Hypergraph Modeling via Spectral Embedding Connection: Hypergraph Cut, Weighted Kernel k-means, and Heat Kernel

Shota Saito

University College London
ssaito@cs.ucl.ac.uk

Abstract

We propose a theoretical framework of multi-way similarity to model real-valued data into hypergraphs for clustering via spectral embedding. For graph cut based spectral clustering, it is common to model real-valued data into graph by modeling pairwise similarities using kernel function. This is because the kernel function has a theoretical connection to the graph cut. For problems where using multi-way similarities are more suitable than pairwise ones, it is natural to model as a hypergraph, which is generalization of a graph. However, although the hypergraph cut is well-studied, there is not yet established a hypergraph cut based framework to model multi-way similarity. In this paper, we formulate multi-way similarities by exploiting the theoretical foundation of kernel function. We show a theoretical connection between our formulation and hypergraph cut in two ways, generalizing both weighted kernel k-means and the heat kernel, by which we justify our formulation. We also provide a fast algorithm for spectral clustering. Our algorithm empirically shows better performance than existing graph and other heuristic modeling methods.

Introduction

Graphs are widely used data representations for data that have pairwise relationships. One of the main aims for graph machine learning is clustering vertices, and the graph cut based spectral clustering is a popular method (Shi and Malik 1997; von Luxburg 2007). For clustering purposes, spectral clustering is also useful for real-valued data. We model real-valued data as a graphs by forming a vertex from each data point and an edge from pairwise similarity of each pair of data points (Goyal and Ferrara 2018). One popular modeling method uses kernel functions. The kernel has been theoretically justified; for example, dot product kernel is shown to be linked to the normalized graph cut via weighted kernel k-means (Dhillon, Guan, and Kulis 2004) and Gaussian kernel is justified via heat kernel (Belkin and Niyogi 2003).

Hypergraphs generalize graphs (Berge 1984), and hence are suitable to model data that have multi-way relationships, such as videos (Huang, Liu, and Metaxas 2009) and cells (Klamt, Haus, and Theis 2009). For hypergraphs, cut-based spectral clustering has also been established (Zhou, Huang, and Schölkopf 2006; Hein et al. 2013). Therefore, from the discussion on graphs, it is natural to model real-valued data as hypergraphs for clustering. By looking at multi-way relationships, we aim to gain better clustering results for general data as well as to model data that essentially involves multi-way relationships, such as the examples above. However, while heuristic modeling as hypergraphs has been done in several domains (Govindu 2005; Sun et al. 2017; Yu et al. 2018), we are yet to have a modeling framework that is theoretically connected to hypergraph cut problems.

This paper proposes a hypergraph modeling and its spectral embedding framework for clustering, which we theoretically connect to the established hypergraph cut problems. This framework models real-valued data as an even order m-uniform hypergraphs, all of whose edges connect m vertices. For this purpose, we propose a *biclique kernel*, which formulates multi-way similarity, by exploiting the kernel function's ability to model similarity but in a way where we expand from pairs to multiplets. We give a theoretical foundation to biclique kernel: a biclique kernel is equivalent to semi-definite even-order tensor (Thm. 1). We show that biclique kernel is theoretically connected to the established hypergraph cut problems proposed by (Zhou, Huang, and Schölkopf 2006; Saito, Mandic, and Suzuki 2018; Ghoshdastidar and Dukkipati 2015) via two problems, weighted kernel k-means and heat kernels. We provide a spectral clustering algorithm for our formulation, which is faster than existing ones ($O(n^3)$ vs. $O(n^m)$, where n is the number of data points). This speed-up allows us to model as an arbitrarily higher-order hypergraphs in a reasonable computational time. We numerically demonstrate that our algorithm outperforms the existing graph and heuristic embedding methods. Our empirical study also shows that by increasing order of a hypergraph, the performance is gained until a certain point but slightly drops from there. To our knowledge, it is first time to obtain the behavior of performance of spectral clustering using higher-order (say, $m \geq 8$) uniform hypergraph.

Our contributions are as follows; i) We provide a formulation to model real-valued data as an even order m-uniform hypergraph. ii) We show that our formulation is theoretically linked to the established hypergraph cuts in two ways, weighted kernel k-means and heat kernel. iii) We provide a fast spectral clustering algorithm. iv) We numerically show that our method outperforms the standard graph ones and existing heuristic embedding ones. *All proofs are in Appendix*

Related Work

This section reviews the related work of graph and hypergraph modeling. There are several approaches for justification of graph modeling via kernel function. Existing work shows the theoretical connection to the graph cut from the weighted kernel k-means (Dhillon, Guan, and Kulis 2004), energy minimization problem via continuous heat kernel (Belkin and Niyogi 2003), and kernel PCA (Bengio et al. 2004). Our approach follows the first two. A study on hypergraph cut has three approaches. One way is a graph reduction way (Agarwal, Branson, and Belongie 2006), which also works for non-uniform hypergraphs. There are three variants of this; star (Zhou, Huang, and Schölkopf 2006), clique (Rodriguez 2002; Saito, Mandic, and Suzuki 2018), and inhomogeneous (Li and Milenkovic 2017; Veldt, Benson, and Kleinberg 2020; Liu et al. 2021). Other ways are total variation (Hein et al. 2013; Li and Milenkovic 2018) and tensor modeling for uniform hypergraph (Hu and Qi 2012; Chen, Qi, and Zhang 2017; Chang et al. 2020; Ghoshdastidar and Dukkipati 2014, 2015). Our approach follows star and clique ways as well as tensor and its graph reduction approach of (Ghoshdastidar and Dukkipati 2015). We also connect ours to the inhomogeneous way. Comparing to the production of hypergraph cut objectives as above, ways of modeling as hypergraphs have received less attention. There are various studies to model real-valued data as hypergraphs by heuristic ways (Govindu 2005; Sun et al. 2017; Yu et al. 2018). However, to our knowledge, no studies developed a hypergraph cut-based framework to model real-valued data as hypergraphs. Moreover, for hypergraph connection, Whang et al. (2020) considers weighted kernel k-means, but they consider a naive connection between reduced contracted graphs and the standard kernel. Also, Louis (2015) and Ikeda et al. (2018) consider discrete heat equation, which is connected to random walk. However, those three are different to ours since they do not intend to formulate multi-way relationships.

