A Provable Generalized Tensor Spectral Method for Uniform Hypergraph Partitioning

Debarghya Ghoshdastidar, Ambedkar Dukkipati

Department of Computer Science & Automation
Indian Institute of Science
Graph Partitioning

Graphs are ubiquitous . . .
- Real-world networks:

- Graphs constructed from data:

And partitioning is essential for network analysis / clustering.
Spectral Graph Partitioning

Input Graph

“Strongly connected” groups
Spectral Graph Partitioning

Input Graph

"Strongly connected" groups

Adjacency matrix

Find $k$ leading eigenvectors

Run $k$-means on rows
What is a hypergraph?

- Generalization of a graph
- Each edge can connect more than two nodes
- Hypergraph is \( m \)-uniform if each edge connects \( m \) nodes

Graph (2-uniform hypergraph)  3-uniform hypergraph  Non-uniform hypergraph
Hyper-Graphs are also Ubiquitous

- Hypergraph partitioning is the key to *divide and conquer* approach for *circuit design*

  [Schweikert & Kernighan '79; Karypis & Kumar '00]

- Hypergraphs also used to represent *categorical data* and *complex biological networks*

  [Gibson et al. '00; Michoel & Nachtergaele '12]

- Uniform hypergraphs model *online tagging* networks

  [Ghosal et al. '09]
And **Uniform** Hypergraphs are Essential in Vision

### Geometric Grouping:

[Agarwal et al. '05; Govindu '05]

<table>
<thead>
<tr>
<th>Well-separated</th>
<th>Overlapping</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Well-separated" /></td>
<td><img src="image2.png" alt="Overlapping" /></td>
</tr>
</tbody>
</table>

- Pairwise relations not suitable
- Need to check if \( m(\geq 4) \) points fit a circle or not
- \( m \)-way affinities naturally leads to \( m \)-uniform hypergraph

<table>
<thead>
<tr>
<th>Subspace clustering</th>
<th>Motion segmentation</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image3.png" alt="Subspace clustering" /></td>
<td><img src="image4.png" alt="Motion segmentation" /></td>
</tr>
</tbody>
</table>
Existing Algorithms

Approaches for uniform hypergraph partitioning:

- Higher-order SVD [Govindu '05]
- Hypergraph reduction by clique averaging [Agarwal et al. '05]
- Non-negative tensor factorization [Shashua et al. '06]
- Optimization with $\ell_1$-norm constraints [Liu et al. '10]
- Tensor power iterations [Duchhene et al. '11]
- Game-theoretic approach [Rota Bulo & Pelillo '13]
Normalized Associativity Maximization

Partition nodes into groups $\mathcal{V}_1, \ldots, \mathcal{V}_k$ such that it maximizes

$$N\text{-Assoc}(\mathcal{V}_1, \ldots, \mathcal{V}_k) = \sum_{j=1}^{k} \frac{\text{Assoc}(\mathcal{V}_j)}{|\mathcal{V}_j|^m/2},$$

where $\text{Assoc}(\mathcal{V}_j)$ is total weight of edges contained in $\mathcal{V}_j$. 
Normalized Associativity Maximization

Partition nodes into groups \( \mathcal{V}_1, \ldots, \mathcal{V}_k \) such that it maximizes

\[
N\text{-Assoc}(\mathcal{V}_1, \ldots, \mathcal{V}_k) = \sum_{j=1}^{k} \frac{\text{Assoc}(\mathcal{V}_j)}{\left| \mathcal{V}_j \right|^{m/2}},
\]

where \( \text{Assoc}(\mathcal{V}_j) \) is total weight of edges contained in \( \mathcal{V}_j \).

An equivalent optimization problem:

- Given \( m \)-uniform hypergraph on \( n \) nodes
- Adjacency tensor:
  
  \( m^{th} \)-order \( n \)-dimensional tensor \( \mathbf{A} \in \{0, 1\}^{n \times n \times \ldots \times n} \)
  
  \( \mathbf{A}_{i_1 i_2 \ldots i_m} = 1 \) if there is an edge connecting \( i_1, i_2, \ldots, i_m \)
  
  Tensor is symmetric
An equivalent optimization problem

Let $H$ be assignment matrix with normalized columns $A H A^T$.

