An Analysis of How Hypergraph Spectral Clustering Deals with Higher-order Relationships

Ana Carolina Holzmeister Cunha
Federal University of Espirito Santo
Vitoria, Espirito Santo 29075-910
Email: anacarolholz@outlook.com

Fabiano Petronetto
Federal University of Espirito Santo
Vitoria, Espirito Santo 29075-910
Email: fabiano.carmo@ufes.br

Alcebiades Dal Col
Federal University of Espirito Santo
Vitoria, Espirito Santo 29075-910
Email: alcebiades.col@ufes.br

***Abstract*—Spectral clustering has recently been extended to hypergraphs, which are formed by vertices and higher-order relationships between these vertices. In this article, we consider a methodology to qualitatively compare hypergraph spectral clustering against the classical graph spectral clustering. More precisely, we use a graph representation to create a graph from a given hypergraph, thus allowing a comparison between spectral clustering methods. Experiment shows that the hypergraph spectral clustering deals differently with higher-order relationships.**

I. Introduction

Spectral clustering is a well-established technique for data analysis. In order to perform the analysis, a similarity graph is usually derived from data and its structure is employed in clustering [1]. Spectral clustering is widely used due to its simplicity and because it outperforms other classical methods.

Spectral clustering has recently been extended to hypergraphs, a set of vertices endowed with relations of two or more vertices. It has been called *hypergraph spectral clustering* [2].

In this work, we propose a qualitative comparison between graph spectral clustering and hypergraph spectral clustering. More precisely, hypergraph spectral clustering is applied to path and squid hypergraphs and graph spectral clustering is used in their (clique expansion) graph representations [3], where each hyperedge of theses hypergraphs is replaced by a clique (see Section IV).

II. Graph Spectral Clustering

Graphs can be used in numerous applications, such as health care, neuroscience, urban transportation, network analysis and social sciences [4]–[7]. Each edge connects two vertices and has a weight that shows the similarity between these vertices.

On mathematical terms, a *weighted graph* is defined by $G = (V, E)$, where V is a finite set of vertices whose cardinality is N and E is a set of edges that can be encoded in a weighted adjacency matrix W. If vertices v_i and v_j are connected by an edge $e = \{v_i, v_j\}$, $e \in E$, then the value w_{ij} measures the similarity between vertices v_i and v_j. The *unnormalized graph Laplacian* can be obtained by $L = D - W$, where D is the *degree matrix*, whose i-th diagonal element d_{ii} are defined by the sum of the weights of all the edges connected to the vertex v_i, $d_{ii} = \sum_j w_{ij}$.

Since the symmetric matrix L is positive semi-definite, it has N real-valued eigenvalues $0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$. The multiplicity k of the eigenvalue 0 of L equals the number of connected components in the graph. If a graph is connected, the smallest eigenvalue is $\lambda_1 = 0$ and it has as corresponding eigenvector the constant eigenvector $\mathbb{1}$.

The goal of graph spectral clustering is to ensure, as much as possible, that vertices in a same group (or cluster) are similar to each other. In other words, two vertices in the same group must have an edge with high weight and vertices in different groups should have low edge weight [1].

Graph spectral clustering with the unnormalized graph Laplacian: Given a graph $G = (V, E)$, we compute the graph unnormalized Laplacian L and the first k eigenvectors $u_1, u_2, ..., u_k$ of L, considering them organized according to a crescent order of the corresponding eigenvalues. We define a matrix U with these vectors as columns $U = [u_1\, u_2\, \ldots\, u_k] \in \mathbb{R}^{N \times k}$. For each $i = 1, \ldots, N$, we consider $y_i \in \mathbb{R}^k$ as the vector corresponding to the i-th row of U. After that, we cluster the points $(y_i)_{i=1,\ldots,n}$ into k clusters $C_1, C_2, ..., C_k$ applying the k-means algorithm to these points. As a result, we have the clusters $S_1, \ldots, S_k$ with $S_i = \{j | y_j \in C_i\}$ [1].