Tensors and Uniform Hypergraphs

This section introduces notations of tensors and hypergraphs. We define an m-order tensor as $\mathcal{A} \in \mathbb{R}^{n_1 \times \ldots \times n_m}$, whose $(i_1, i_2, \ldots, i_m)$-th element is $a_{i_1 i_2 \ldots i_m} \in \mathbb{R}$. If all the dimensions of an m-order tensor $\mathcal{A}$ are identical, i.e., $n_1 = \ldots = n_m = n$, we call this tensor as *cubical*. Letting $\mathfrak{S}_m$ be a set of permutations σ on $\{1, ..., m\}$, an even m-order cubical tensor is called as *half-symmetric* if for every elements

$$\mathcal{A}_{i_{\sigma(1)} \ldots i_{\sigma(m/2)} i_{m/2+\sigma'(1)} \ldots i_{m/2+\sigma'(m/2)}} = \mathcal{A}_{i_{m/2+\sigma'(1)} \ldots i_{m/2+\sigma'(m/2)} i_{\sigma(1)} \ldots i_{\sigma(m/2)}}, \forall \sigma, \sigma' \in \mathfrak{S}_{\frac{m}{2}}, \quad (1)$$

see Appendix for examples. In the following, we assume a half-symmetric even order cubical tensor. We define the *mode-k product* of $\mathcal{A} \in \mathbb{R}^{n_1 \times \ldots \times n_m}$ and a vector $\mathbf{x} \in \mathbb{R}^{n_k}$ as $\mathcal{A} \times_k \mathbf{x} \in \mathbb{R}^{n_1 \times \ldots \times n_{k-1} \times 1 \times n_{k+1} \times \ldots \times n_m}$, whose element is

$$(\mathcal{A} \times_k \mathbf{x})_{i_1 \ldots i_{k-1} 1 i_{k+1} \ldots i_m} := \sum_{i_k=1}^{n_k} \mathcal{A}_{i_1 \ldots i_k \ldots i_m} x_{i_k} \quad (2)$$

We define a *contracted matrix* $A^{(m)}$ for a half-symmetric even m-order cubical tensor $\mathcal{A}$ as

$$A^{(m)} := \mathcal{A} \times_2 \mathbf{1} \times_3 \cdots \times_{\frac{m}{2}-1} \mathbf{1} \times_{\frac{m}{2}+1} \mathbf{1} \cdots \times_m \mathbf{1} \quad (3)$$

Note that $A^{(m)}$ is symmetric. For details, see (Lim 2005).

An m-uniform hypergraph, a generalized graph, can be represented by an m-order cubical tensor. A *hypergraph* G is a set of $(V, E, \mathbf{w})$, where an element of V is called a *vertex*, an element of E is called as an *edge*, and $\mathbf{w}$ is a vector $\{w(e)\}_{e \in E}$ where $w\colon E \to \mathbb{R}^+$ associates each edge with a *weight*. When all the edge contains the same number of vertices, we call *uniform*. A hypergraph is *connected* if there is a path for every pair of vertices. If an edge contains the same vertex multiple times, we call that this edge has a *self-loop*. We define an *adjacency tensor* $\mathcal{A}$ for uniform hypergraph, where we assign the weight of edge $e = \{i_1, \ldots, i_m\}$ to $(i_1, \ldots, i_m)$-th element of m-order cubical tensor. A uniform hypergraph is *half-undirected* when its adjacency tensor is half-symmetric. Note that a uniform hypergraph is half-undirected if undirected. The following assumes that a hypergraph G is uniform, connected, half-undirected, and has self-loops unless noted. We define the *degree* of a vertex $v \in V$ as $d_i = \sum_{e \in E: i \in e} w(e)$, and define a degree matrix D_v whose diagonal elements are the degree of vertices. Let $W_e \in \mathbb{R}^{|E| \times |E|}$ be a diagonal matrix, whose diagonal elements are weight of edge e. Let $H \in \mathbb{R}^{|V| \times |E_u|}$ be an *index matrix*, whose element $h(v, e) = \sqrt{\rho_{v,e}}$ if a vertex v is connected to an edge e, and 0 otherwise, where $\rho_{v,e}$ counts how many times the edge e contains the vertex v, e.g., if edge is $e = (v, v, v_1, v_2)$ for 4 uniform hypergraph, $\rho_{v,e} = 2$. Other than this tensor way, there is another way to represent hypergraphs as *adjacency matrix*, which contracts hypergraphs into graphs. There have been three popular ways for this, star (Zhou, Huang, and Schölkopf 2006) and two variants of clique methods (Rodriguez 2002; Saito, Mandic, and Suzuki 2018). In terms of clustering for half-undirected uniform hypergraph, which is our focus, these three different methods produce the same result (see Appendix). This paper uses the star method, which contracts a hypergraph into a graph by forming $A_s := H W_e H^\top / m$.

Formulation of Multi-way Similarity

This section proposes a formulation of multi-way similarity and discusses its properties. Looking back at a pairwise similarity, kernel function is a convenient tool to model a similarity. However, kernel functions consider pairwise similarities, not multi-way similarities. The idea to construct a multi-way similarity framework is that we take the benefits of the kernel framework's modeling ability, but at the same time, we expand to multiplets from pairs.

Biclique Kernel and Tensor Semi-definitness

This section formulates multi-way similarity as a *biclique kernel* and discusses its semi-definite property. For two sets of $m/2$ variables, $\{\mathbf{x}_{i\cdot}\}$ and $\{\mathbf{t}_{l\cdot}\}$, $\mathbf{x}_{i\cdot}, \mathbf{t}_{l\cdot} \in \mathbf{X}$, $\mathbf{X} \subseteq \mathbb{R}^d$, we formulate even m multi-way similarity function $\kappa^{(m)}(\mathbf{x}_{i_1}, .., \mathbf{x}_{i_{m/2}}, \mathbf{t}_{l_1}, .., \mathbf{t}_{l_{m/2}}) : \mathbf{X}^{m/2} \times \mathbf{X}^{m/2} \to \mathbb{R}$ as

$$\kappa^{(m)}(\{\mathbf{x}_{i\cdot}\}, \{\mathbf{t}_{l\cdot}\}) := \sum_{\gamma=1}^{m/2} \sum_{\nu=1}^{m/2} \kappa(\mathbf{x}_{i_\gamma}, \mathbf{t}_{l_\nu}), \quad (4)$$

where $\kappa : \mathbf{X} \times \mathbf{X} \to \mathbb{R}$ is a standard kernel. We call κ as a *base kernel*. By construction, $\kappa^{(m)}$ is also a kernel. Therefore, we call $\kappa^{(m)}$ as *biclique kernel*. Let $\mathcal{K}$ be a *gram tensor*

of $\kappa^{(m)}$, i.e., an m-order cubical tensor formed by Eq. (4), whose $(i_1,...,i_m)$-th element is $\kappa^{(m)}(\mathbf{x}_{i_1},...,\mathbf{x}_{i_m})$. Note that $\mathcal{K}$ is half-symmetric due to the construction of $\kappa^{(m)}$. Seeing Eq. (4), we can obtain arbitrary even m order biclique kernel from a standard kernel function κ.

The biclique kernels are connected to the semi-definite even order tensors, which serves as a theoretical ground of the biclique kernel. For the standard kernel, a gram matrix for a kernel function is equivalent to a semi-definite matrix (Shawe-Taylor, Cristianini et al. 2004). This characteristic is one of the theoretical foundations of kernel function. Here, we establish a generalization of this characteristics for the gram tensor $\mathcal{K}$. We begin with the definition of semi-definiteness of even-order tensors. An even m-order cubical tensor $\mathcal{A}$ is *semi-definite* if $\mathcal{A}\times_1\mathbf{x}...\times_m\mathbf{x}\geq 0$, $\forall\mathbf{x}\in\mathbb{R}$. Note that the semi-definiteness can be applied only to even order tensors since no odd-order tensors satisfy this semi-definiteness (see Appendix). For this semi-definiteness of tensors, the following theorem for a tensor formed by a biclique kernel holds.