Objective:
Maximize $\text{Trace}(\overline{A})$ over all cluster assignments $H$.
Existing Algorithms

Approaches for uniform hypergraph partitioning:

- Higher-order SVD (HOSVD) [Govindu '05]
- Hypergraph reduction by clique averaging [Agarwal et al. '05]
- Non-negative tensor factorization [Shashua et al. '06]
- Optimization with $\ell_1$-norm constraints [Liu et al. '10]
- Tensor power iterations [Duchhene et al. '11]
- Game-theoretic approach [Rota Bulo & Pelillo '13]

Interesting fact:

All methods other than HOSVD can be formulated as relaxations of the presented optimization problem.
Existing Algorithms

**Approaches for uniform hypergraph partitioning:**
- Higher-order SVD (HOSVD) [Govindu '05]
- Hypergraph reduction by clique averaging [Agarwal et al. '05]
- Non-negative tensor factorization [Shashua et al. '06]
- Optimization with $\ell_1$-norm constraints [Liu et al. '10]
- Tensor power iterations [Duchhene et al. '11]
- Game-theoretic approach [Rota Bulo & Pelillo '13]

**Interesting fact:**
All methods other than HOSVD can be formulated as relaxations of the presented optimization problem.

**Question:** Does the optimization provide a good partition?
Consistency of Spectral Partitioning

Key ingredients:

- Algorithm for solving the optimization
- Underlying model
  - A true (unknown) partition of nodes
  - A random hypergraph on the nodes

Question

Is there an algorithm for solving the optimization for which error = o(1) with probability 1 − o(1)?
Consistency of Spectral Partitioning

Key ingredients:
- Algorithm for solving the optimization
- Underlying model
  - A true (unknown) partition of nodes
  - A random hypergraph on the nodes

Question
Is there an algorithm for solving the optimization for which error = \( o(1) \) with probability \( 1 - o(1) \)?

Consistency of spectral graph partitioning:
- Stochastic blockmodel \[ \text{[Holland & Leinhardt '81]} \]
- First consistency result \[ \text{[McSherry '01]} \]
- Consistency of spectral clustering \[ \text{[Rohe, Chatterjee & Yu '11]} \]
- Consistency of max modularity \[ \text{[Zhao, Levina & Zhu '12]} \]
Planted Partition in Graph: [Holland & Leinhardt '81]

- Given \( n \) nodes
- There are \( k \) unknown classes of equal size
- Unknown probabilities \( p, q \in [0, 1] \) with \( p > q \)
- Independent edges with probabilities depending on labels

\[
\text{Prob}(\text{Class-1}, \text{Class-2}) = p, \quad \text{Prob}(\text{Class-1}, \text{Class-3}) = q, \quad \text{Prob}(\text{Class-2}, \text{Class-3}) = q \quad \text{etc.}
\]
Planted Partition in $m$-Uniform Hypergraph:

[Ghoshdastidar & Dukkipati '14]

- Given $n$ nodes
- There are $k$ unknown classes of equal size
- Unknown probabilities $p, q \in [0, 1]$ with $p > q$

- Independent edges with probability $p$ if all nodes are from same class, and $q$ otherwise
The Algorithm

$m$-uniform hypergraph

Adjacency tensor, $A$

Maximize over orthonormal $H \in \mathbb{R}^{n \times k}$
The Algorithm (restated)

$m$-uniform hypergraph

Find $k$ leading eigenvectors
The Algorithm (restated)

$m$-uniform hypergraph

Run \( k \)-means on rows

Find \( k \) leading eigenvectors
Consistency of Algorithm

If $k = O(n^{1/2m})$, then with probability $1 - o(1)$,

$$\text{Number of misclustered nodes} = O\left(\frac{\log n}{(p-q)^2n^{m-3} + \frac{1}{m}}\right).$$

So error $= o(1)$ for $m > 2$, and fractional error $= o(1)$ for $m = 2$. 
### Consistency of Algorithm

If $k = O(n^{1/2m})$, then with probability $1 - o(1)$,

$$
\text{Number of misclustered nodes} = O\left(\frac{\log n}{(p-q)^2 n^{m-3} + \frac{1}{m}}\right).
$$

So error $= o(1)$ for $m > 2$, and fractional error $= o(1)$ for $m = 2$.

