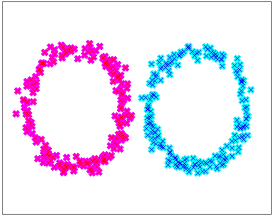
Spectral Clustering using Multilinear SVD

Analysis, Approximations and Applications

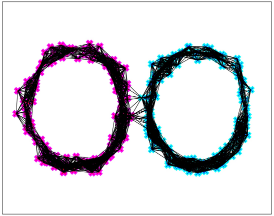
Debarghya Ghoshdastidar, Ambedkar Dukkipati

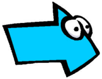
Dept. of Computer Science & Automation
Indian Institute of Science

Input: Data points

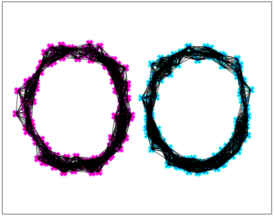


Step 1: Construct graph

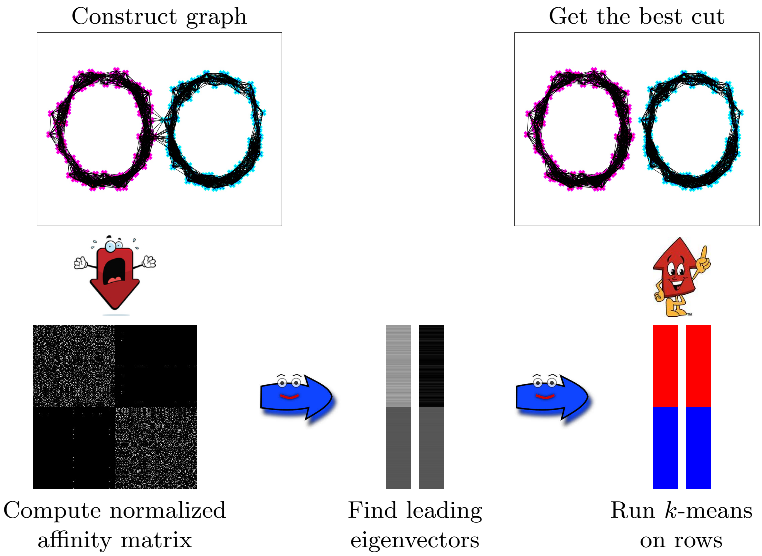




Step 2: Find the best cut



Clustering: The Elegant And Simple Way



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]

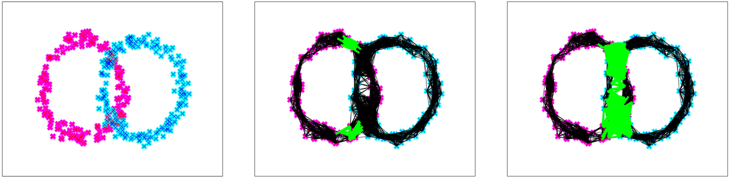
Spectral Clustering

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

Best cut Correct cut



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

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 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 [Govindu '05]
- Reduce hypergraph to graph [Agarwal et al. '05]
- Decompose joint probability tensor [Shashua, Zass & Hazan '06]
- Construct evolutionary game [Rota Bulo & Pelillo '13]
- Use other optimization criteria [Liu et al. '10; Ochs & Brox '11]

and many more ...

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

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

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

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

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

Squared associativity of the partition

- For $C \subseteq \mathcal{V}$, $\quad \mathrm{Assoc}(C) = \sum_{i_1, \ldots, i_m \in C} \mathbf{A}_{i_1 i_2 \ldots i_m}$
- For $C_1, C_2, \ldots, C_k$ partition of $\mathcal{V}$,

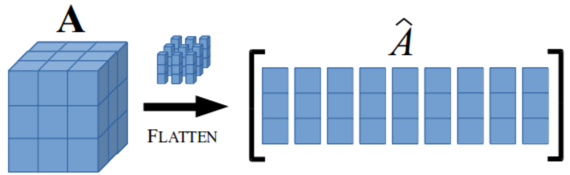
$$\mathrm{SqAssoc}(C_1, C_2, \ldots, C_k) = \sum_{j=1}^{k} \frac{\left(\mathrm{Assoc}(C_j)\right)^2}{|C_j|^m}$$

