Hyperedge Prediction using Tensor Eigenvalue Decomposition

Deepak Maurya, Balaraman Ravindran, Shankar Narasimhan

Robert Bosch Centre for Data Science and Artificial Intelligence, Indian Institute of Technology Madras, India

Objective

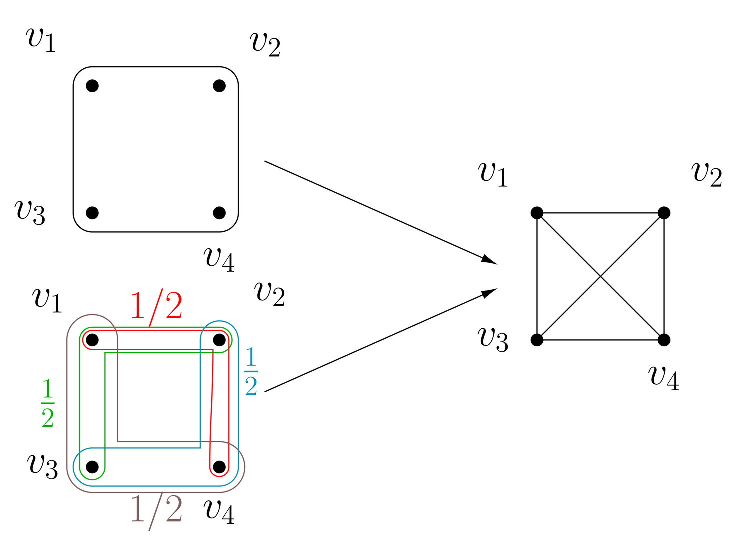
Given a **k-uniform undirected hypergraph** $G = (V, E)$, predict the **new hyperedges** which are most likely to be formed.

Hyperedge Reduction

A hypergraph $G = (V, E)$ can be represented by vertex-edge incidence matrix $\mathbf{H}$ of dimension $|V| \times |E|$ whose entry $h(i,j) = 1$ if $v_i \in e_j$ and 0 otherwise. The adjacency matrix for reduced hypergraph using clique expansion:

$$\mathbf{A}_r = \mathbf{H}\mathbf{W}\mathbf{H}^T - \mathbf{D} \quad (1)$$

where $\mathbf{D}$ is a diagonal matrix of dimension $|V| \times |V|$ containing degrees.



Hypergraph Representation

A natural representation of hypergraphs is a k-order n-dimensional tensor $\mathcal{A}$ [1], which consists of n^k entries:

$$a_{i_1 i_2 \ldots i_k} = \begin{cases} w_{e_j} \frac{1}{(k-1)!} & \text{if } (i_1, i_2, \ldots, i_k) = \{e_j\} \quad e_j \in E \\ 0 & \text{otherwise} \end{cases}$$

It should be noted that $\mathcal{A}$ is a "*super-symmetric*" tensor.The degree of a vertex v_i is given by

$$d(v_i) = \sum_{i_k=1}^{n} \ldots \sum_{i_3=1}^{n} \sum_{i_2=1}^{n} a_{i i_2 i_3 \ldots i_k} \quad (2)$$

The Laplacian tensor $\mathcal{L}$ is defined as:

$$\mathcal{L} = \mathcal{D} - \mathcal{A}$$

Spectral decomposition [2] using

$$\mathcal{L}\mathbf{x}^{k-1} = \lambda \mathbf{x}$$
$$\mathbf{x}^T \mathbf{x} = 1$$

where $(\lambda, \mathbf{x}) \in (\mathbb{R}, \mathbb{R}^n \backslash \{0\}^n)$ satifying above is called the Z-eigenpair and $\mathcal{L}\mathbf{x}^{k-1} \in \mathbb{R}^n$, whose i^{th} component is defined

$$\left[\mathcal{L}\mathbf{x}^{k-1}\right]_i = \sum_{i_k=1}^{n} \ldots \sum_{i_3=1}^{n} \sum_{i_2=1}^{n} l_{i i_2 i_3 \ldots i_k} x_{i_2} x_{i_3} \ldots x_{i_k}$$

Spectral Analysis

Tensor eigenvalue decomposition arises from:

$$\min_{\mathbf{x}} \quad \mathcal{L}\mathbf{x}^k = \sum_{i_k=1}^{n} \ldots \sum_{i_2=1}^{n} \sum_{i_1=1}^{n} l_{i_1 i_2 \ldots i_k} x_{i_1} x_{i_2} \ldots x_{i_k}$$

such that $\quad \mathbf{x}^T \mathbf{x} = 1$

The eigenvector with minimum positive λ satisfying above equation is termed as Fiedler eigenvector and can be computed by following optimization problem

$$\mathbf{v}_\star = \underset{\mathbf{x}}{\text{argmin}} \quad \mathcal{L}\mathbf{x}^k > 0,$$
$$\text{s. t} \quad \mathbf{x}^T \mathbf{x} = 1$$

