Hypergraph clustering with non-uniform partitions of hyperedges

Pan Li, Weihao Gao and Tao Mo

CS 512
University of Illinois at Urbana-Champaign

May 8, 2017
Clustering over Graphs

- **Classical Graph Clustering Problem:** Given pairwise connections between individuals, identify “clusters” using edge density information.

- **Social networks:** Given social connections between pairs of individuals, indentify “communities”.

Zachary’s karate club graph (Zachary 1977)
Traditional clustering methods: simplest structures, e.g., edges connecting pairs of individuals.

Social Networks have large number of network motifs (such as triangles). Community detection has to be performed through careful placement of network motifs.

A real-world network consists of motifs.
Traditional clustering methods: simplest structures, e.g., edges connecting pairs of individuals.

Higher-order relations (termed as hyperedges): involving multiple individuals
Hypergraph clustering: higher-order relations

- Traditional clustering methods: simplest structures, e.g., edges connecting pairs of individuals.

- Higher-order relations (termed as hyperedges): involving multiple individuals

- Hypergraph clustering: Incorporating hyperedges into the clustering procedure
Hypergraph clustering: higher-order relations

- Traditional clustering methods: simplest structures, e.g., edges connecting pairs of individuals.

- Higher-order relations (termed as hyperedges): involving multiple individuals

- Hypergraph clustering: Incorporating hyperedges into the clustering procedure

- Classical methods for hypergraph clustering can only handle uniform partition of hyperedges.
Hypergraph clustering: higher-order relations

- Traditional clustering methods: simplest structures, e.g., edges connecting pairs of individuals.

- Higher-order relations (termed as hyperedges): involving multiple individuals

- Hypergraph clustering: Incorporating hyperedges into the clustering procedure

- Classical methods for hypergraph clustering can only handle uniform partition of hyperedges.

- A new algorithm can handle non-uniform partition of hyperedges.
What is hypergraph clustering with uniform partitions of hyperedges? non-uniform partitions?
Hypergraph clustering with **uniform partitions** of hyperedges

- A hypergraph: $\mathcal{H} = (V, E)$.
- Hypergraph clustering (2 clusters): $V = S \cup \bar{S}$.

![Hypergraph diagram]

1. $\bar{S}$ includes vertices 2, 4, 5, 6, 7, and 9.
2. $S$ includes vertices 1, 3, and 8.
3. The hyperedge $e$ is partitioned/cut by $(S, \bar{S})$ if $e \in \partial S$.
4. Uniform partition of hyperedges: any types of partitioning $e$ pay a scalar cost $w_e$.
5. Total cut (uniform cases): $\text{Cut}_u(S) = \sum_{e \in \partial S} w_e$.
6. Hypergraph clustering with uniform partitions of hyperedges (normalized version): $\arg\min_S \text{NCut}_u(S) = \text{Cut}_u(S) \times \text{Normalization-term}(S)$.
Hypergraph clustering with uniform partitions of hyperedges

- A hypergraph: \( H = (V, E) \).
- Hypergraph clustering (2 clusters): \( V = S \cup \bar{S} \).
- Boundary of set: \( \partial S \triangleq \{ e \in E | e \cap S \neq \emptyset, e \cap \bar{S} \neq \emptyset \} \).
  Hyperedge \( e \) is partitioned/cut by \((S, \bar{S})\) if \( e \in \partial S \)
Hypergraph clustering with **uniform partitions** of hyperedges

- A hypergraph: \( \mathcal{H} = (V, E) \).
- Hypergraph clustering (2 clusters): \( V = S \cup \bar{S} \).
- Boundary of set: \( \partial S \triangleq \{ e \in E \mid e \cap S \neq \emptyset, e \cap \bar{S} \neq \emptyset \} \).
  Hyperedge \( e \) is **partitioned/cut** by \( (S, \bar{S}) \) if \( e \in \partial S \).
- **Uniform partition** of hyperedges: any types of partitioning \( e \) pay a scalar cost \( w_e^u \).

