Spectral Clustering using Multilinear SVD Analysis, Approximations and Applications

Debarghya Ghoshdastidar, Ambedkar Dukkipati

Dept. of Computer Science & Automation
Indian Institute of Science
Clustering: The Elegant Way

**Input:** Data points

**Step 1:** Construct graph

**Step 2:** Find the best cut
Clustering: The Elegant And Simple Way

Construct graph

Get the best cut

Compute normalized affinity matrix

Find leading eigenvectors

Run $k$-means on rows
Spectral Clustering

The Good

- Well-defined formulation based on graph partitioning
- Minimize normalized cut / maximize normalized associativity \(\text{[Shi & Malik '00]}\)
- Solved by matrix eigen-decomposition
- Guarantees from perturbation theory \(\text{[Ng, Jordan & Weiss '02]}\)
- Use matrix sampling techniques \(\text{[Fowlkes et al. '04]}\)
Spectral Clustering

The Good

- Well-defined formulation based on graph partitioning
- Minimize normalized cut / maximize normalized associativity [Shi & Malik '00]
- Solved by matrix eigen-decomposition
- Guarantees from perturbation theory [Ng, Jordan & Weiss '02]
- Use matrix sampling techniques [Fowlkes et al. '04]

The Bad

- Cannot use higher-order relations (clusters are circles)
Higher-order Clustering

And The Solution

- Use multi-way relations
  Need $m(\geq 4)$ points to decide a circle or not
- Construct graph $m$-uniform hypergraph
  Each edge connects $m$ nodes
- Relations encoded in matrix $m$-way tensor
Higher-order Clustering

And The Solution

- Use multi-way relations
  - Need \( m(\geq 4) \) points to decide a circle or not
- Construct graph \( m \)-uniform hypergraph
  - Each edge connects \( m \) nodes
- Relations encoded in matrix \( m \)-way tensor

And The Algorithms

- Approximate tensor by matrix \[\text{(Govindu '05)}\]
- Reduce hypergraph to graph \[\text{(Agarwal et al. '05)}\]
- Decompose joint probability tensor \[\text{(Shashua, Zass & Hazan '06)}\]
- Construct evolutionary game \[\text{(Rota Bulo & Pelillo '13)}\]
- Use other optimization criteria \[\text{(Liu et al. '10; Ochs & Brox '11)}\]
  and many more ...
Matrix → Tensor / Graph → Hypergraph

And Finally ... The Ugly

- NO notion of cut / associativity in terms of affinity tensor
- NO motivation for using eigenvectors
- NO idea about what’s the best way to sample

Our contribution: Bridge the gap by defining squared associativity of hypergraph using multilinear singular value decomposition generalizing matrix sampling methods to tensors
And Finally … The Ugly

- NO notion of cut / associativity in terms of affinity tensor
- NO motivation for using eigenvectors
- NO idea about what’s the best way to sample

Our contribution:
Bridge the gap by

- defining squared associativity of hypergraph
- using multilinear singular value decomposition
- generalizing matrix sampling methods to tensors
Squared Associativity

$m$-uniform hypergraph $(\mathcal{V}, \mathcal{E}, w)$

- Set of vertices $\mathcal{V} = \{1, 2, \ldots, n\}$
- Set of edges $\mathcal{E}$: each edge $e = \{i_1, \ldots, i_m\}$ with weight $w(e)$
- $m$-way affinity tensor

$$A_{i_1i_2\ldots i_m} = \begin{cases} 
  w(e) & \text{if } e = \{i_1, i_2, \ldots, i_m\} \in \mathcal{E} \\
  0 & \text{otherwise}
\end{cases}$$
Squared Associativity

$m$-uniform hypergraph $(\mathcal{V}, \mathcal{E}, w)$

- Set of vertices $\mathcal{V} = \{1, 2, \ldots, n\}$
- Set of edges $\mathcal{E}$: each edge $e = \{i_1, \ldots, i_m\}$ with weight $w(e)$
- $m$-way affinity tensor

$$A_{i_1i_2\ldots i_m} = \begin{cases} w(e) & \text{if } e = \{i_1, i_2, \ldots, i_m\} \in \mathcal{E} \\ 0 & \text{otherwise} \end{cases}$$

Squared associativity of the partition

- For $C \subseteq \mathcal{V}$, $\text{Assoc}(C) = \sum_{i_1, \ldots, i_m \in C} A_{i_1i_2\ldots i_m}$
- For $C_1, C_2, \ldots, C_k$ partition of $\mathcal{V}$,

$$\text{SqAssoc}(C_1, C_2, \ldots, C_k) = \sum_{j=1}^{k} \left( \frac{\text{Assoc}(C'_j)}{|C'_j|^m} \right)^2$$
Maximize SqAssoc: A Multilinear SVD Problem

Our objective:
Find $k$ non-overlapping cluster assignment vectors that maximize SqAssoc

Result
Relaxation of above objective equivalent to:
Find $k$ leading left singular vectors $\hat{A}$

Result based on multilinear SVD of tensors
[De Lathauwer, De Moore & Vandewalle '00; Chen & Saad '09]
Similar approach also used in [Govindu '05]
Higher-order Clustering: The Elegant And Simple Way