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 $\widehat{A}$

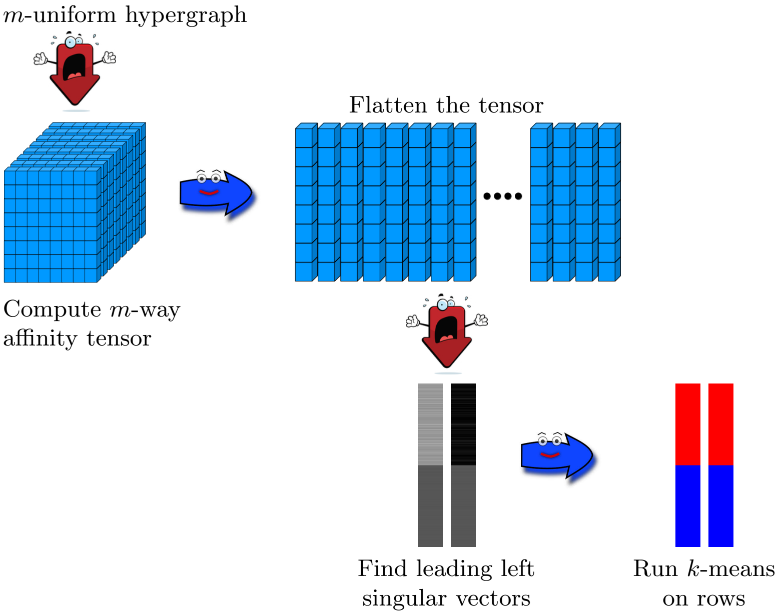


Result based on multilinear SVD of tensors

[De Lathauwer, De Moore & Vandewalle '00; Chen & Saad '09]

Similar approach also used in [Govindu '05]

Higher-order Clustering: The Elegant And Simple Way



The ideal case:

- $C_1, \ldots, C_k$ are known a priori.
- Affinity tensor

$$\mathcal{A}_{i_1 i_2 \ldots i_m} = \begin{cases} 1 & \text{if } i_1, i_2, \ldots, i_m \in C_j \text{ for some } j, \\ 0 & \text{otherwise.} \end{cases}$$