Theorem 1. *Given a function $\kappa^{(m)}:\mathbf{X}^{m/2}\times\mathbf{X}^{m/2}\to\mathbb{R}$ defined by $\kappa^{(m)}(\{\mathbf{x}_{i_\cdot}\},\{\mathbf{t}_{l_\cdot}\})=\sum_{\gamma,\nu}\kappa(\mathbf{x}_{i_\gamma},\mathbf{t}_{l_\nu})$, where κ is a function $\kappa:\mathbf{X}\times\mathbf{X}\to\mathbb{R}$, then κ can be decomposed as $\kappa(\mathbf{x},\mathbf{z})=\langle\psi(\mathbf{x}),\psi(\mathbf{z})\rangle$ if and only if $\kappa^{(m)}$ is half-symmetric and has the m-order tensor semi-definite property.*

This theorem gives a theoretical foundation of the biclique kernel. Thm. 1 shows that a half-symmetric even-order semi-definite tensor and a biclique kernel are equivalent, which is similar to the foundations of the standard kernel function.

Contraction of Biclique Kernel

Despite of the nice property of Thm. 1, tensors are practically hard to work with. Many tensor problems of generalized common operations to matrix are NP-hard (Hillar and Lim 2013), such as computing eigenvalues. This motivates us to explore a practically easy while theoretical guaranteed way to deal with biclique kernel. This section argues that a contracted matrix of a gram tensor can address this issue.

We consider a contacted matrix $K^{(m)}$ (defined in Eq. (3)) of a gram tensor $\mathcal{K}$ of the biclique kernel $\kappa^{(m)}$. We call this contracted matrix $K^{(m)}$ as a *gram matrix* of $\kappa^{(m)}$. In the following, we see this gram matrix is more computationally efficient while equivalent to the original biclique kernel. We first observe the following lemma and corollary by contracting a gram tensor into a gram matrix.

Lemma 1. *Assume $\kappa(\mathbf{x},\mathbf{z})=\langle\psi(\mathbf{x}),\psi(\mathbf{z})\rangle_\kappa$ is a base kernel of the biclique kernel $\kappa^{(m)}$. Let $\psi_i:=\psi(\mathbf{x}_i)$, and $\Psi:=\sum_{l=1}^n\psi_l/n$. The gram matrix $K^{(m)}$ of $\kappa^{(m)}$ is equal to a gram matrix formed by a kernel $\kappa':X\times X\to\mathbb{R}$ as*

$$\frac{\kappa'(\mathbf{x}_i,\mathbf{x}_j)}{n^{m-2}}:=\Big\langle\psi_i+\frac{m-2}{2}\Psi,\psi_j+\frac{m-2}{2}\Psi\Big\rangle_\kappa. \quad (5)$$

Corollary 1. *The gram matrix $K^{(m)}$ is semi-definite.*

From this lemma, we observe that $K^{(m)}$ is more computationally efficient than $\mathcal{K}$ of the following reason. Computing Eq. (5), we can rewrite $K^{(m)}$ by using the gram matrix K of the base kernel κ as

$$\frac{K^{(m)}_{ij}}{n^{m-2}}=K_{ij}+\frac{m-2}{2n}(\delta_i+\delta_j)+\frac{(m-2)^2}{4n^2}\rho \quad (6)$$

where δ_i is the sum of i-th row of K and ρ is a sum of all the elements of K, i.e., $\rho=\sum_{i,j}K_{ij}$. Since we can pre-compute δ_i, δ_j and ρ from K in $O(n^2)$, the overall computational time for $K^{(m)}$ is $O(n^2)$, whereas $O(n^m)$ if we naively form $K^{(m)}$ from the original tensor and Eq. (3). Note that if we see $K^{(m)}$ as a graph, its degree matrix is equal to a degree matrix D_v of a hypergraph formed by $\mathcal{K}$. Using this lemma, we obtain the following proposition about equivalence of $\mathcal{K}$ and $K^{(m)}$.

Proposition 1. *There exists only one kernel κ' from a biclique kernel $\kappa^{(m)}$. Also, we can compose only one biclique kernel $\kappa^{(m)}$ from a kernel κ' and even-order m.*

This proposition shows that a biclique kernel $\kappa^{(m)}$ and a set of a kernel function κ' and even order m are equivalent. Therefore, Prop. 1 is a theoretical guarantee to use a computationally cheaper gram matrix $K^{(m)}$ instead of a computationally expensive gram tensor $\mathcal{K}$.

Hypergraph Cut and Spectral Clustering

Similarly to the graph case, we want to ground our formulation of multi-way similarity by biclique kernel on a hypergraph cut theory. This section discusses uniform hypergraph cut, to which we aim to link our formulation later. Here we consider partitioning a hypergraph G into two disjoint vertices sets $V_1,V_2\subset V$, $V_1\cap V_2=\varnothing$. Since the hypergraph edges contain multiple vertices, a generalization from graph cut to hypergraph cut is not straightforward. The line of the research of graph contraction ways (Zhou, Huang, and Schölkopf 2006; Saito, Mandic, and Suzuki 2018) defines hypergraph cut to penalize by a balance of the number of intersected vertices in edge by a partition. More formally, following (Zhou, Huang, and Schölkopf 2006), a hypergraph cut is defined as

$$\mathrm{Cut}(V_1,V_2):=\sum_{e\in E}w(e)|e\cap V_1||e\cap V_2|/m \quad (7)$$

We define the normalized hypergraph cut problem as

$$\mathrm{NCut}(V_1,V_2):=\mathrm{Cut}(V_1,V_2)\left(\mathrm{vol}^{-1}(V_1)+\mathrm{vol}^{-1}(V_2)\right),$$

where $\mathrm{vol}(V_j)=\sum_{i\in V_j}d_i$. We extend this to k-way normalized hypergraph cut problem as

$$\mathrm{kNCut}(\{V_i\}_{i=1}^k):=\sum_{j=1}^k\mathrm{NCut}(V_j,V\backslash V_j). \quad (8)$$

We can rewrite the minimization problem of Eq. (8) as

$$\begin{aligned}&\min\mathrm{kNCut}(\{V_i\}_{i=1}^k)\\&=\min\mathrm{trace}Z^\top D_v^{-1/2}L_sD_v^{-1/2}Z\ \mathrm{s.t.}Z^\top Z=I \quad (9)\\&=\max\mathrm{trace}Z^\top D_v^{-1/2}A_sD_v^{-1/2}Z\ \mathrm{s.t.}Z^\top Z=I, \quad (10)\end{aligned}$$

where $L_s:=D_v-A_s$ is a *hypergraph Laplacian* and $Z_{ij}=(d_i/\sum_{l\in V_j}d_l)^{1/2}$ when $i\in V_j$ otherwise 0. Eq. (9) and Eq. (10) become eigenproblem if we relax Z into real-values. As discussed, there are three types of adjacency matrix for hypergraph; star and two cliques. We can define similar cuts for the other two (Saito, Mandic, and Suzuki 2018). For uniform hypergraphs, which are our focus, these three cuts would produce the same results. See Appendix for more discussion.

Algorithm 1: Spectral clustering for hypergraph embedded by generalized kernel.

Require: Data $\mathbf{X}$, κ, and m

Compute K from the base kernel kernel κ from data X.

Construct a gram matrix $K^{(m)}$ of the biclique kernel $\kappa^{(m)}$ from K by using Eq. (6).

Compute degree matrix D_v from $K^{(m)}$ and obtain top k-eigenvectors of $D_v^{-1/2}K^{(m)}D_v^{-1/2}$.

Conduct k-means to the obtained top k-eigenvectors

Ensure: The clustering result.

