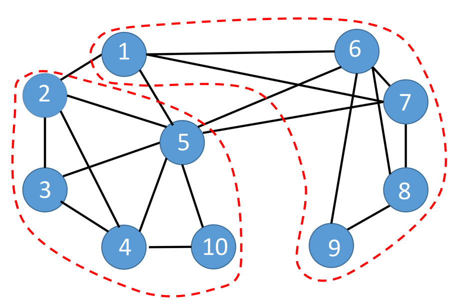
Inhomogeneous Hypergraph Clustering with Applications

Pan Li [1] Olgica Milenkovic [1]

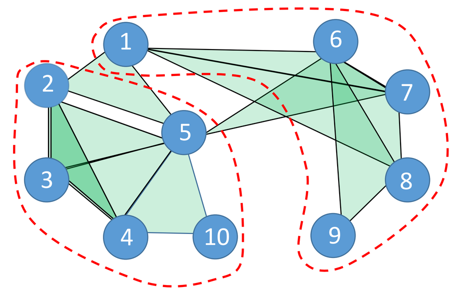
[1]University of Illinois at Urbana-Champaign

Introduction

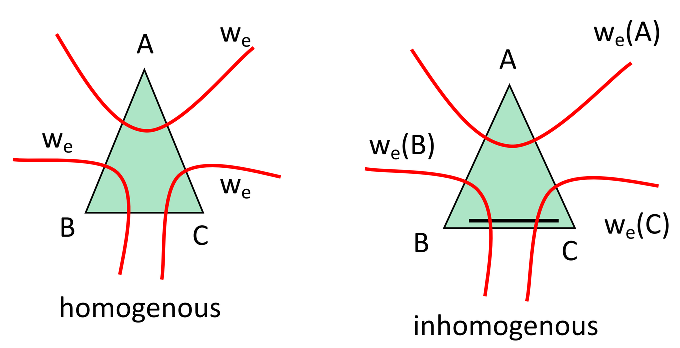
- Graph clustering: Group the vertices into clusters such that there should be many edges within each cluster and relatively few between the clusters.

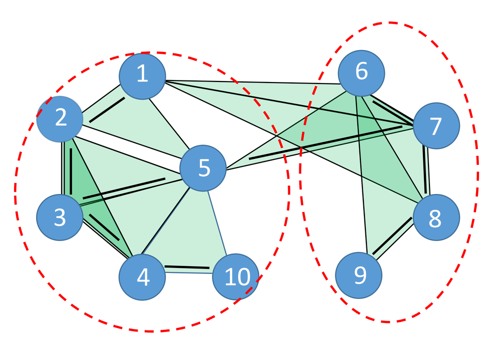


- Hypergraph clustering: Relationships among the vertices are more complex than pairwise, modeled as hyperedges.



- Inhomogeneous Hypergraph (V, E, w) Clustering: Different subsets of vertices within the same hyperedge may have different structural importance. The weight of hyperedges can be a set function: for $e \in E$, $w_e(\cdot) : 2^e \to \mathbb{R}_{\geq 0}$, $w_e(S) = w_e(\bar{S})$, $w_e(\emptyset) = 0$.





Balanced partition and normalized cut:

- Partition of V: $V = S \cup \bar{S}$.
- Cost of cut (inhomogeneous case): for $S \subseteq V$,

$$\mathrm{Cut}(S) = \sum_{e \in \partial S} w_e(S \cap e).$$

- Volume of set S:

$$\mathrm{Vol}(S) = \sum_{v \in S} d_v = \sum_{v \in S} \sum_{e: v \in e} w_e(\{v\}).$$

- Inhomogenous Hypergraph Partitioning: Minimize normalized hypergraph cut

$$\arg\min_S \mathrm{NCut}(S) = \mathrm{Cut}(S) \times \left(\frac{1}{\mathrm{Vol}(S)} + \frac{1}{\mathrm{Vol}(\bar{S})} \right).$$

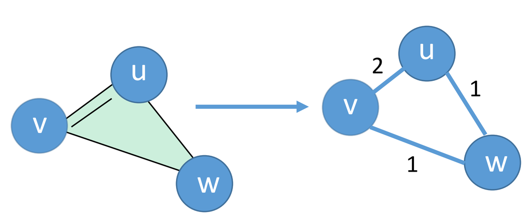
- Generalization of graph/hypergraph normalized cut.

Algorithm

- Projection: Use a complete graph to "approximate" each hyperedge. $\{w_{v\tilde{v}}^{(e)}\}_{v, \tilde{v} \in e}$ stands for projected edge weights from hyperedge e.