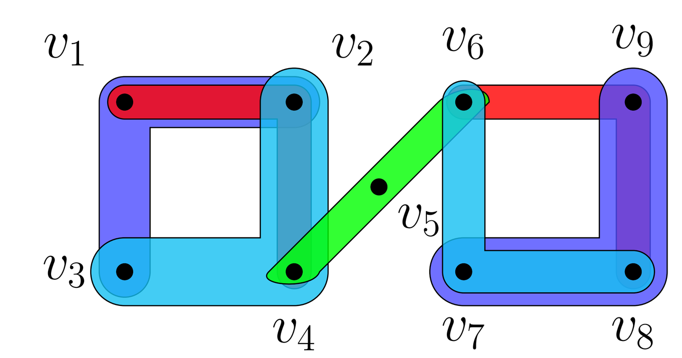
The corresponding eigenvalue can be computed as $\lambda_\star = \mathcal{L}\mathbf{v}_\star^k$.

Challenges

- Eigenvectors may not be orthogonal for symmetric tensors.
- Odd order tensor have negative eigenvalues.

Example 1

Given the following hypergraph, predict new hyperedges.



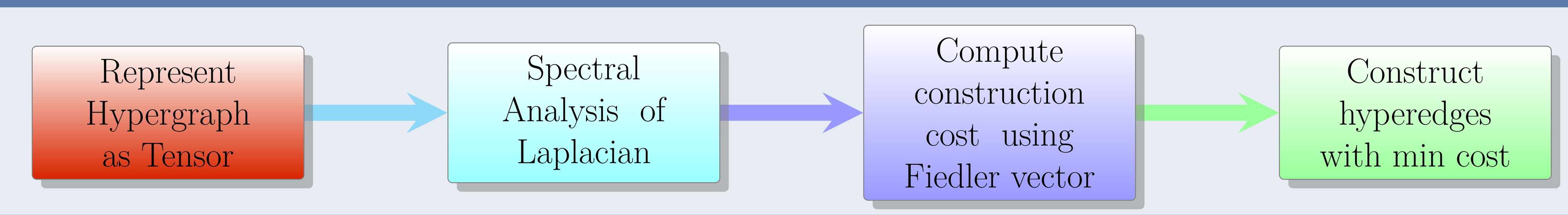
hyperedges$_1$	hyperedges$_2$	cost
{6, 7, 9}	{1, 3, 4}	0.0028
{5, 6, 8}	{2, 4, 5}	0.0139
{1, 3, 4}	{6, 7, 9}	0.0142
{5, 6, 7}	{1, 4, 5}	0.0152
{5, 6, 9}	{3, 4, 5}	0.0152
{5, 8, 9}	{1, 2, 5}	0.0195
{5, 7, 8}	{2, 3, 5}	0.0195
{5, 7, 9}	{1, 3, 5}	0.0205
{3, 4, 5}	{5, 6, 9}	0.0365
{1, 4, 5}	{3, 5, 6}	0.0379

Unnormalized Laplacian

hyperedges$_1$	hyperedges$_2$	cost
{6, 7, 9}	{1, 3, 4}	3.3×10^{-4}
{2, 3, 5}	{5, 8, 9}	0.0142
{1, 2, 5}	{5, 7, 8}	0.0160
{1, 3, 5}	{5, 7, 9}	0.0172
{1, 3, 4}	{6, 7, 9}	0.0173
{3, 4, 5}	{5, 6, 9}	0.0197
{2, 4, 5}	{5, 6, 8}	0.0254
{4, 5, 7}	{1, 5, 6}	0.0375
{4, 5, 9}	{3, 5, 6}	0.0375
{1, 4, 5}	{5, 6, 7}	0.0386

Normalized Laplacian

Proposed Algorithm



Construction Cost

Theorem: *The hypergraph Laplacian cost function for a k-uniform hypergraph can be expressed as*

$$\mathcal{L}\mathbf{x}^k = \sum_{e_j \in E} l_{e_j}(\mathbf{x})$$

$$l_{e_j}(\mathbf{x}) = w_{e_j} \left(\sum_{i_k \in e_j} x_{i_k}^k - k \prod_{i_k \in e_j} x_{i_k} \right)$$

$$= w_{e_j} k \left(\underset{i_j \in e_j}{\text{A.M}} \left(x_{i_k}^k\right) - \underset{i_j \in e_j}{\text{G.M}} \left(|x_{i_k}|^k\right) (-1)^{n_s} \right)$$

where $n_s = |\{i_j : x_{i_j} < 0\}|$, A.M and G.M stand for the arithmetic and geometric means, respectively.