In the line of tensor modeling of uniform hypergraph research (Ghoshdastidar and Dukkipati 2015, 2017), k-way partitioning problem is also considered, which we refer as *GD*. Slightly changing from GD, we form an adjacency matrix A_g as a contracted matrix of $\mathcal{A}$. A change from GD is the "position" of mode-k products, i.e., GD defines a contraction as $\mathcal{A}\times_3\mathbf{1}...\times_m\mathbf{1}$. The reason for this change is that we want a contraction of half-undirected hypergraph to be symmetric. On the other hand, this change does not affect the result in GD since GD assumes undirected hypergraph and symmetric tensor and hence contraction does not change by the position of mode-k products. The clustering algorithm of GD is to solve the eigenproblem as

$$\max \mathrm{trace} Z^\top D_v^{-1/2} A_g D_v^{-1/2} Z, \mathrm{s.t.} Z^\top Z = I. \quad (11)$$

We here show the connection between these two algorithms through the following proposition.

Proposition 2. *For half-symmetric uniform hypergraphs, Eq.* (11) *and Eq.* (10) *are equivalent.*

We call solving these eigenproblems as *spectral clustering*.

Proposed Algorithm

We propose an algorithm for clustering real-valued data via modeling as an even m-uniform hypergraph and using hypergraph cut. The overall algorithm is shown in Alg. 1. The core of our algorithm is that we model real-valued data as a hypergraph by our biclique kernel (Eq. (4)) and use hypergraph spectral clustering (Prop. 2). To do this efficiently, we firstly compute $K^{(m)}$ using Eq. (6) (the first and second step of Alg. 1) and then conduct spectral clustering (the third step). The fourth step uses a simple k-means algorithm for obtained eigenvectors to decide the split points, same as the previous studies (Zhou, Huang, and Schölkopf 2006; Ghoshdastidar and Dukkipati 2015). The overall computation time of Alg. 1 is $O(n^3)$, since it takes $O(n^2)$ to compute K as well as $K^{(m)}$, and takes $O(n^3)$ to compute eigenvectors, which is equivalent to the standard graph spectral methods. Alg. 1 is faster than spectral algorithms naively using Eq. (10) (Zhou, Huang, and Schölkopf 2006; Saito, Mandic, and Suzuki 2018) and Eq. (11) (Ghoshdastidar and Dukkipati 2015) for a hypergraph formed by $\mathcal{K}$. Both of these cost $O(n^m)$ to compute $\mathcal{K}$ and $K^{(m)}$, while ours takes overall $O(n^3)$. This reduction allows us to model as an arbitrary even m-uniform hypergraphs in a reasonable computation time, e.g., for a 20-uniform hypergraph $O(n^3)$ vs. $O(n^{20})$. Therefore, Alg. 1 is as scalable as the standard graph methods in terms of n, and more scalable than the existing hypergraph methods in terms of m.

The question is, what are theoretical justifications for Alg. 1? At this point, it seems ad-hoc to model real-valued data as a hypergraph via biclique kernel for spectral clustering since we do so without any justifications. To justify Alg. 1, next two sections connect Alg. 1 to the weighted kernel k-means and explain Alg. 1 with Gaussian-type biclique kernel from a heat kernel view.

Kernel k-means and Spectral Clustering

The graph cut and the standard kernel have a connection through a trace maximization problem via weight kernel k-means (Dhillon, Guan, and Kulis 2004). This section explores a similar connection between our biclique kernel and the hypergraph cuts. To do so, we first revisit the connection for the standard case and give an alternative way of connection for *any* kernel, instead of the dot product kernel originally discussed in (Dhillon, Guan, and Kulis 2004). This way is a kernel function approach instead of an explicit feature map approach done in (Dhillon, Guan, and Kulis 2004). We generalize this way of the graph case to our biclique kernel setting. We show that this generalized weighted kernel k-means objective for our biclique kernel is equivalent to the established cut in Prop. 2, which we see as a justification of Alg. 1.

Revisiting Spectral Connection

This section revisits the claim in (Dhillon, Guan, and Kulis 2004) that weighted kernel k-means and graph cuts are connected. We here give an alternative way of connection. This alternative way allows us to handle *any* inner product kernels, while the original in (Dhillon, Guan, and Kulis 2004) only assumes the dot product kernel. We define clusters by π_j, a partitioning of points as $\{\pi_j\}_{j=1}^k$, and the weighted kernel k-means objective for this as

$$J(\{\pi_j\}_{j=1}^k) := \sum_{\mathbf{x}_i\in\pi_j,j} w(\mathbf{x}_i)\|\psi(\mathbf{x}_i) - \mathbf{m}_j\|^2, \quad (12)$$

where $\mathbf{m}_j$ is a weighted mean, which is defined as $\mathbf{m}_j := \sum_{\mathbf{x}_i\in\pi_j} w(\mathbf{x}_i)\psi(\mathbf{x}_i)/s_j$, $s_j{:=}\sum_{\mathbf{x}_i\in\pi_j} w(\mathbf{x}_i)$, and $\|\cdot\|$ is a norm induced by *any* inner product forming a kernel function $\kappa(\mathbf{x},\mathbf{y}){=}\langle\psi(\mathbf{x}),\psi(\mathbf{y})\rangle$. Let $\kappa_{ij}{:=}\kappa(\mathbf{x}_i,\mathbf{x}_j)$, $\psi_i{:=}\psi(\mathbf{x}_i)$, and $w_i{:=}w(\mathbf{x}_i)$. Using the kernel κ and its gram matrix K we can rewrite Eq. (12) as

$$\begin{aligned}
J(\{\pi_j\}_{j=1}^k) &= \sum_{i\in\pi_j,j} w_i(\|\psi_i\|^2 - 2\langle\psi_i,\mathbf{m}_j\rangle + \|\mathbf{m}_j\|^2) \\
&= \sum_{i\in\pi_j,j}\Big(w_i\kappa_{ii} - 2w_i\sum_{l\in\pi_j}\frac{w_l}{s_j}\kappa_{il} + w_i\sum_{l,r\in\pi_j}\frac{w_l w_r}{s_j^2}\kappa_{lr}\Big) \\
&= \sum_{i\in\pi_j,j} w_i\kappa_{ii} - \sum_{r,l\in\pi_j,j} w_r w_l\kappa_{rl}/s_j \quad (13) \\
&= \mathrm{trace} W^{1/2}KW^{1/2} - \mathrm{trace} YW^{1/2}KW^{1/2}Y, \quad (14)
\end{aligned}$$

where $Y_{ij}{=}(w(\mathbf{x}_i)/s_j)^{1/2}$ when $\mathbf{x}_i{\in}\pi_j$ otherwise 0, and W is a diagonal matrix whose diagonal element is w_i. To minimize Eq. (14), we want to maximize the second term because the first term is constant w.r.t. the partitioning variable Y.

Since $Y^\top Y{=}I$, maximizing the second term is taking the top k eigenvectors of $W^{1/2}KW^{1/2}$. Taking K as a graph and W as inverse of the degree matrix, Eq. (14) becomes the relaxed graph cut problem. This gives an alternative way to connect the weighted kernel k-means and the graph cut.