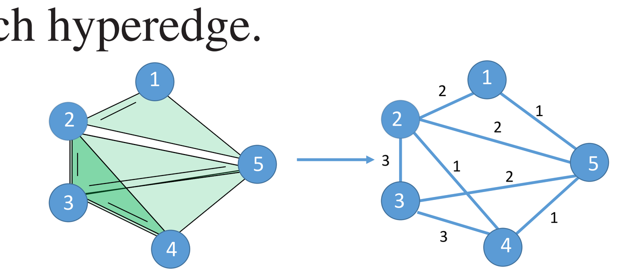
$$w_e(\cdot) \to \{w_{uv}^{(e)}\}_{u,v \in e},$$

e.g. $e = \{u, v, s\}$, $w_e(u) = w_e(v) = 3$, $w_e(s) = 2$.



- Merging: Merge subgraphs obtained from each hyperedge.

$$w_{vu} = \sum_{e \in E} w_{vu}^{(e)}$$



- Spectral partitioning: Run standard spectral clustering algorithm.

Projection and Infeasibility

- The projection step is critical.

$$\textbf{P1:} \quad \min_{w^{(e)}} \beta^{(e)} \quad \text{s.t. } w_e(S) \leq \sum_{v \in S, u \in e/S} w_{vu}^{(e)} \leq \beta^{(e)} \, w_e(S),$$

$$\text{for all } S \in 2^e \text{ for which } w_e(S) \text{ is defined,}$$

- Quadratic optimality: If there exist feasible constants $\beta^{(e)}$ for all hyperedges e and $w_{vu} = \sum_{e \in E} w_{vu}^{(e)} \geq 0$ for all $\{v, u\}$, then $\alpha =$NCut of spectral clustering satisfies

$$(\beta^*)^3 \alpha^* \geq \frac{\alpha^2}{8} \geq \frac{(\alpha^*)^2}{8},$$

where α^* is the optimal normalized cut and $\beta^* = \max_{e \in E} \beta^{(e)}$.

- $w_{v\tilde{v}}^{(e)}$ may be negative or even the optimization problem is infeasible, e.g. $e = \{1, 2, 3, 4\}$, with $w_e(\{1, 4\}) = w_e(\{2, 3\}) = 1$ and $w_e(S) = 0$ for all other choices of sets S.

Performance guarantee with submodular weights

- Submodular weights:

$$w_e(S_1) + w_e(S_2) \geq w_e(S_1 \cap S_2) + w_e(S_1 \cup S_2) \quad \text{for all } S_1, S_2 \in 2^e.$$

- If $w_e(\cdot)$ is submodular, then there are nonnegative $\{w_{vu}^{(e)}\}_{v,u \in e}$ and $\beta^{(e)}$ feasible to **P1**. For general $|e|$, $\beta^{(e)} \leq |e| - 1$ [1].
- Optimal linear projection (OLP):

$$w_{vu}^{*(e)} = \sum_{S \in 2^e / \{\emptyset, e\}, v \in S} \left[\frac{w_e(S)}{|S|(\delta(e) - |S|)} 1_{u \notin S} - \frac{w_e(S)}{(|S| - 1)(\delta(e) - |S| + 1)} 1_{u \in S} \right].$$

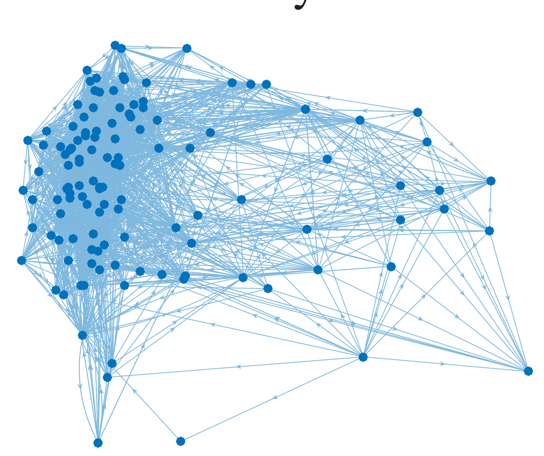
- OLP is proper for practical use.
  a) Best known approximation ratio $\beta^{(e)}$ when $|e| \leq 7$.

$\lvert e \rvert$	2	3	4	5	6	7
β	1	1	1.5	2	4	6

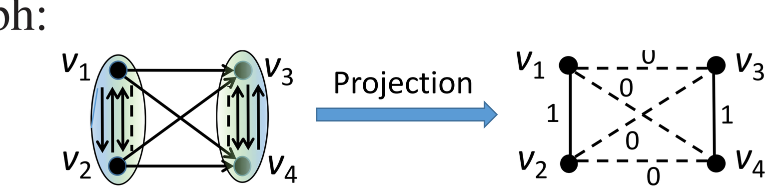
  b) Consistency: if $w_e(\cdot)$ is from graph cuts, then OLP recovers the underlying graph.
  c) Efficiently computable.