![Diagram of hypergraph clustering with uniform partitions of hyperedges]

**Equation:**

\[
\text{Total cut (uniform cases): } \text{Cut}_{u}(S) = \sum_{e \in \partial S} w_e^u
\]

**Equation (1):**

\[
\arg \min_{S} \text{NCut}_{u}(S) = \text{Cut}_{u}(S) \times \text{Normalization-term}(S) \]
A hypergraph: $\mathcal{H} = (V, E)$.

Hypergraph clustering (2 clusters): $V = S \cup \bar{S}$.

Boundary of set: $\partial S \triangleq \{ e \in E | e \cap S \neq \emptyset, e \cap \bar{S} \neq \emptyset \}$. Hyperedge $e$ is partitioned/cut by $(S, \bar{S})$ if $e \in \partial S$.

Uniform partition of hyperedges: any types of partitioning $e$ pay a scalar cost $w^u_e$.

Total cut (uniform cases):

$$\text{Cut}^u(S) = \sum_{e \in \partial S} w^u_e$$
A hypergraph: $\mathcal{H} = (V, E)$.

Hypergraph clustering (2 clusters): $V = S \cup \bar{S}$.

Boundary of set: $\partial S \triangleq \{ e \in E | e \cap S \neq \emptyset, e \cap \bar{S} \neq \emptyset \}$.

Hyperedge $e$ is partitioned/cut by $(S, \bar{S})$ if $e \in \partial S$.

Uniform partition of hyperedges: any types of partitioning $e$ pay a scalar cost $w^u_e$.

Total cut (uniform cases):

$$\text{Cut}_u(S) = \sum_{e \in \partial S} w^u_e$$

Hypergraph clustering with uniform partitions of hyperedges (normalized version):

$$\arg \min_S \text{NCut}_u(S) = \text{Cut}_u(S) \times \text{Normalization-term}(S). \quad (1)$$
Non-uniform case?
Hypergraph clustering with non-uniform partitions of hyperedges

- A hypergraph: $\mathcal{H} = (V, E)$.
- Hypergraph clustering (2 clusters): $V = S \cup \bar{S}$.
- Boundary of set: $\partial S \triangleq \{ e \in E | e \cap S \neq \emptyset, e \cap \bar{S} \neq \emptyset \}$.
  Hyperedge $e$ is partitioned/cut by $(S, \bar{S})$ if $e \in \partial S$.

![Diagram of hypergraph clustering with non-uniform partitions]

1. 2 3
2. 4 5
3. 6 7
4. 8
5. 9
6. S
7. \bar{S}
8. S \ \bar{S}
9. S \ \bar{S}
Hypergraph clustering with **non-uniform partitions** of hyperedges

- A hypergraph: $\mathcal{H} = (V, E)$.
- Hypergraph clustering (2 clusters): $V = S \cup \overline{S}$.
- Boundary of set: $\partial S \triangleq \{ e \in E | e \cap S \neq \emptyset, e \cap \overline{S} \neq \emptyset \}$.
  Hyperedge $e$ is **partitioned/cut** by $(S, \overline{S})$ if $e \in \partial S$.
- **Non-uniform partitions** of hyperedges: different types of partitioning $e$ pay **different costs** $w_{e}^{nu}(\cdot)$. Cost function $w_{e}^{nu}(\cdot)$: if $e$ is partitioned into $e = e_1 \cup e_2$, pay $w_{e}^{nu}(e_1)$ ($= w_{e}^{nu}(e_2)$).

![Diagram of hypergraph clustering with non-uniform partitions](image)
Hypergraph clustering with **non-uniform partitions** of hyperedges

- A hypergraph: $\mathcal{H} = (V, E)$.
- Hypergraph clustering (2 clusters): $V = S \cup \bar{S}$.
- Boundary of set: $\partial S \triangleq \{ e \in E | e \cap S \neq \emptyset, e \cap \bar{S} \neq \emptyset \}$. Hyperedge $e$ is **partitioned/cut** by $(S, \bar{S})$ if $e \in \partial S$.
- **Non-uniform partitions** of hyperedges: different types of partitioning $e$ pay **different costs** $w_{e}^{nu}(\cdot)$. Cost function $w_{e}^{nu}(\cdot)$: if $e$ is partitioned into $e = e_1 \cup e_2$, pay $w_{e}^{nu}(e_1) (= w_{e}^{nu}(e_2))$.
- Total cut (non-uniform cases):

$$\text{Cut}_{nu}(S) = \sum_{e \in \partial S} w_{e}^{nu}(S \cap e)$$
Hypergraph clustering with **non-uniform partitions** of hyperedges