III. Hypergraph Spectral Clustering

While a graph can describe a pairwise relation among data, *hypergraphs* can describe the relationship between multiple data, *hyperedges* connect more than two vertices and model the polyadic interactions of the hypergraph, and the *hypergraph signals* are those associated to the vertices of the hypergraph. For example, if our data are authors and we have an edge connecting two authors if they published an article together, we would not be able to visualize if three authors published together using a graph because it is composed of only pairwise relationships, however in a hypergraph all the three authors could be in a hyperedge and we would visualize that there is an article that the three authors have written together.

A hypergraph is defined by $H = (V(H), E(H))$, where $V(H) = \{v_1, ..., v_n\}$ is a set of vertices and $E(H) = \{e_1, ..., e_k\}$ is a set of hyperedges, whose the elements are multi-elements subsets of $V(H)$. We define the maximum cardinality of the hyperedges as $M = \max\{|e_i| : e_i \in E(H)\}$. A hypergraph can be represented by matricial or tensorial algebraic descriptions. In this article, we use the tensorial representation [2].

The *adjacency tensor* of a hypergraph $H = (V(H), E(H))$, with N vertices and maximum cardinality M is an Mth-order N-dimensional tensor $\mathcal{A} \in \mathbb{R}^{N^M} = \mathbb{R}^{N\times\ldots\times N}$ with entries $a_{p_1,p_2,\ldots,p_M}$, $1 \leq p_1, p_2, ..., p_M \leq N$. A hyperedge $e_i = \{v_{l_1}, v_{l_2}, ..., v_{l_{c_i}}\} \in E(H)$ of cardinality $c_i = |e_i| \leq M$ is represented in $\mathcal{A}$ by the entries $a_{p_1,p_2,\ldots,p_M} = c_i/\alpha_i$,

$$\alpha_i = \sum_{\substack{k_1,k_2,\ldots,k_{c_i}\geq 1 \\ k_1+k_2+\ldots+k_{c_i}=M}} \frac{M!}{k_1!k_2!\cdots k_{c_i}!}, \tag{1}$$

where c_i indices in the set $\{p_1, p_2, ..., p_M\}$ are $\{l_1, l_2, ..., l_{c_i}\}$ and the other $M - c_i$ indices take into account every possible subset combination from $\{l_1, l_2, \ldots, l_{c_i}\}$ [8].

The *degree* of a hypergraph vertex $v_k \in V(H)$ is $d(v_k) = \sum_{i_2,i_3,\ldots,i_m=1}^{N} a_{k,i_2,i_3,\ldots,i_M}$, that is, the number of hyperedges containing v_k. The *degree tensor* is defined by $\mathcal{D} = d_{p_1,p_2,\ldots,p_M}, 1 \leq p_1, p_2, ..., p_M \leq N$, which is a super-diagonal tensor with diagonal elements $d_{k,\ldots,k} = d(v_k), 1 \leq k \leq N$, and zero otherwise. The *Laplacian tensor* is then defined by $\mathcal{L} = \mathcal{D} - \mathcal{A} \in \mathbb{R}^{N^M}$ [2].

Representing vectors as lowercase letters, matrices as uppercase letters and tensors as calligraphic letters, the (i, j)-th part of a 3rd-order tensor $\mathcal{A} \in \mathbb{R}^{N_1\times N_2\times N_3}$ is its tube scalar denoted by $a_{ij} = \mathcal{A}(i, j, :) \in \mathbb{R}^{1\times 1\times N_3}$, the j-th part of the tensor $\mathcal{A}$ is its lateral slides $\overrightarrow{\mathcal{A}}_j \equiv \mathcal{A}(:, j, :) \in \mathbb{R}^{N_1\times 1\times N_3}$, which is also a set of tubal scalars. The k-th part of the tensor $\mathcal{A}$ is its frontal slices denoted by $A^{(k)} \equiv \mathcal{A}(:, :, k) \in \mathbb{R}^{N_1\times N_2\times 1}$ and it forms a matrix.

Let us study the operations using the tensors. First, we have the *t-product operation*, the product of two tubal scalars results in another tubal scalar, that is, if we have the t-product between the tubal scalars $a \in \mathbb{R}^{1\times 1\times N_3}$ and $b \in \mathbb{R}^{1\times 1\times N_3}$ it will result in a tubal scalar $c \in \mathbb{R}^{1\times 1\times N_3}$ computed as $c = a \star b$, where $\star$ represents the circular convolution between two vectors.