### Consistency of HOSVD

[Ghoshdastidar & Dukkipati '14]

If $k = O(n^{1/2m})$, then with probability $1 - o(1)$,

$$
\text{Number of misclustered nodes} = O\left(\frac{(\log n)^2}{(p-q)^4 n^{m-3} + \frac{1}{2m}}\right).
$$
Comparison with HOSVD

Bi-partitioning 3-uniform hypergraph
Ours (solid line) vs. HOSVD (dotted line)

\[(p - q) = 0.1\]  
\[(p - q) = 0.05\]

- Our method makes less error than HOSVD
- Less affected by decrease in probability gap
Proof Outline

Step 1: (Expected case)
- Error = 0 if adjacency tensor is $E[A]$

Step 2: (Matrix perturbation)
- Random case is perturbation of expected case
- Perturbation of eigenvectors = $O(\|A - E[A]\|_{op})$
  
  [Davis & Kahan '70]

A key result

$$\|A - E[A]\|_{op} \leq C\sqrt{n} \text{ with probability } 1 - o(1)$$

Step 3: ($k$-means error)
- Number of nodes misclustered by $k$-means is bounded by eigenvector deviation
  
  [Rohe et al. '11; Lei & Rinaldo '15]
Subspace (Line) Clustering

**Problem:** Cluster points sampled from 3 noisy 5D lines

**Solution:**
- Compute 3-way similarity based on line fitting error

![2D projections](image)

<table>
<thead>
<tr>
<th>Method</th>
<th>$\sigma = 0.02$</th>
<th>$\sigma = 0.05$</th>
</tr>
</thead>
<tbody>
<tr>
<td>NTF</td>
<td>2.50</td>
<td>8.58</td>
</tr>
<tr>
<td>Game Theoretic</td>
<td>8.33</td>
<td>22.17</td>
</tr>
<tr>
<td>Clique Average</td>
<td>3.33</td>
<td>10.92</td>
</tr>
<tr>
<td>Ours</td>
<td>3.25</td>
<td>10.33</td>
</tr>
<tr>
<td>hMETIS</td>
<td>4.50</td>
<td>11.75</td>
</tr>
<tr>
<td>HOSVD</td>
<td>5.17</td>
<td>12.58</td>
</tr>
</tbody>
</table>

**Observation:**
- Error incurred is comparable with the extent optimization is relaxed
- Outperforms HOSVD, hMETIS
Problem: Cluster motions in video

Dataset: Hopkins 155 database (2 motion)

Solution:
- Similar to geometric grouping (of motion trajectories)
- Sampling required to reduce complexity

<table>
<thead>
<tr>
<th>Method</th>
<th>% error</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSA</td>
<td>4.23</td>
</tr>
<tr>
<td>SCC</td>
<td>2.89</td>
</tr>
<tr>
<td>LRR</td>
<td>4.10</td>
</tr>
<tr>
<td>LRR-H</td>
<td>2.13</td>
</tr>
<tr>
<td>LRSC</td>
<td>3.69</td>
</tr>
<tr>
<td>SSC</td>
<td>1.52</td>
</tr>
<tr>
<td>HOSVD</td>
<td>1.83</td>
</tr>
<tr>
<td>HOSVD_{samp}</td>
<td>1.05</td>
</tr>
<tr>
<td>SGC</td>
<td>1.03</td>
</tr>
<tr>
<td>Ours</td>
<td>1.54</td>
</tr>
</tbody>
</table>
Summary

Facts known before:

- Uniform hypergraph partitioning with HOSVD is consistent
- A large number of other algorithms for the same purpose

Our contributions:

- Most of the other methods solve the same optimization
- There is a consistent algorithm to solve the problem
- Our algorithm is provably better than HOSVD

Unanswered questions:

- Why do sampled variants of HOSVD perform better?
- Can we consistently partition non-uniform hypergraphs?

Summary

Facts known before:

- Uniform hypergraph partitioning with HOSVD is consistent
- A large number of other algorithms for the same purpose

Our contributions:

- Most of the other methods solve the same optimization
- There is a consistent algorithm to solve the problem
- Our algorithm is provably better than HOSVD

Summary

Facts known before:
- Uniform hypergraph partitioning with HOSVD is consistent
- A large number of other algorithms for the same purpose

Our contributions:
- Most of the other methods solve the same optimization
- There is a consistent algorithm to solve the problem
- Our algorithm is provably better than HOSVD

Unanswered questions:
- Why do sampled variants of HOSVD perform better?
- Can we consistently partition non-uniform hypergraphs?¹

¹ See [Ghoshdastidar & Dukkipati, arXiv:1505.01582]
Questions?

Acknowledgment:
Debarghya is supported by Google Ph.D. Fellowship