Application I: Foodweb hierarchical clustering

- Florida Bay food web

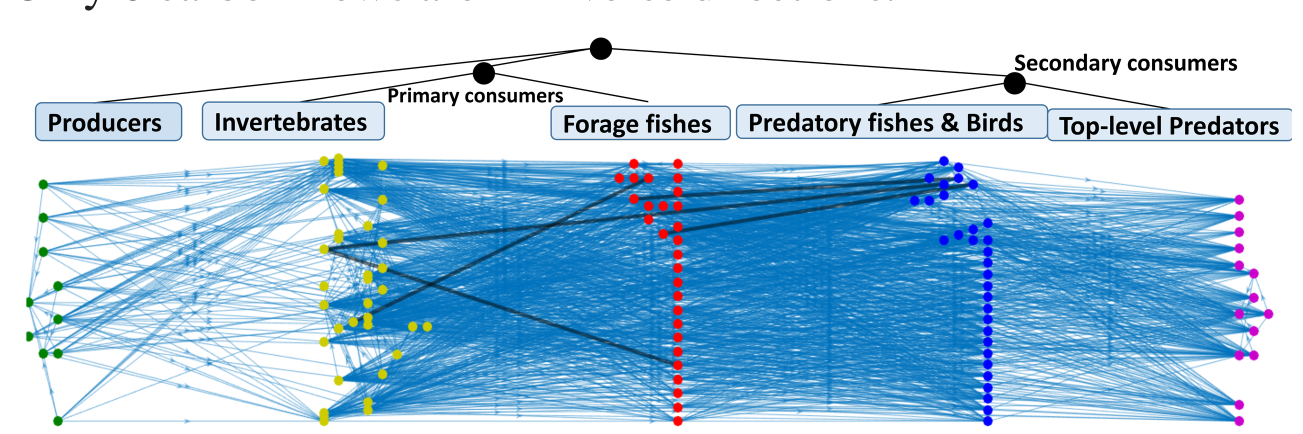


- The network motif to construct the inhomogeneous hypergraph:



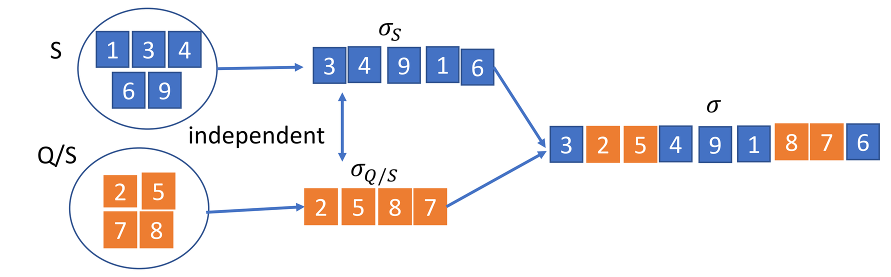
$w_e(\{v_i\}) = 1$ for $i = 1, 2, 3, 4$
$w_e(\{v_1, v_3\}) = 2$, and $w_e(\{v_1, v_4\}) = 2$
$w_e(\{v_1, v_2\}) = 0$

- 5 clusters; Only 6 carbon flows are in inverse directions.



Application II: Category Learning in Rankings

- Riffled independence: Rankings of candidates within different categories are independent from each other [2].
- Goal: Detecting the riffled independent category S.
  Q: Set of candidates to be ranked.
  S: $\subseteq Q$, a specific category of candidates.
  $\sigma : Q \to [n]$, a full ranking over Q.



- Define

$$I_{i;j,k} \triangleq I(\sigma(i); 1_{\sigma(j) < \sigma(k)}) = \sum_{\sigma(i)} \sum_{1_{\sigma(j) < \sigma(k)}} \mathbb{P}(\sigma(i), 1_{\sigma(j) < \sigma(k)}) \log \frac{\mathbb{P}(\sigma(i), 1_{\sigma(j) < \sigma(k)})}{\mathbb{P}(\sigma(i)) \mathbb{P}(1_{\sigma(j) < \sigma(k)})},$$

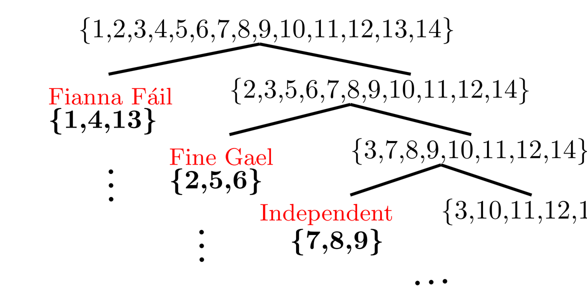
$I_{i;j,k} = 0$ indicates the position of i is independent from the rankings of j and k.