Generalizing, the tensor $\mathcal{C} \in \mathbb{R}^{N_1\times N_4\times N_3}$ is the result of the t-product of two 3rd-order tensors $\mathcal{A} \in \mathbb{R}^{N_1\times N_2\times N_3}$ and $\mathcal{B} \in \mathbb{R}^{N_2\times N_4\times N_3}$ computed as

$$\mathcal{C} = \mathcal{A} * \mathcal{B} = fold(bcirc(\mathcal{A}) \cdot unfold(\mathcal{B})) \tag{2}$$

$$\mathcal{C} = fold\left(\begin{bmatrix} A^{(1)} & A^{(N_3)} & \cdots & A^{(2)} \\ A^{(2)} & A^{(1)} & \cdots & A^{(3)} \\ \vdots & \vdots & \ddots & \vdots \\ A^{(N_3)} & A^{(N_3-1)} & \cdots & A^{(1)} \end{bmatrix} \begin{bmatrix} B^{(1)} \\ B^{(2)} \\ \vdots \\ B^{(N_3)} \end{bmatrix}\right) \tag{3}$$

As an example of application of the t-product, we consider the following three frontal slides of 3rd-order tensors $\mathcal{A}$ and $\mathcal{B}$ and show the operations made with them to obtain the three slides of the tensor $\mathcal{C}$.

$$A^{(1)} = \begin{bmatrix} 1 & 4 & 7 \\ 1 & 4 & 4 \\ 1 & 1 & 7 \end{bmatrix}, A^{(2)} = \begin{bmatrix} 2 & 5 & 2 \\ 5 & 3 & 8 \\ 2 & 5 & 8 \end{bmatrix}, A^{(3)} = \begin{bmatrix} 3 & 8 & 9 \\ 3 & 6 & 9 \\ 7 & 6 & 6 \end{bmatrix}$$

$$B^{(1)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 7 \end{bmatrix}, B^{(2)} = \begin{bmatrix} 0 & 5 & 0 \\ 2 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, B^{(3)} = \begin{bmatrix} 0 & 0 & 9 \\ 0 & 0 & 9 \\ 3 & 6 & 0 \end{bmatrix}$$

First, we multiply the first frontal slide of $\mathcal{A}$ with the first frontal slide of $\mathcal{B}$, the second frontal slide of $\mathcal{A}$ with the last frontal slide of $\mathcal{B}$ and the last frontal slide of $\mathcal{A}$ with the second frontal slide of $\mathcal{B}$

$$A^{(1)} \times B^{(1)} = \begin{bmatrix} 1 & 16 & 49 \\ 1 & 16 & 28 \\ 1 & 4 & 49 \end{bmatrix}, A^{(2)} \times B^{(3)} = \begin{bmatrix} 6 & 12 & 63 \\ 24 & 48 & 72 \\ 24 & 48 & 63 \end{bmatrix}$$

$$\text{and } A^{(3)} \times B^{(2)} = \begin{bmatrix} 16 & 15 & 0 \\ 12 & 15 & 0 \\ 12 & 35 & 0 \end{bmatrix}.$$

We sum up the results to find the first frontal slide of $\mathcal{C}$.

$$C^{(1)} = A^{(1)} \times B^{(1)} + A^{(3)} \times B^{(2)} + A^{(2)} \times B^{(3)}$$
$$C^{(1)} = \begin{bmatrix} 23 & 43 & 112 \\ 37 & 79 & 100 \\ 37 & 87 & 112 \end{bmatrix}.$$

In order to obtain the second and the last frontal slides of $\mathcal{C}$, we perform the calculations

$$C^{(2)} = A^{(2)} \times B^{(1)} + A^{(1)} \times B^{(2)} + A^{(3)} \times B^{(3)}$$
$$C^{(3)} = A^{(3)} \times B^{(1)} + A^{(2)} \times B^{(2)} + A^{(1)} \times B^{(3)}$$