Spectral Connection for Multi-way Similarity

This section aims to establish a connection between our formulation of multi-way similarity and the hypergraph cut problem, similarly to the graph one. To do so, we first generalize a weighted kernel k-means for our biclique kernel. Looking at Eq. (13), the objective function of weighted kernel k-means uses the kernel function κ directly. Therefore, we consider generalizing by replacing κ in Eq. (13) to our biclique kernel. This discussion leads us to define an objective function for weighted kernel k-means for multi-way similarity as follows:

$$J'(\{\pi_j\}_{j=1}^k) := \sum_{i\in\pi_j,j}\sum_{\{i.\}\subset\pi_j} w_i\kappa^{(m)}(i,i.,i,i.) - \sum_{i,l\in\pi_j,j}\sum_{\{i.\},\{l.\}\subset\pi_j} w_iw_l\kappa^{(m)}(i,i.,l,l.)/s_j, \quad (15)$$

where we write i instead of $\mathbf{x}_i$, and write $i.$ instead of $\{\mathbf{x}_{i.}\}$, a set of $m/2{-}1$ variables. Seeing the way we form the gram matrix $K^{(m)}$ of $\kappa^{(m)}$ (Eq. (3)), we can rewrite Eq. (15) as

$$J'(\{\pi_j\}_{j=1}^k) = \sum_{\mathbf{x}\in\pi_j,j} w_iK^{(m)}_{ii} - \sum_{i,l\in\pi_j,j} w_iw_jK^{(m)}_{il}/s_j = \mathrm{trace}W^{\frac{1}{2}}K^{(m)}W^{\frac{1}{2}} - \mathrm{trace}YW^{\frac{1}{2}}K^{(m)}W^{\frac{1}{2}}Y, \quad (16)$$

where Y is defined as Eq. (14) and $K^{(m)}$ is a gram matrix of biclique kernel $\kappa^{(m)}$. Similarly to the graph case, Eq. (16) can be solved by taking top k eigenvectors of $W^{1/2}K^{(m)}W^{1/2}$.

This discussion draws a connection between hypergraph cut and biclique kernel, and justifies Alg. 1. Recall that a gram matrix $K^{(m)}$ is obtained by a contraction of a gram tensor $\mathcal{K}$. Taking a gram matrix $K^{(m)}$ as a contracted matrix from the adjacency tensor of m-uniform hypergraph and $W{=}D_v^{-1}$, where D_v is its degree matrix, Eq. (16) is equivalent to the hypergraph cut problem (Prop 2 and Eq. (7)). Thus, the hypergraph cut problem for a hypergraph formed by $\kappa^{(m)}$ is equivalent to the weighted kernel k-means objective function for $\kappa^{(m)}$ (Eq. (15)) with a particular weight. This discussion justifies Alg. 1, since Alg. 1 turns out to be equivalent to a generalization of weighted kernel k-means for $\kappa^{(m)}$. Note that since we form $\mathcal{K}$ by $\kappa^{(m)}$, elements of $\mathcal{K}$ can be negative. This contradicts the assumption that all the weight of an edge is positive. However, this can be practically resolved in a way that does not affect topological structures, e.g., by adding the same constant to all the data points. Finally, we remark that we can rewrite Eq. (15) as an Eq. (12)-style objective function. Let $\psi_i' := n^{\frac{m-2}{2}}(\psi_i + \frac{m-2}{2}\sum_{l=1}^n \psi_l/n)$. Observing Eq. (16), we can rewrite Eq. (15) as

$$J'(\{\pi_j\}_{j=1}^k) = \sum_{i\in\pi_j,j} w_i\|\psi_i^{'} - \mathbf{m}_j'\|^2, \quad (17)$$

where $\mathbf{m}_j' := \sum_{i\in\pi_j} w_i\psi_i' / \sum_{i\in\pi_j} w_i$.

Connection to inhomogeneous cut. An inhomogeneous hypergraph cut is a cut objective which assigns different costs to the different cuts of edges (Li and Milenkovic 2017; Veldt, Benson, and Kleinberg 2020). More formally, the 2-way cut Eq. (7) can be rewritten for inhomogeneous cut as

$$\mathrm{Cut}(V_1, V_1\backslash V) = \sum_{e\in E} w_e(e\cap V_1), \quad (18)$$

where w_e is called as *splitting function* for edge e and a split of e incurred by V. This extends to k-way cut in the same way as Eq. (8). Assume that $w_e(e\cap V)$ is submodular and only depends on cardinally, i.e., depends only on $|V|$. If we change in the objective Eq. (15) as

$$\kappa^{(m)}(i,i.,l,l.) \to \kappa^{(m)}_{\mathrm{inh}}(i,i.,l,l.;V_i), \quad (19)$$

where $\kappa^{(m)}_{\mathrm{inh}}$ serves as a cost of splitting function for a multi-way modeling for $w_e(e\cap V)$, the weighted kernel k-means objective style discussion can be connected to the inhomogeneous cut Eq. (18). However, since $\kappa^{(m)}_{\mathrm{inh}}$ is *not* a biclique kernel in general, we cannot apply the other theoretical results in this paper to $\kappa^{(m)}_{\mathrm{inh}}$. For more details, see Appendix.

Heat Kernels and Spectral Clustering

This section establishes a connection between heat kernel and biclique kernel to justify Alg 1. In the graph case, for a graph made from a gram matrix of Gaussian kernel formed by randomly generated data, the cut of this graph can be seen as an analog of the asymptotic case of an energy minimization problem of the single variable heat equation using Gaussian kernel as a heat kernel (Belkin and Niyogi 2003). It is also shown that the graph Laplacian converges to the continuous Laplace operator with infinite number of data points (Belkin and Niyogi 2005). We formulate a multivariate heat equation, to which we can similarly connect our biclique kernel. We show that the hypergraph cut problem converges to an asymptotic case of the energy minimization problem of this heat equation using our biclique kernel as heat kernel if the number of data points is infinite.

We define a discrete Laplacian $L^{(m)}_{t,n}$ for $m/2$ variables $\{\mathbf{x}_{i.}\}{\in}\mathbf{X}^{m/2}$, $\mathbf{X}{\subset}\mathbb{R}^d$ and a function $f{:}\mathbf{X}^{m/2}{\to}\mathbb{R}$ which is "decomposable" as $f(\{\mathbf{x}_{i.}\}){=}\sum_{\mu=1}^{m/2} f'(\mathbf{x}_{i_\mu})$, $f'{:}\,\mathbf{X}{\to}\mathbb{R}$ as

$$L^{(m)}_{t,n}f(\{\mathbf{x}_{i.}\}) := -\sum_{\{\mathbf{y}_{i.}\}} H^{(m)}_t(\{\mathbf{x}_{i.}\},\{\mathbf{y}_{i.}\})f(\{\mathbf{y}_{i.}\}) + \sum_{\{\mathbf{y}_{i.}\}} H^{(m)}_t(\{\mathbf{x}_{i.}\},\{\mathbf{y}_{i.}\})f(\{\mathbf{x}_{i.}\})(m/2)^{-1} \quad (20)$$

where $H^{(m)}_t$ is a biclique kernel formed as

$$H^{(m)}_t(\{\mathbf{x}_{i.}\},\{\mathbf{y}_{i.}\}) := \sum_{\gamma,\nu=1}^{m/2,m/2} G_t(\mathbf{x}_{i_\gamma},\mathbf{y}_{i_\nu}),$$

$$\text{where } G_t(\mathbf{x},\mathbf{y}) := \exp\left(-\|\mathbf{x}-\mathbf{y}\|^2/4t\right)/(4\pi t)^{d/2}.$$

Note that G_t is a Gaussian kernel. Note also that the coefficient $m/2$ in Eq. (20) comes from approximation of

heat equation. Also, define an energy as $S_2(H_t^{(m)}, f) := \sum_{\{\mathbf{x}_{i\cdot}\},\{\mathbf{y}_{i\cdot}\}} L_{t,n}^{(m)} f(\{\mathbf{x}_{i\cdot}\}) f(\{\mathbf{y}_{i\cdot}\})$ with proper constraints. Minimizing this energy is equivalent to the 2-way hypergraph cut problem for a hypergraph formed by $H_t^{(m)}$. See Appendix for the detail of both remarks.