- A hypergraph: $\mathcal{H} = (V, E)$.
- Hypergraph clustering (2 clusters): $V = S \cup \bar{S}$.
- Boundary of set: $\partial S \triangleq \{ e \in E | e \cap S \neq \emptyset, e \cap \bar{S} \neq \emptyset \}$.
  Hyperedge $e$ is **partitioned/cut** by $(S, \bar{S})$ if $e \in \partial S$.
- **Non-uniform partitions** of hyperedges: different types of partitioning $e$ pay **different costs** $w_{e}^{nu}(\cdot)$. Cost function $w_{e}^{nu}(\cdot)$: if $e$ is partitioned into $e = e_1 \cup e_2$, pay $w_{e}^{nu}(e_1)$ ($= w_{e}^{nu}(e_2)$).
- Total cut (non-uniform cases):
  \[
  \text{Cut}_{nu}(S) = \sum_{e \in \partial S} w_{e}^{nu}(S \cap e)
  \]
- Hypergraph clustering with **non-uniform partitions** of hyperedges (normalized version):
  \[
  \arg \min_{S} \text{NCut}_{nu}(S) = \text{Cut}_{nu}(S) \times \text{Normalization-term}(S).
  \]
Why consider non-uniform partitions of hyperedges?

- Application 1: Network motif clustering
- Application 2: Subspace clustering
- Application 3: Ranking data analysis
Application 1: Network motif clustering

- Analyze Florida Bay ecosystem food web.
Application 1: Network motif clustering

- Analyze Florida Bay ecosystem food web.
- Vertices: 128 species.
Application 1: Network motif clustering

- Analyze Florida Bay ecosystem food web.
- Vertices: 128 species.
- Directed edges: Carbon flow. $A \rightarrow B$: Carbon flows from $A$ to $B$. $A$ is a preyer of $B$ while $B$ is a predator of $A$. 
A. Benson, D. Gleich and J. Leskovec (Science'16) do hypergraph clustering via clustering network motifs (hyperedge with uniform cost).

\[ w_e^u = 1 \]
A. Benson, D. Gleich and J. Leskovec (Science’16) do hypergraph clustering via clustering network motifs (hyperedge with uniform cost).

\[ w_e^u = 1 \]

Observe three clusters.
A. Benson, D. Gleich and J. Leskovec (Science’16) do hypergraph clustering via clustering network motifs (hyperedge with uniform cost).

\[ w_e^u = 1 \]

Observe three clusters.

- Cluster 2: 9 microfauna + 1 macroinvertebrate + 2 detrius. Purity: 75%. Missing: 3.
Application 1: Network motif clustering

- We do hypergraph clustering via clustering network motifs (non-uniform case).

\[ w_e(\{v_i\}) = 1, \quad i = 1, 2, 3, 4. \]

\[ w_e(\{v_1, v_2\}) = 0. \]

\( v_1 \) and \( v_2 \) both are preyers of \( v_3 \) and \( v_4 \):
Application 1: Network motif clustering

- We do hypergraph clustering via clustering network motifs (non-uniform case).


\[ w_e(\{v_i\}) = 1, \quad i = 1, 2, 3, 4. \]

\[ w_e(\{v_1, v_2\}) = 0. \]

- \( v_1 \) and \( v_2 \) both are preyers of \( v_3 \) and \( v_4 \):

- Five clusters:

```
Producer: 10/10/7
Macroinvertebrates: 30/36/2
Pelagic fishes: 18/28/4
Benthic fishes: 16/33/7
Terrestrial species: 12/15/8
```

- Non-uniform
  - 7 singletons
  - Covering all classes
  - Higher purity

- Uniform
  - 67 singletons
  - Covering 3/5 classes
  - Lower purity
Input: A hypergraph with non-uniform hyperedges $\mathcal{H} = (V, E, w^{un})$.