Besides the t-product operation, we have the *t-eigendecomposition*. Let $\mathcal{A} \in \mathbb{R}^{N_1\times N_1\times N_3}$ be a tensor, if each frontal slice of $\hat{\mathcal{A}} \in \mathbb{R}^{N_1\times N_1\times N_3}$ is diagonalizable, where $\hat{\mathcal{A}}$ is the discrete Fourier transform of $\mathcal{A}$ along the third dimension, that is, $\hat{\mathcal{A}}^{(k)} = \hat{V}^{(k)}\hat{\Lambda}^{(k)}(\hat{V}^{(k)})^{-1}$, then the t-eigendecomposition of $\mathcal{A}$ is given by [9]

$$\mathcal{A} = \mathcal{V} * \Lambda * \mathcal{V}^{-1}, \tag{4}$$

where Λ is an f-diagonal tensor and $\mathcal{V}$ is an orthogonal tensor.

Hypergraph spectral clustering with the Laplacian tensor: Given a hypergraph $H = (V(H), E(H))$, the Laplacian tensor $\mathcal{L}$ is calculated. We then consider the symmetrized version of the Laplacian tensor $\mathcal{L}_s \in \mathbb{R}^{N\times N\times(2N+1)}$ [8] and its discrete Fourier transform along the third dimension $\hat{\mathcal{L}}_s$.

Each frontal slice of $\hat{\mathcal{L}}_s$ is symmetric and therefore diagonalizable, then $\mathcal{L}_s$ can be t-decomposed as $\mathcal{L}_s = \mathcal{V} * \Lambda * \mathcal{V}^{-1}$. Hypergraph spectral clustering employs the first frontal slice $\hat{V}^{(1)} = [v_1\ v_2\ \ldots\ v_N] \in \mathbb{R}^{N\times N}$ of $\hat{\mathcal{V}}$ and the first frontal slice $\hat{\Lambda}^{(1)} = \text{diag}(\hat{\Lambda}^{(1)}(1, 1), \ldots, \hat{\Lambda}^{(1)}(N, N)) \in \mathbb{R}^{N\times N}$ of $\hat{\Lambda}$. More precisely, given the number k of clusters, hypergraph spectral clustering takes the matrix $U_H = [v_1\, v_2\, \ldots\, v_k] \in \mathbb{R}^{N\times k}$ and for each $i = 1, \ldots, n$ it is considered $y_i \in \mathbb{R}^k$ the vector corresponding to the i-th row of U. The points $(y_i)_{i=1,\ldots,n}$ are clustered into k clusters $C_1, C_2, \ldots, C_k$ applying the k-means algorithm to these points. Finally, the clusters $S_1, \ldots, S_k$, with $S_i = \{j|y_j \in C_i\}$, are obtained.

IV. Experiments and Results

In this section, we consider path and squid hypergraphs [10] with additional hyperedges to perform comparisons between graph and hypergraph spectral clustering. In addition, we use the (clique expansion) graph representation [3] to generate a graph from a given hypergraph. This method replaces each hyperedge with a clique, that is, every two distinct vertices in a hyperedge are connected by an edge in the graph. More precisely, given a hypergraph $H = (V(H), E(H))$, the graph representation leads to a graph $G = (V, E)$ in which $V = V(H),\ E = \{\{u, v\} \in e, e \in E(H), u \neq v\}$ [3]. The weight $w(u, v)$ of an edge $\{u, v\}$ is commonly calculated by the sum of the weight $w(e)$ of the hyperedges $e \in E(H)$ that contain vertices u and v, that is,

$$w(u, v) = \sum_{\substack{e \in E(H) \\ u, v \in e}} w(e). \tag{5}$$

In this article, we set the weight of all hyperedges equal to 1, thus $w(u, v)$ counts the number of hyperedges that contain u and v. Figure 1 depicts a path hypergraph with $N = 7$ vertices and a squid hypergraph with $N = 13$ vertices and the graph representation of these hypergraphs.