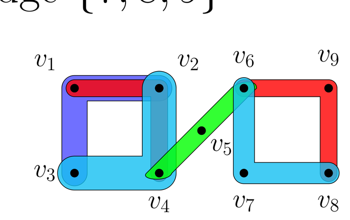
Computations reduced from $\mathbf{O}(|V|^k)$ to $\mathbf{O}(|E|)$

Example: Consider a hypergraph $G = (V, E)$ with $V = \{1, 2, 3\}$ and $E = \{\{1, 2, 3\}\}$. The laplacian corresponding to the hyperedge is given by

$$l_{e_j}(\mathbf{x}) = x_1^3 + x_2^3 + x_3^3 - 3x_1 x_2 x_3$$

Example 2

Removed the hyperedge $\{7, 8, 9\}$

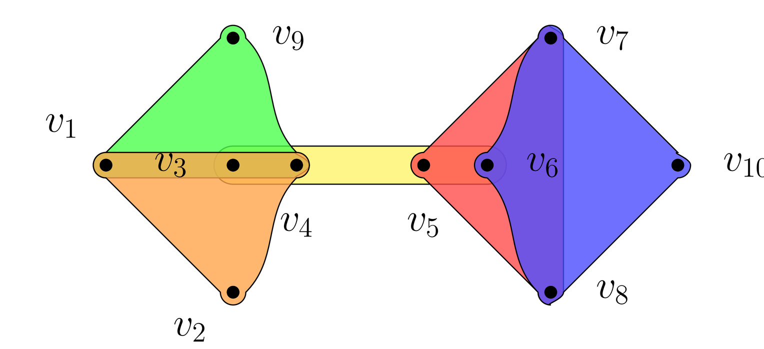


As there are 9 nodes, one could have $\binom{9}{3} = 84$. Seven hyperedges are further removed as they already exist in the hypergraph, which leaves us with $84 - 6 = 78$ potential hyperedges.

Unnormalized	Normalized
{1, 3, 4}	{1, 3, 4}
{7, 8, 9}	{7, 8, 9}
{2, 4, 5}	{5, 8, 9}
{1, 4, 5}	{5, 6, 9}
{3, 4, 5}	{5, 6, 8}
{5, 6, 9}	{5, 7, 8}
{1, 2, 5}	{5, 7, 9}
{2, 3, 5}	{6, 7, 9}
{1, 3, 5}	{1, 5, 6}
{5, 8, 9}	{3, 5, 6}

Example 3

Consider the 4-uniform hypergraph



Top 10 preferential hyperedges of the 205 potential hyperedges using the unnormalized and normalized Laplacian tensor.

Unnormalized	Normalized
{1, 2, 3, 9}	{2, 5, 6, 9}
{1, 2, 4, 9}	{1, 2, 3, 9}
{2, 3, 4, 9}	{1, 2, 4, 9}
{5, 7, 8, 10}	{1, 2, 5, 9}
{5, 6, 8, 10}	{1, 2, 6, 9}
{1, 3, 4, 5}	{5, 7, 8, 10}
{2, 3, 4, 5}	{2, 3, 4, 9}
{3, 4, 5, 9}	{1, 5, 6, 9}
{1, 2, 4, 5}	{1, 2, 5, 6}
{1, 3, 5, 9}	{2, 3, 5, 9}

Conclusions

1. The key idea of proposed algorithm is inclusion of new hyperedges such that there is minimal perturbation in the "smoothness" of the hypergraph, which is qauntified by Fiedler eigenvalue.
2. Spectral analysis using tensors representation of hypergraphs helps to draw better insights compared to hypergraph reduction approaches.
3. The normalized Laplacian attempts to grant similar weight to all the nodes by normalizing with the degree of the node.

References

[1] A. Banerjee, A. Char, and B. Mondal, "Spectra of general hypergraphs," *Linear Algebra and its Applications*, vol. 518, pp. 14–30, 2017.

[2] L. Qi and Z. Luo, *Tensor analysis: spectral theory and special tensors*, vol. 151, Siam 2017.

Contact Information

- Email: maurya@cse.iitm.ac.in, ravi@cse.iitm.ac.in, naras@iitm.ac.in
- Web: d-maurya.github.io/web/, www.cse.iitm.ac.in/ ravi/, www.che.iitm.ac.in/ naras/