We consider to relate discrete operator $L_{t,n}^{(m)}$ to continuous Laplace operator. Let us begin with the Laplace operator. Assume a compact differentiable d-dimensional manifold $\mathcal{M}$ isometrically embedded into $\mathbb{R}^N$, a set of $m/2$ variables $\{x_i\}_{i=1}^{m/2}$, $x_i \in \mathcal{M}$, abbreviated as $\{x_\cdot\}$, and a measure μ. Consider a problem to obtain a function $f : \mathcal{M}^{m/2} \to \mathbb{R}$, that minimizes $S_2(f) := \|\nabla f\|^2$ s.t. $\|f\|^2 = 1$, and f is decomposable as $f(\{x_\cdot\}) = \sum_i f'(x_i)$ using $f' : \mathcal{M} \to \mathbb{R}$. In physics analogy, we can recognize $S_2(f)$ as energy, and the problem as an energy minimization problem. This problem often appears where we want to know a profile minimizing energy, such as velocity profile in fluid dynamics (Courant and Hilbert 1962). In machine learning, ∇f can be seen to measure how close each data point is when we embed data from a manifold to the Euclidean space. Then, this problem can be thought to find a suitable mapping f best preserving locality, and hence as a clustering algorithm (Belkin and Niyogi 2003).

By using Stokes theorem, $\|\nabla f\|^2 = \langle \Delta f, f \rangle$, which rewrites this energy minimization problem as

$$\min(S_2(f) = \langle \Delta f, f \rangle) \text{ s.t. } \|f\|^2 = 1, \langle f, c1 \rangle = 0, \quad (21)$$

where c is constant. Since Laplace operator Δ is semi-definite and $\|f\|^2 = 1$ in constraint, the minimizer of Eq. (21) is given as an eigenfunction of Δf. The first eigenfunction is a constant function that maps variables $x_i \in \mathcal{M}$ to one point. To avoid this, we introduce the second constraint since the second eigenfunction is orthogonal to the first, which is constant.

We now formulate a multivariate heat equation to analyze Δf. For even m and $m/2$ variables $x_i \in \mathcal{M} \subset \mathbb{R}^d$, consider the following heat equation on a manifold $\mathcal{M}^{m/2}$ as

$$\left(\frac{\partial}{\partial t} + \Delta \right) U(t, \{x_\cdot\}) = 0, \; U(0, \{x_\cdot\}) = f(\{x_\cdot\}), \quad (22)$$

where $\{f(x_{i\cdot})\} = \sum_{\mu=1}^{m/2} f'(x_{i_\mu})$, as defined as "decomposable" in Eq. (20). Eq. (22) governs an $m/2$ variables system, which evolves by $m/2$ variables interacting with each other but the initial conditions f' only depend on one variable. The solution is given as to satisfy

$$U = \int H_t(\{x_\cdot\}, \{y_\cdot\}) U(0, \{y_\cdot\}) d\mu(y_*) \quad (23)$$

where $d\mu(y_*) := \prod_{i=1}^{m/2} d\mu(y_i)$ and H_t is a *heat kernel*. A well-known example of heat kernel is Gaussian, which gives a solution to one variable Eq. (22) when $\mathcal{M} = \mathbb{R}^n$. However, it is difficult to obtain a concrete form of heat kernel for a general manifold. For details of heat kernel, refer to (Rosenberg and Steven 1997). Since we can prove that $H_t^{(m)}$ is also a heat kernel, there exists a heat equation on manifolds $\mathcal{M}'$ and $\mathcal{M}''$, where $\mathcal{M}' = \mathcal{M}''^{m/2}$, whose solution is given as Eq. (23) using $H_t = H_t^{(m)}$. In the following, we consider the heat equation on this manifold $\mathcal{M}'$.

Using Eq. (23), we can relate the energy minimization problem to hypergraph cut and justify Alg. 1. The energy minimization problem Eq.(21) can be approximated as

$$S_2(f) = \int_{\mathcal{M}'} dx_* \langle \Delta f, f \rangle \approx \frac{1}{t} S_2(H^{(m)}, f), \quad (24)$$

with proper constraints in Eq. (21) (see Appendix for details). As discussed when we defined discrete Laplacian (Eq. (20)), the third term $S_2(H^{(m)}, f)$ is equivalent to the 2-way hypergraph cut problem using a hypergraph formed by a biclique kernel $H_t^{(m)}$ if properly treating constraints. Hence, the energy minimization problem Eq.(21) can be seen as a continuous analog to the hypergraph spectral clustering. This discussion supports our biclique kernel with Gaussian kernel and Alg. 1, since Alg. 1 with the Gaussian-type biclique kernel can be thought as an approximation of energy minimization problem Eq. (21). The key observation is that taking a different m corresponds to taking a different manifold satisfying heat equation Eq. (22). This is because the biclique kernel $H^{(m)}$ is a different heat kernel for each m, and each heat kernel has a manifold, on which Eq. (22) holds. This key observation gives an intuitive insight; choosing better m corresponds to choosing a manifold $\mathcal{M}'$ to which the given data space $\mathbf{X}$ fits better. We conclude this section by theoretically formulating the above discussion in the following theorem.

Theorem 2. *Let $\mathcal{M}' = \mathcal{M}^{m/2}$ be a manifold, on which Eq.* (22) *satisfies with solutions using $H_t^{(m)}$. Let the data points $\mathbf{x}_1, \cdots \mathbf{x}_n$ be sampled from a uniform distribution on a manifold $\mathcal{M}$, and $f \in C^\infty(\mathcal{M}')$. Putting $t_n = n^{-1/(2+\alpha)}$, where $\alpha > 0$, there exists a constant C such that*

$$\lim_{n \to \infty} C(nt_n)^{-1} L_{n,t_n}^{(m)} f(\{\mathbf{x}_{i\cdot}\}) = \Delta f(\{\mathbf{x}_{i\cdot}\}) \textit{ in probability.}$$

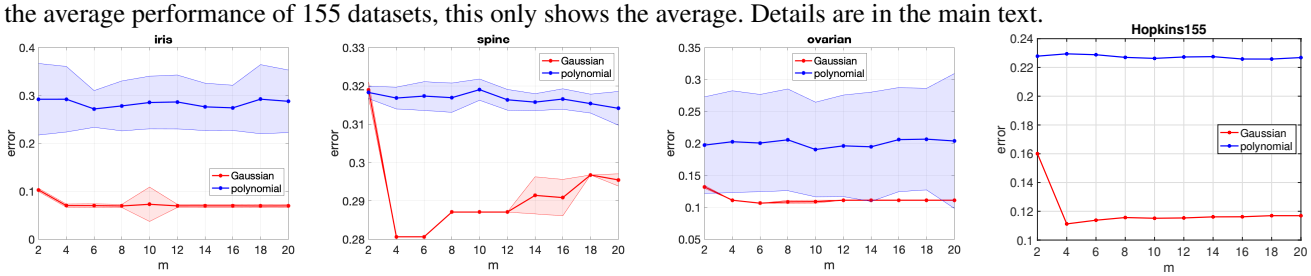
This theorem theoretically supports the discussion in this section; if we have infinite number of data, Eq. (20) converges to the continuous Laplace operator and approximation in Eq. (24) becomes exact.