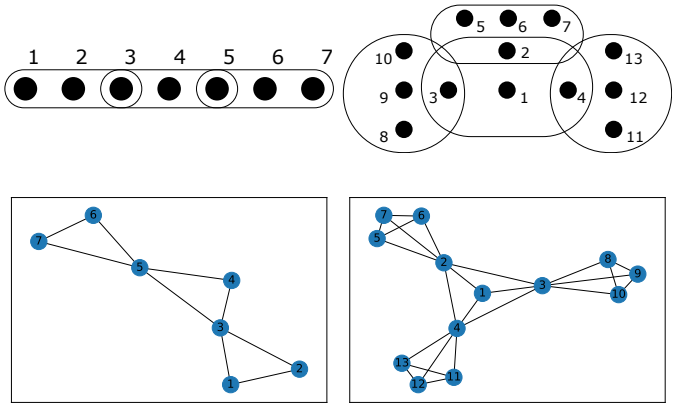


Fig. 1: Path and squid hypergraph (top line) and its graph representations (bottom line).

Figure 2 shows the graph representation of a path hypergraph with $N = 7$ vertices and an extra hyperedge $\{v_1, v_2\}$. In the graph representation, $w(v_1, v_2) = 2$ since hyperedges $\{v_1, v_2, v_3\}$ and $\{v_1, v_2\}$ contain the vertices v_1 and v_2.

In order to compare spectral clustering methods, we applied the hypergraph spectral clustering to the hypergraph and the graph spectral clustering to its graph representation with three, four, five and six clusters ($k = 3, 4, 5, 6$).

As we increase the number of clusters for the hypergraph spectral clustering, it is possible to see that the vertices in the additional hyperedge (v_1 and v_2) remained in the same cluster. On the other hand, on the graph representation, when the number of clusters for the graph spectral clustering became six ($k = 6$), the vertices v_1 and v_2 were separated into different clusters. It is easier for hypergraph spectral clustering to separate the vertices v_3 and v_5 than for graph spectral clustering. In other words, the hypergraph spectral clustering considers the connection between vertices v_1 and v_2 ($\{v_1, v_2, v_3\}$ and $\{v_1, v_2\}$) stronger than the connection between vertices v_3 and v_5 ($\{v_3, v_4, v_5\}$). This result indicates that hypergraph spectral clustering allow us to better understand the existence of a hyperedge located inside of another hyperedge $\{v_1, v_2\} \subset \{v_1, v_2, v_3\}$.

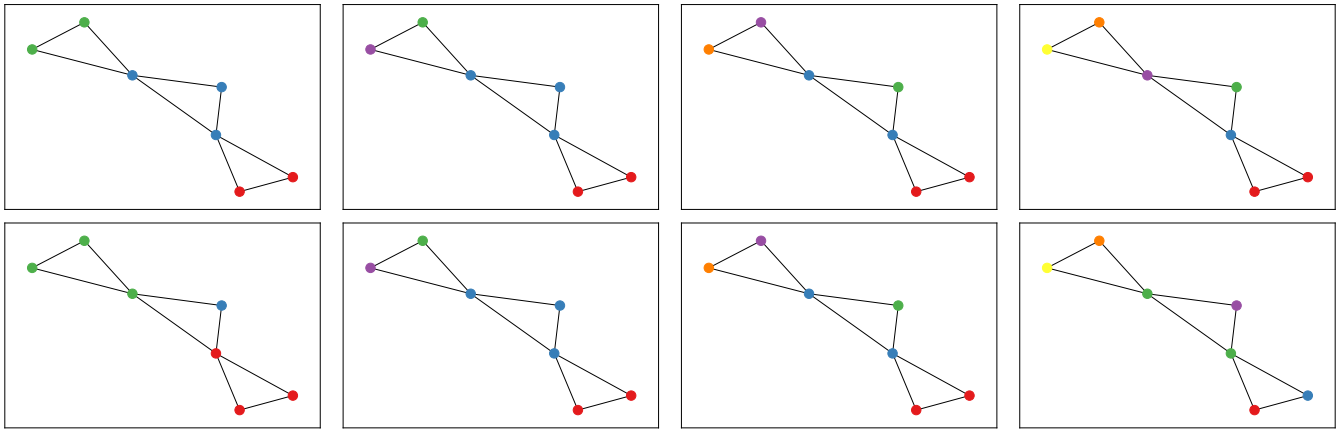
In the next experiment, we consider a squid hypergraph with $N = 13$ vertices and an extra hyperedge $\{v_2, v_3, v_8\}$. Figure 3 illustrates the graph representation of this squid hypergraph with the additional hyperedge highlighted by a triangle. Note that $w(v_2, v_3) = 2$ since hyperedges $\{v_1, v_2, v_3, v_4\}$ and $\{v_2, v_3, v_8\}$ contain the vertices v_2 and v_3, and $w(v_3, v_8) = 2$ since hyperedges $\{v_3, v_8, v_9, v_{10}\}$ and $\{v_2, v_3, v_8\}$ contain the vertices v_3 and v_8. The other edges have a weight equal to 1.