$m$-uniform hypergraph

Compute $m$-way affinity tensor

Flatten the tensor

Find leading left singular vectors

Run $k$-means on rows
**Perturbation Result**

**The ideal case:**
- $C_1, \ldots, C_k$ are known a priori.
- Affinity tensor
  \[
  A_{i_1i_2\ldots i_m} = \begin{cases} 
  1 & \text{if } i_1, i_2, \ldots, i_m \in C_j \text{ for some } j, \\
  0 & \text{otherwise.}
  \end{cases}
  \]

**Result**

If
- $n >$ some threshold,
- each cluster is not too small, and
- $\|\hat{A} - \tilde{A}\|_2 = O(k^{-m}n^{m-\alpha})$ for some $\alpha > 0$,

then number of misclustered nodes is $O(kn^{-2\alpha})$.

- Above bound improves upon [Chen & Lerman '09]
Higher-order Clustering: Computation too high

$m$-uniform hypergraph

Compute $m$-way affinity tensor

Flatten the tensor

Find leading left singular vectors

Run $k$-means on rows
Higher-order Clustering: Approximations Needed

$m$-uniform hypergraph

Flatten the tensor

Compute $m$-way affinity tensor

Find leading left singular vectors

Run $k$-means on rows

Need to Sample
Random sampling used for tensor completion [Jain & Oh '14]
Poor performance when used in clustering
Random sampling used for tensor completion [Jain & Oh '14]
Poor performance when used in clustering

We focus on:
- Column sampling [Drineas, Kannan & Mahoney '06]
- Nyström approximation [Fowlkes et al. '04]

We generalize these samplings to tensors
Column Sampling

Sample columns → SVD

Let column \( a_i \) be sampled with probability \( p_i \). Uniform sampling is not optimal. Optimal sampling
\[ p_i \propto \|a_i\|_2^2 \]
(computation costly)
Column Sampling

Can also represent as:

Sample columns

SVD

Uniform sampling is not optimal

Optimal sampling \( p_i \propto \|a_i\|^2 \) (computation costly)
Column Sampling

Can also represent as:

- Well-studied approach [Drineas, Kannan & Mahoney '06]
- Let column $a_i$ be sampled with probability $p_i$
- Uniform sampling is not optimal
- Optimal sampling $p_i \propto \|a_i\|_2^2$ (computation costly)
Improved sampling:

- Observe: Sample 1 column \( \equiv \) fix \((m - 1)\) data points
- Run initial clustering \((k\text{-means} / k\text{-subspace})\)
- Each time choose \((m - 1)\) points from a cluster
- (optional) Reject column \(a\) if \(\|a\|_2\) too small
Nyström Approximation

Matrix case:

- Compute eigenvectors of sub-matrix and extend
- Extension minimizes reconstruction error of some entries
Nyström Approximation

Matrix case:
- Compute eigenvectors of sub-matrix and extend
- Extension minimizes reconstruction error of some entries

Generalization to tensors:

Sample → SVD → Choose initial sub-tensor wisely (run initial clustering)
Nyström Approximation

**Matrix case:**
- Compute eigenvectors of sub-matrix and extend
- Extension minimizes reconstruction error of some entries

**Generalization to tensors:**

![Diagram showing the process of Nyström Approximation for tensors](image)

- Sample
- Minimize reconstruction error
- SVD
Nyström Approximation

Matrix case:
- Compute eigenvectors of sub-matrix and extend
- Extension minimizes reconstruction error of some entries

Generalization to tensors:
- Choose initial sub-tensor wisely (run initial clustering)
Numerical Results

Line clustering

- Column sampling (black)
- Nyström method (red)
- Comparable time taken

Motion segmentation

<table>
<thead>
<tr>
<th>Method</th>
<th>2-motion</th>
<th>3-motion</th>
<th>All</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSA</td>
<td>4.23</td>
<td>7.02</td>
<td>4.86</td>
</tr>
<tr>
<td>SCC</td>
<td>2.89</td>
<td>8.25</td>
<td>4.10</td>
</tr>
<tr>
<td>LRR</td>
<td>4.10</td>
<td>9.89</td>
<td>5.41</td>
</tr>
<tr>
<td>LRR-H</td>
<td>2.13</td>
<td>4.03</td>
<td>2.56</td>
</tr>
<tr>
<td>LRSC</td>
<td>3.69</td>
<td>7.69</td>
<td>4.59</td>
</tr>
<tr>
<td>SSC</td>
<td>1.52</td>
<td>4.40</td>
<td>2.18</td>
</tr>
<tr>
<td>SGC</td>
<td>1.03</td>
<td>5.53</td>
<td>2.05</td>
</tr>
</tbody>
</table>

Multilinear SVD with column sampling

<table>
<thead>
<tr>
<th>Uniform</th>
<th>1.83</th>
<th>9.31</th>
<th>3.52</th>
</tr>
</thead>
<tbody>
<tr>
<td>with initial (k)-means</td>
<td>1.05</td>
<td>5.72</td>
<td>2.11</td>
</tr>
</tbody>
</table>

- Column sampling better than Nyström approximation
- Significant improvement if we use initial clustering
This work was supported by

Google

under the

Google Ph.D. Fellowship Program