- Project each non-uniform hyperedge onto an ordinary subgraph;
**Algorithm sketch**

Input: A hypergraph with non-uniform hyperedges $\mathcal{H} = (V, E, w^{un})$.

- Project each non-uniform hyperedge onto an ordinary subgraph;
  - For each hyperedge $(e, w_e^{un})$, construct a complete ordinary subgraph $G_e = (V(e), E(e), w(e))$ to approximate $e$.

$$\min_{w(e), \beta(e)} \beta(e) \quad \text{s.t.} \quad w_e^{un}(S) \leq \sum_{v \in S, \tilde{v} \in e / S} w_v^{(e)} \leq \beta(e) w_e^{un}(S),$$

for all $S \subseteq e$ that $w_e^{un}(S)$ is defined.
Algorithm sketch

Input: A hypergraph with non-uniform hyperedges $\mathcal{H} = (V, E, w^{un})$.

- Project each non-uniform hyperedge onto an ordinary subgraph;
  - For each hyperedge $(e, w_e^{un})$, construct a complete ordinary subgraph $G_e = (V^{(e)}, E^{(e)}, w^{(e)})$ to approximate $e$.

$$\min_{w^{(e)}, \beta^{(e)}} \beta^{(e)} \quad \text{s.t.} \quad w^{un}_e(S) \leq \sum_{v \in S, \bar{v} \in e \setminus S} w^{(e)}_{v\bar{v}} \leq \beta^{(e)} w^{un}_e(S),$$

  for all $S \subseteq e$ that $w^{un}_e(S)$ is defined.

- Construct an ordinary graph $G = (V, E_o, w)$ by combining subgraphs $G_e$’s,

  $$w_{v\bar{v}} \triangleq \sum_{e \in E} w^{(e)}_{v\bar{v}}, \quad \text{for all pairs } \{v, \bar{v}\} \in E_o.$$
Algorithm sketch

Input: A hypergraph with non-uniform hyperedges \( \mathcal{H} = (V, E, \mathbf{w}^{un}) \).

- Project each non-uniform hyperedge onto an ordinary subgraph;
  - For each hyperedge \((e, \mathbf{w}_e^{un})\), construct a complete ordinary subgraph \( G_e = (V^{(e)}, E^{(e)}, w^{(e)}) \) to approximate \( e \).

\[
\min_{\mathbf{w}^{(e)}, \beta^{(e)}} \beta^{(e)} \text{ s.t. } w^{un}(S) \leq \sum_{v \in S, \tilde{v} \in e/S} w^{(e)}_{v \tilde{v}} \leq \beta^{(e)} w^{un}(S),
\]

for all \( S \subseteq e \) that \( w^{un}(S) \) is defined.

- Construct an ordinary graph \( G = (V, E_o, \mathbf{w}) \) by combining subgraphs \( G_e \)'s,

\[
w_{v \tilde{v}} \triangleq \sum_{e \in E} w^{(e)}_{v \tilde{v}}, \text{ for all pairs } \{v, \tilde{v}\} \in E_o.
\]

- Apply spectral clustering over the obtained graph \( G \).
If $w_e^{un}$ satisfies submodularity, i.e., for all $S_1, S_2 \subseteq e$,

$$w_e^{un}(S_1) + w_e^{un}(S_2) \geq w_e^{un}(S_1 \cap S_2) + w_e^{un}(S_1 \cup S_2),$$

the 2-way partitioning solution given by the proposed algorithm is quadratically approximate to the optimality. Mathematically, let $\hat{S}$ be the partitioning solution and define $OPT \triangleq \arg\min_S NCut_{nu}(S)$, for some constant $c \geq 1$,

$$OPT \leq NCut_{nu}(\hat{S}) \leq c\sqrt{2}OPT.$$ 

Note that $OPT \leq 2$. 

Thanks! Questions?