Experiment

This section numerically demonstrates the performance of our Alg. 1 using our formulation of multi-way similarity with biclique kernel. We evaluated our modeling by comparing the standard kernel and other heuristic hypergraph embeddings. To focus on this purpose, we varied the embeddings and kept fixed the cut objective function as Eq. (7). Our experiments were performed on classification datasets, iris and spine from the UCI repository (Dua and Graff 2021), and ovarian cancer data (Petricoin III et al. 2002). We also used Hopkins155 dataset (Tron and Vidal 2007), which contains 155 motion segmentation datasets. We used Gaussian kernel ($\kappa(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2)$) and polynomial kernel ($\kappa(\mathbf{x}_i, \mathbf{x}_j) = (\sum_i x_i x_j + c)^d$) as a base kernel to form a biclique kernel $\kappa^{(m)}$, and conduct Alg. 1. We used $m = 2, 4..., 20$. For comparison, we employed the following types of modeling. First, we used $m = 2$, the standard graph method, for both kernels as a baseline. Second, we used adhoc modeling used in the experiment of (Ghoshdastidar and

Kernel and Method	iris	spine	Ovarian	Hopkins155
Gaussian (m=2, the standard graph)	0.1027 $\pm$ 0.0033	0.3191 $\pm$ 0.0025	0.1315 $\pm$ 0.0023	0.1600
Gaussian Ours ($m\geq$4)	**0.0693** $\pm$ 0.0033	**0.2807** $\pm$ 0.0000	**0.0841** $\pm$ 0.0000	**0.1112**
Gaussian (Ghoshdastidar and Dukkipati 2015)	0.0737 $\pm$ 0.0318	0.3000 $\pm$ 0.0000	0.1806 $\pm$ 0.0000	0.1465
Gaussian (Affine Subpace)	0.2267 $\pm$ 0.0000	0.2839 $\pm$ 0.0000	0.1690 $\pm$ 0.0023	0.1294
Gaussian (d^{H-2} (Li and Milenkovic 2017))	0.2407 $\pm$0.0662	0.3195 $\pm$ 0.0078	0.3317 $\pm$ 0.0892	0.1490
Polynomial (m=2, the standard graph)	0.2922 $\pm$ 0.0746	0.3183 $\pm$ 0.0295	0.2043 $\pm$ 0.0780	0.2278
Polynomial Ours ($m\geq$4)	0.2719 $\pm$ 0.0383	0.3142 $\pm$ 0.0452	0.1898 $\pm$ 0.0794	0.2258
Polynomial (Ghoshdastidar and Dukkipati 2015)	0.4359 $\pm$ 0.0546	0.3219 $\pm$ 0.0050	0.2817 $\pm$ 0.1201	0.2934
Polynomial (Yu et al. 2018)	0.3227 $\pm$ 0.0199	0.3828 $\pm$ 0.0754	0.4399 $\pm$ 0.0093	0.2654

Table 1: Experimental Results. The standard deviation is from randomness involved in the fourth step of Alg. 1. Since Hopkins155 is the average performance of 155 datasets, this only shows the average. Details are in the main text.



(a) Iris (b) Spine (c) Ovarian (d) Hopkins155

Figure 1: Experimental results. Red shows the result for Gaussian and blue shows for polynomial. The shade shows the standard deviation of the fourth step of Alg. 1. Since Hopkins155 is the average performance of 155 datasets, this only shows the average.

Dukkipati 2015) for both kernels. Third, we employ Gaussian-type modeling used in various papers such as (Govindu 2005; Li and Milenkovic 2017), which is the mean Euclidean distance to the optimal fitted affine subspace. Fourth, we used Gaussian-type modeling used in (Li and Milenkovic 2017), which is referred as d^{H-2}. Lastly, for polynomial, we used a generalized dot product form (Yu et al. 2018). Note that all the hypergraph comparison methods work for any uniform hypergraph. We can say that we compare five Gaussian-type methods (ours, baseline, (Ghoshdastidar and Dukkipati 2015), affine subspace, and d^{H-2}) and four polynomial-type methods (ours, baseline, (Ghoshdastidar and Dukkipati 2015), and (Yu et al. 2018)). We restrict hypergraph comparison methods to be m=3 to make the comparison fair in terms of computational time. By this, all of the comparisons and ours equally cost $O(n^3)$. For the comparisons, we also used the spectral clustering as Eq. (10), and conduct the forth step of Alg. 1. We used a free parameter $\gamma\in\{10^{-3}, 10^{-2},..., 10^{5}\}$ for Gaussian, and $d\in\{1, 3, ..., 9\}$ and c=0, 1 for polynomial. Since the fourth step of Alg. 1 involves randomness at k-means, we repeat this step 100 times. We evaluated our performance on error rate, i.e., (# of mis-clustered data points)/(# of data points), same as the previous studies (Zhou, Huang, and Schölkopf 2006; Li and Milenkovic 2017). We report average errors and standard deviations caused from the fourth step except for Hopkins155. Since Hopkins155 contains 155 tasks and standard deviations vary by each task, we only report an average error of 155 tasks, similar to the previous studies (Ghoshdastidar and Dukkipati 2014, 2017).

We summarize the results in Table 1 and Fig. 1. From Table 1, we see that ours with Gaussian kernel outperforms the other methods at all the datasets. Ours with polynomial kernel also outperforms other polynomial methods. Additionally, for most cases in Fig. 1, if we increase m, results are improved until a certain point but slightly drop from there. This corresponds to the intuition; multi-way relations could be too "multi" beyond a certain point: Too many relations could work as noise to separate the data. To our knowledge, it is first time to obtain insights on behaviors of higher-order (say, $m\geq$ 8) uniform hypergraph on spectral clustering. Moreover, for Gaussian methods, the variance for ours is smaller than one for the others. This means that our methods offer more separated embedding. Additional results are in Appendix.

Conclusion

To conclude, we have provided a hypergraph modeling method, and a fast spectral clustering algorithm that is connected to the hypergraph cut problems proposed by (Zhou, Huang, and Schölkopf 2006; Ghoshdastidar and Dukkipati 2015; Saito, Mandic, and Suzuki 2018). A future direction would be to explore other constructions of multi-way similarity which can connect to other uniform and non-uniform hypergraph cuts not having kernel characteristics, such as Laplacian tensor ways (Chen, Qi, and Zhang 2017; Chang et al. 2020), total variation and its submodular extension (Hein et al. 2013; Yoshida 2019). Also, it would be interesting to study more on connections between this work and a general splitting functions of inhomogeneous cut (Li and Milenkovic 2017; Chodrow, Veldt, and Benson 2021), e.g., to see which class of splitting functions can be connected to the biclique kernel. The limitation of our work is that we cannot apply our formulation to an odd-order uniform hypergraph. The reason for this limitation is that our biclique kernel is equivalent to half-symmetric semi-definite even-order tensor while odd-order semi-definiteness is indefinite as discussed.