We applied graph and hypergraph spectral clustering with different numbers of clusters taking into account the squid hypergraph. When we consider $k = 3$ clusters, the hypergraph spectral clustering considers the bond of the additional hyperedge $\{v_2, v_3, v_8\}$ to be strong, then the vertex v_2 is inside the cluster with the vertices v_3, v_8, v_9 and v_{10}, while on the graph spectral clustering the vertex v_2 is not inside this cluster. In the graph representation, $w(v_2, v_5) = w(v_2, v_6) = w(v_2, v_7) = 1$, and $w(v_2, v_1) = w(v_2, v_8) = 1$ and $w(v_2, v_3) = 2$. Then, there are three connections between vertex v_2 and vertices in the green cluster (v_5, v_6 and v_7) with edge weight one and three connections between vertex v_2 and vertices in the red cluster (v_1, v_3, v_8, v_9 and v_{10}), one connection with weight two and two connections with weight one. The graph spectral clustering then prioritizes a cluster with small connection weights to vertex v_2. When $k = 4$, both graph and hypergraph spectral clustering separate the vertex v_1 into a cluster and keep the other vertices in clusters according to the hyperedges. For $k = 5$ and $k = 6$ clusters, the hypergraph spectral cluster considers the cohesion of the group with vertices v_2, v_3, v_8, v_9, v_{10} and chooses to keep them in the same cluster and separate the other clusters. On the other hand, in the $k = 5$ case, the graph spectral clustering separates the group with vertices v_3, v_8, v_9, v_{10}, and, in the $k = 6$ case, it prioritizes a cluster given by the vertices in the hyperedge $\{v_1, v_2, v_3, v_4\}$ with the addition of the vertex v_8 considering the additional connection between vertices v_2, v_3 and v_8. Furthermore, the graph spectral clustering divided vertices in the hyperedge $\{v_3, v_8, v_9, v_{10}\}$ into three different clusters.

V. Conclusion

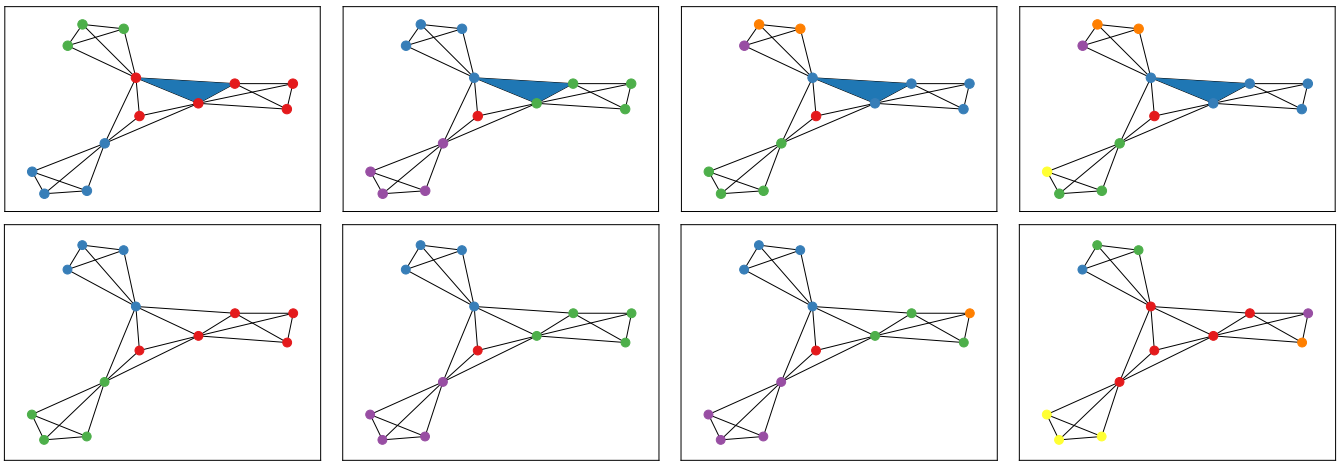
Our work innovates by comparing hypergraph spectral clustering with its graph counterpart while other works compare it with specific hypergraph clustering methods. With this, we contribute with an analysis of the novelties generated by the use of hypergraphs.

