorithms which require less memory (Hein et al. 2013), and nodal domain theorem (Tudisco and Hein 2016). It is also worth to find more applications where hypergraph is used such as (Huang, Liu, and Metaxas 2009; Liu, Latecki, and Yan 2010; Klamt, Haus, and Theis 2009) and where hypergraph Laplacian is the most effective approach compared to the other machine learning approaches. In addition, it would be valuable if we choose the best parameter  $p$ , especially in the clustering case where we have to assume that no labelled data is available. Moreover, it would be interesting to explore a theoretical connection between hypergraph Laplacian and continuous Laplacian, like in the case of graph where the graph Laplacian is shown to converge to continuous Laplacian (Belkin and Niyogi 2003).

**Acknowledgments.** We wish to thank Daiki Nishiguchi for useful comments. This research is supported by JST ER-

Table 3: The experimental result on clustering: error rate of clustering. For the result of proposed  $p$ , we attached the value of  $p$  giving the optimal value in the parentheses next to the error value.

	Two-Class				Multiclass		
	mushroom	cancer	chess	congress	zoo	20 newsgroups	nursery
Proposed $p$	0.2329 (1)	<b>0.0243</b> (1.5)	<b>0.2847</b> (2.6)	<b>0.1195</b> (2.5)	-	-	-
Proposed $p = 2$	0.3156	0.0286	0.4775	0.1241	0.2287	<b>0.3307</b>	<b>0.2400</b>
Zhou's	0.3156	0.0300	0.4925	0.1241	0.1975	<b>0.3307</b>	0.2426
Rodriguez's	0.4791	0.3419	0.4931	0.3885	0.5376	0.4318	0.2607
Hein's	<b>0.1349</b>	0.3362	0.4778	0.3034	<b>0.1881</b>	0.5113	0.5131

ATO Kawarabayashi Large Graph Project, Grant Number JPMJER1201, Japan, and by JST CREST, Grant Number JPMJCR14D2, Japan.

## References

- Agarwal, S.; Branson, K.; and Belongie, S. 2006. Higher order learning with graphs. In *Proc. ICML*, 17–24.
- Belkin, M., and Niyogi, P. 2003. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Comput.* 15(6):1373–1396.
- Berge, C. 1984. *Hypergraphs: combinatorics of finite sets*, volume 45. Elsevier.
- Bishop, C. M. 2007. *Pattern Recognition and Machine Learning*. Springer.
- Bolla, M. 1993. Spectra, euclidean representations and clusterings of hypergraphs. *Discrete Math.* 117(1-3):19–39.
- Bougleux, S.; Elmoataz, A.; and Melkemi, M. 2007. Discrete regularization on weighted graphs for image and mesh filtering. In *Proc. SSMV*, 128–139.
- Branin, F. H. 1966. The algebraic topological basis for network analogies and the vector calculus. In *Proc. Symposium on Generalized Networks*, 452–491.
- Bühler, T., and Hein, M. 2009. Spectral clustering based on the graph  $p$ -Laplacian. In *Proc. ICML*, 81–88.
- Bulò, S. R., and Pelillo, M. 2009. A game-theoretic approach to hypergraph clustering. In *Proc. NIPS*, 1571–1579.
- Cooper, J., and Dutle, A. 2012. Spectra of uniform hypergraphs. *Linear Algebra Appl.* 436(9):3268–3292.
- Courant, R., and Hilbert, D. 1962. *Methods of Mathematical Physics Volume 2. Methods of Mathematical Physics*. Interscience Publishers.
- Ghoshdastidar, D., and Dukkipati, A. 2014. Consistency of spectral partitioning of uniform hypergraphs under planted partition model. In *Proc. NIPS*, 397–405.
- Gibson, D.; Kleinberg, J.; and Raghavan, P. 2000. Clustering categorical data: An approach based on dynamical systems. *The VLDB Journal* 8(3-4):222–236.
- Golub, G. H., and Van Loan, C. F. 1996. *Matrix Computations (3rd Ed.)*. Baltimore, MD, USA: Johns Hopkins University Press.
- Grady, L., and Schwartz, E. L. 2003. Anisotropic interpolation on graphs: The combinatorial Dirichlet problem. Technical report, Boston University.
- Grady, L. 2006. Random walks for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* 28(11):1768–1783.
- Hein, M.; Setzer, S.; Jost, L.; and Rangapuram, S. S. 2013. The total variation on hypergraphs - learning on hypergraphs revisited. In *Proc. NIPS*, 2427–2435.
- Hu, S., and Qi, L. 2015. The Laplacian of a uniform hypergraph. *J. Comb. Optim.* 29(2):331–366.
- Huang, Y.; Liu, Q.; and Metaxas, D. 2009. Video object segmentation by hypergraph cut. In *Proc. CVPR*, 1738–1745.
- Klamt, S.; Haus, U.-U.; and Theis, F. 2009. Hypergraphs and Cellular Networks. *PLoS Comput. Biol.* 5(5):e1000385+.
- Li, W.-C. W., and Solé, P. 1996. Spectra of regular graphs and hypergraphs and orthogonal polynomials. *Europ. J. Combinatorics* 17(5):461 – 477.
- Liu, H.; Latecki, L. J.; and Yan, S. 2010. Robust clustering as ensembles of affinity relations. In *Proc. NIPS*, 1414–1422.
- Meila, M., and Shi, J. 2001. A random walks view of spectral segmentation. In *Proc. AISTATS*.
- Rodriguez, J. A. 2002. On the Laplacian eigenvalues and metric parameters of hypergraphs. *Linear Multilinear Algebra* 50(1):1–14.
- Shi, J., and Malik, J. 1997. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* 22:888–905.
- Tan, S.; Guan, Z.; Cai, D.; Qin, X.; Bu, J.; and Chen, C. 2014. Mapping users across networks by manifold alignment on hypergraph. In *Proc. AAAI*, 159–165.
- Tudisco, F., and Hein, M. 2016. A nodal domain theorem and a higher-order Cheeger inequality for the graph  $p$ -Laplacian. *arXiv:1602.05567*.
- von Luxburg, U. 2007. A tutorial on spectral clustering. *Stat. Comput.* 17(4):395–416.
- Yu, S. X., and Shi, J. 2003. Multiclass spectral clustering. In *Proc. ICCV*, 313–319.
- Zhou, D., and Schölkopf, B. 2006. Discrete regularization. In *Semi-supervised Learning*. MIT Press.
- Zhou, D.; Huang, J.; and Schölkopf, B. 2006. Learning with hypergraphs: Clustering, classification, and embedding. In *Proc. NIPS*, 1601–1608.
- Zien, J. Y.; Schlag, M. D. F.; and Chan, P. K. 1999. Multilevel spectral hypergraph partitioning with arbitrary vertex sizes. *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.* 18(9):1389–1399.