Acknowledgement

We would like to thank Mark Herbster for valuable discussions.

Ethical Statement

Since this work is in the line of normalized hypergraph cut research, this work shares negative societal impacts with existing works in this line, such as (Zhou, Huang, and Schölkopf 2006; Saito, Mandic, and Suzuki 2018). As pointed out in (Hein et al. 2013), the clustering result of normalized cut tends to bias towards large weights. This could enhance existing negative biases as a clustering result.

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.

Belkin, M.; and Niyogi, P. 2005. Towards a theoretical foundation for Laplacian-based manifold methods. In *Proc. COLT*, 486–500. Springer.

Bengio, Y.; Delalleau, O.; Roux, N. L.; Paiement, J.-F.; Vincent, P.; and Ouimet, M. 2004. Learning eigenfunctions links spectral embedding and kernel PCA. *Neural Comput.*, 16(10): 2197–2219.

Berge, C. 1984. *Hypergraphs: combinatorics of finite sets*, volume 45. Elsevier.

Bishop, C. M. 2007. *Pattern Recognition and Machine Learning*. Springer.

Chang, J.; Chen, Y.; Qi, L.; and Yan, H. 2020. Hypergraph Clustering Using a New Laplacian Tensor with Applications in Image Processing. *SIAM J. Imaging Sci.*, 13(3): 1157–1178.

Chen, Y.; Qi, L.; and Zhang, X. 2017. The Fiedler vector of a Laplacian tensor for hypergraph partitioning. *SIAM J. Sci. Comput.*, 39(6): A2508–A2537.

Chodrow, P. S.; Veldt, N.; and Benson, A. R. 2021. Generative hypergraph clustering: From blockmodels to modularity. *Sci. Adv.*, 7(28): eabh1303.

Courant, R.; and Hilbert, D. 1962. *Methods of Mathematical Physics*. Methods of Mathematical Physics. Interscience Publishers.

Dhillon, I. S.; Guan, Y.; and Kulis, B. 2004. Kernel k-means: spectral clustering and normalized cuts. In *Proc. KDD*, 551–556.

Dua, D.; and Graff, C. 2021. UCI Machine Learning Repository.

Ghoshdastidar, D.; and Dukkipati, A. 2014. Consistency of Spectral Partitioning of Uniform Hypergraphs under Planted Partition Model. In *Proc. NIPS*, 397–405.

Ghoshdastidar, D.; and Dukkipati, A. 2015. A provable generalized tensor spectral method for uniform hypergraph partitioning. In *Proc. ICML*, 400–409.

Ghoshdastidar, D.; and Dukkipati, A. 2017. Uniform hypergraph partitioning: Provable tensor methods and sampling techniques. *J. Mach. Learn. Res.*, 18(1): 1638–1678.

Govindu, V. M. 2005. A tensor decomposition for geometric grouping and segmentation. In *Proc CVPR*, volume 1, 1150–1157. IEEE.

Goyal, P.; and Ferrara, E. 2018. Graph embedding techniques, applications, and performance: A survey. *Knowl. Based Syst.*, 151: 78–94.

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.

Hillar, C. J.; and Lim, L.-H. 2013. Most tensor problems are NP-hard. *J. ACM*, 60(6): 1–39.

Hu, S.; and Qi, L. 2012. Algebraic connectivity of an even uniform hypergraph. *J. Comb. Optim.*, 24(4): 564–579.

Huang, Y.; Liu, Q.; and Metaxas, D. 2009. Video object segmentation by hypergraph cut. In *Proc. CVPR*, 1738–1745.

Ikeda, M.; Miyauchi, A.; Takai, Y.; and Yoshida, Y. 2018. Finding Cheeger cuts in hypergraphs via heat equation. *arXiv preprint arXiv:1809.04396*.

Klamt, S.; Haus, U.-U.; and Theis, F. 2009. Hypergraphs and Cellular Networks. *PLoS Comput. Biol.*, 5(5): e1000385+.

Li, P.; and Milenkovic, O. 2017. Inhomogoenous hypergraph clustering with applications. In *Proc. NIPS*, 2305–2315.

Li, P.; and Milenkovic, O. 2018. Submodular Hypergraphs: p-Laplacians, Cheeger Inequalities and Spectral Clustering. In *ICML*, 3020–3029.

Lim, L.-H. 2005. Singular values and eigenvalues of tensors: a variational approach. In *Proc. CAMSAP*, 129–132.

Liu, M.; Veldt, N.; Song, H.; Li, P.; and Gleich, D. F. 2021. Strongly local hypergraph diffusions for clustering and semi-supervised learning. In *Proc. WebConf*, 2092–2103.

Louis, A. 2015. Hypergraph Markov Operators, Eigenvalues and Approximation Algorithms. In *Proc. STOC*, 713–722.

Petricoin III, E. F.; et al. 2002. Use of proteomic patterns in serum to identify ovarian cancer. *Lancet*, 359(9306): 572–577.

Qi, L. 2005. Eigenvalues of a real supersymmetric tensor. *J. Symb. Comput.*, 40(6): 1302–1324.

Rodriguez, J. A. 2002. On the Laplacian Eigenvalues and Metric Parameters of Hypergraphs. *Linear Multilinear Algebra*, 50(1): 1–14.

Rosenberg, S.; and Steven, R. 1997. *The Laplacian on a Riemannian manifold: an introduction to analysis on manifolds*. 31. Cambridge University Press.

Saito, S.; Mandic, D. P.; and Suzuki, H. 2018. Hypergraph p-Laplacian: A Differential Geometry View. In *Proc. AAAI*, 3984–3991.

Shawe-Taylor, J.; Cristianini, N.; et al. 2004. *Kernel methods for pattern analysis*. Cambridge university press.

Shi, J.; and Malik, J. 1997. Normalized Cuts and Image Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell*, 22: 888–905.

Sun, Y.; Wang, S.; Liu, Q.; Hang, R.; and Liu, G. 2017. Hypergraph embedding for spatial-spectral joint feature extraction in hyperspectral images. *Remote Sens.*, 9(5): 506.

Tron, R.; and Vidal, R. 2007. A benchmark for the comparison of 3-d motion segmentation algorithms. In *Proc. CVPR*, 1–8. IEEE.

Veldt, N.; Benson, A. R.; and Kleinberg, J. 2020. Hypergraph cuts with general splitting functions. *arXiv preprint arXiv:2001.02817*.

von Luxburg, U. 2007. A tutorial on spectral clustering. *Stat. Comput.*, 17(4): 395–416.

Whang, J. J.; Du, R.; Jung, S.; Lee, G.; Drake, B.; Liu, Q.; Kang, S.; and Park, H. 2020. MEGA: Multi-view semi-supervised clustering of hypergraphs. *Proc. VLDB*, 13(5): 698–711.

Yoshida, Y. 2019. Cheeger inequalities for submodular transformations. In *Proc. SODA*, 2582–2601. SIAM.

Yu, C.-A.; Tai, C.-L.; Chan, T.-S.; and Yang, Y.-H. 2018. Modeling multi-way relations with hypergraph embedding. In *Proc. CIKM*, 1707–1710.

Zhou, D.; Huang, J.; and Schölkopf, B. 2006. Learning with Hypergraphs: Clustering, Classification, and Embedding. In *Proc. NIPS*, 1601–1608.