In the future, we intend to study hypergraph inference given a set of points in a high-dimensional space and then analyze the differences between the application of graph and hypergraph spectral clustering in this data analysis scenario.



(a) $k = 3$ (b) $k = 4$ (c) $k = 5$ (d) $k = 6$

Fig. 2: Hypergraph spectral clustering for a path hypergraph with additional hyperedge $\{v_1, v_2\}$ (top line) and graph spectral clustering for the graph representation of the path hypergraph with additional hyperedge $\{v_1, v_2\}$ (bottom line).



(a) $k = 3$ (b) $k = 4$ (c) $k = 5$ (d) $k = 6$

Fig. 3: Hypergraph spectral clustering for a squid hypergraph with additional hyperedge $\{v_2, v_3, v_8\}$ (top line) and graph spectral clustering for the graph representation of the squid hypergraph with additional hyperedge $\{v_2, v_3, v_8\}$ (bottom line).

ACKNOWLEDGMENT

The authors would like to thank the Fundo Nacional de Desenvolvimento da Educação (FNDE/PET). Alcebiades Dal Col and Fabiano Petronetto was supported in part by Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) under grat 442238/2023-1 and 405903/2023-5.

REFERENCES

[1] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and computing*, vol. 17, pp. 395–416, 2007.

[2] K. Pena-Pena, D. L. Lau, and G. R. Arce, "t-hgsp: Hypergraph signal processing using t-product tensor decompositions," *IEEE Transactions on Signal and Information Processing over Networks*, 2023.

[3] M. T. Schaub, Y. Zhu, J.-B. Seby, T. M. Roddenberry, and S. Segarra, "Signal processing on higher-order networks: Livin'on the edge... and beyond," *Signal Processing*, vol. 187, p. 108149, 2021.

[4] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE signal processing magazine*, vol. 30, no. 3, pp. 83–98, 2013.

[5] A. Sandryhaila and J. M. Moura, "Big data analysis with signal processing on graphs: Representation and processing of massive data sets with irregular structure," *IEEE Signal Processing Magazine*, vol. 31, no. 5, pp. 80–90, 2014.

[6] A. Ortega, P. Frossard, J. Kovačević, J. M. Moura, and P. Vandergheynst, "Graph signal processing: Overview, challenges, and applications," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 808–828, 2018.

[7] W. A. Martins, J. B. Lima, C. Richard, and S. Chatzinotas, "A primer on graph signal processing," in *Signal Processing and Machine Learning Theory*, P. S. Diniz, Ed. Academic Press, 2023, pp. 961–1008.

[8] A. D. Col, F. Petronetto, J. R. de Oliveira Neto, and J. B. Lima, "Windowed hypergraph Fourier transform and vertex-frequency representation," *Signal Processing*, p. 109538, 2024.

[9] M. E. Kilmer, K. Braman, N. Hao, and R. C. Hoover, "Third-order tensors as operators on matrices: A theoretical and computational framework with applications in imaging," *SIAM Journal on Matrix Analysis and Applications*, vol. 34, no. 1, pp. 148–172, 2013.

[10] S. Hu, L. Qi, and J.-Y. Shao, "Cored hypergraphs, power hypergraphs and their laplacian h-eigenvalues," *Linear Algebra and Its Applications*, vol. 439, no. 10, pp. 2980–2998, 2013.