Result

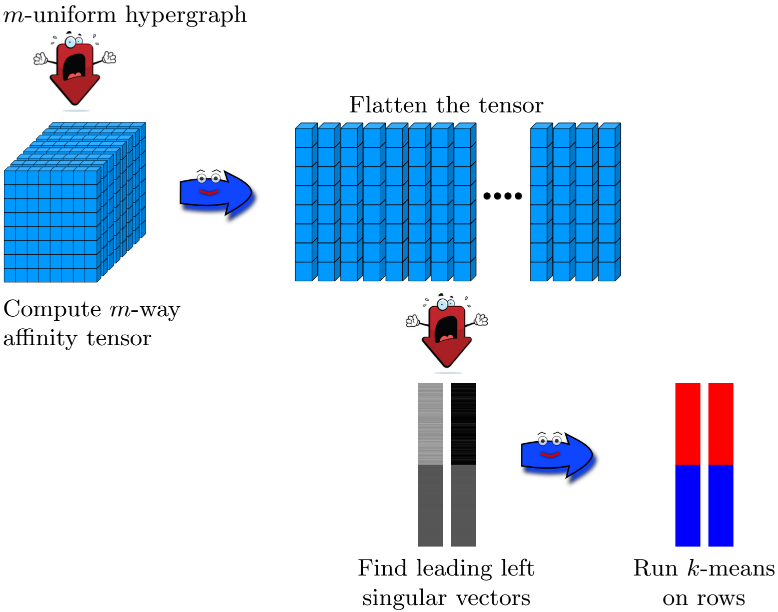
If

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

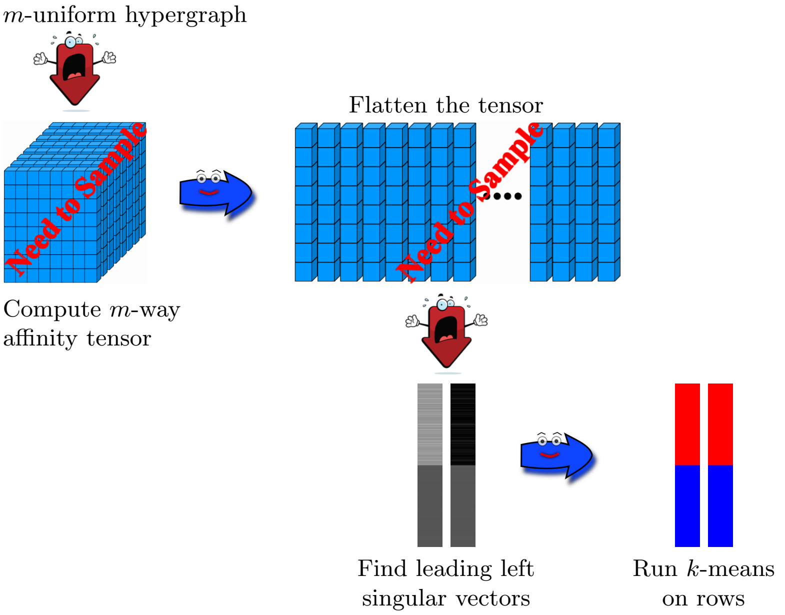
then number of misclustered nodes is $O(kn^{-2\alpha})$.

- Above bound improves upon [Chen & Lerman '09]

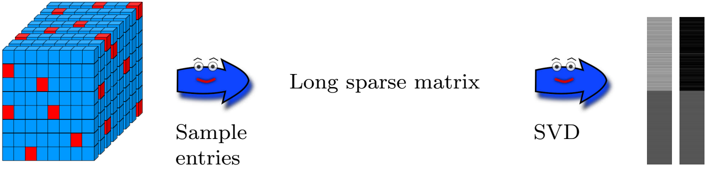
Higher-order Clustering: Computation too high



Higher-order Clustering: Approximations Needed

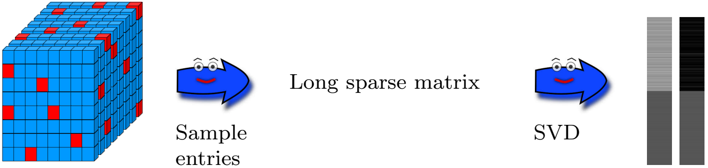


Random Sampling



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

Random Sampling



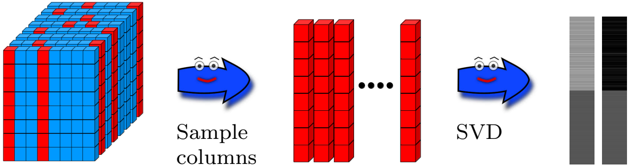
- Random sampling used for tensor completion [Jain & Oh '14]
- Poor performance when used in clustering

We focus on:

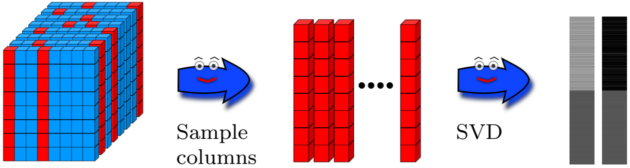
- Column sampling [Drineas, Kannan & Mahoney '06]
- Nyström approximation [Fowlkes et al. '04]

We generalize these samplings to tensors

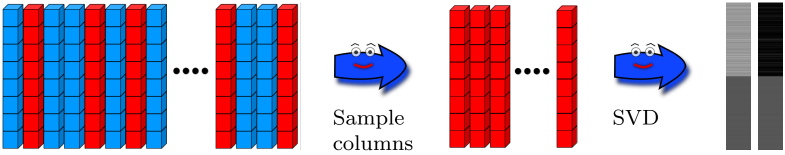
Column Sampling



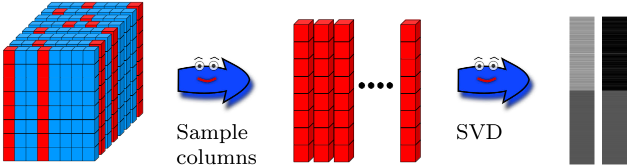
Column Sampling



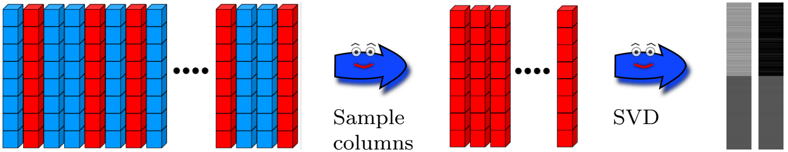
Can also represent as:



Column Sampling

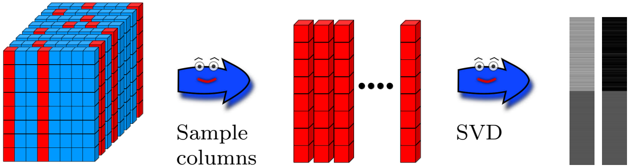


Can also represent as:



- 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)

Column Sampling



Improved sampling:

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

Nyström Approximation

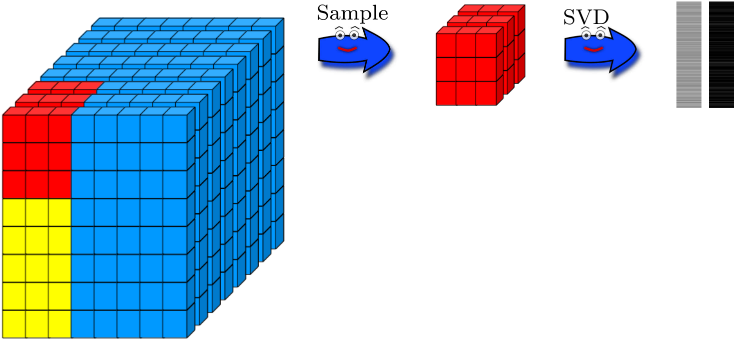
Matrix case:

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

Matrix case:

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

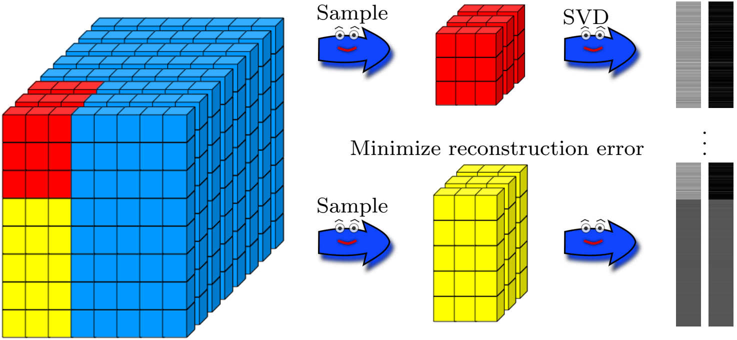
Generalization to tensors:



Matrix case:

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

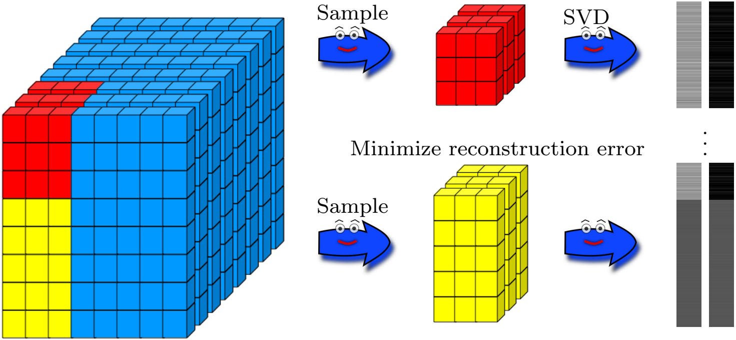
Generalization to tensors:



Matrix case:

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

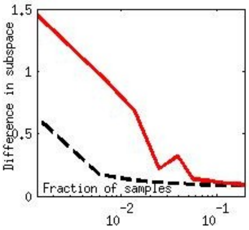
Generalization to tensors:



- Choose initial sub-tensor wisely (run initial clustering)

Numerical Results

Line clustering



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

Motion segmentation

Mean % error on Hopkins 155 dataset

Method	2-motion	3-motion	All
LSA	4.23	7.02	4.86
SCC	2.89	8.25	4.10
LRR	4.10	9.89	5.41
LRR-H	2.13	4.03	2.56
LRSC	3.69	7.69	4.59
SSC	1.52	4.40	2.18
SGC	1.03	5.53	2.05

Multilinear SVD with column sampling

Uniform	1.83	9.31	3.52
with initial k-means	1.05	5.72	2.11

- 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