- The underlying optimization problem:

$$\arg\min_{S \subseteq Q} \mathcal{F}(S) \triangleq \sum_{(i,j,k) \in \Omega^{cross}_{S,\bar{S}}} I_{i;j,k} + \sum_{(i,j,k) \in \Omega^{cross}_{\bar{S},S}} I_{i;j,k},$$

$\Omega^{cross}_{A,B} \triangleq \{(i,j,k) | i \in A, j, k \in B\}$, A, B different categories.

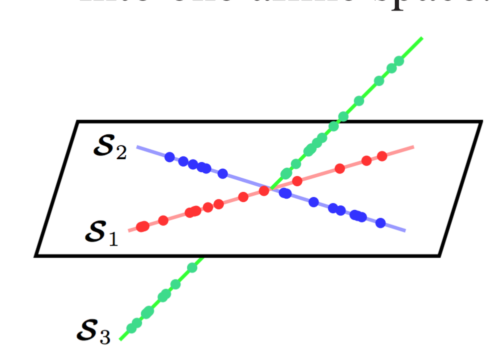
- A normalized form of above problem equivalent to inhomogeneous hypergraph partitioning.
- The correct categories of candidates can be detected with only 70% of samples as opposed to the benchmark.
- Only 30% of triples of mutual information $I_{i;j,k}$ requires to be estimated.



Irish House of Parliament election dataset (2002).

Application III: Subspace clustering

- Subspace segmentation has the goal to partition data according to their intrinsically embedded subspaces.
- The intrinsic affine space: p-dimensional.
- Construct ψ-uniform ($\psi > p + 1$, typically set to $p + 2$) hypergraphs $\mathcal{H} = (V, E, w)$.
- w_e, the weight of hyperedge e, is to characterize how likely the corresponding points in e fit into one affine space.



Union of three subspaces ($p = 1$)

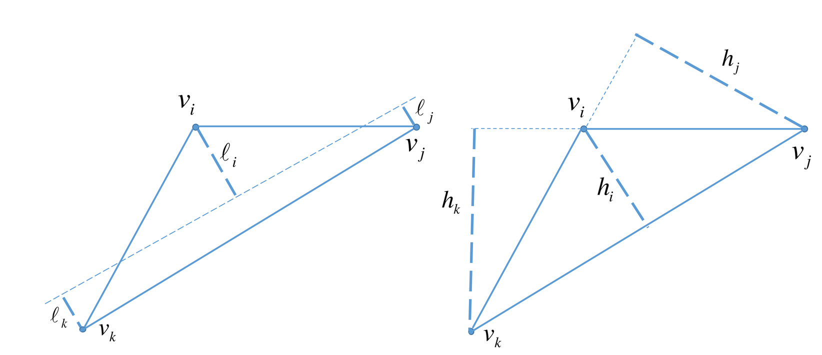
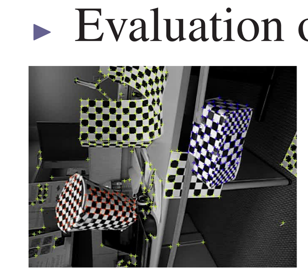


Illustration of the weight ($p = 1$) used for subspace segmentation.

$$d_e^{\mathrm{H}-1} = (\ell_i + \ell_j + \ell_k)/3$$
$$d_e^{\mathrm{H}-2} = (h_i + h_j + h_k)/3$$
$$d_e^{\mathrm{InH}}(\{v_i\}) = h_i$$
$$w_e = \exp(-d_e^2/\theta^2)$$

- Evaluation over the Hopkins 155 dataset:



Misclassification rates $e\%$. (MN: mean; MD: median)

	Two Motions								Three Motions							
Method	Chck.(78)		Trfc.(31)		Artc.(11)		All(120)		Chck.(26)		Trfc.(7)		Artc.(2)		All(115)	
	MN	MD	MN	MD	MN	MD	MN	MD	MN	MD	MN	MD	MN	MD	MN	MD
H+d_e^{H-1}	12.27	5.06	14.91	9.94	12.85	3.66	12.92	6.01	22.13	23.98	21.99	18.12	19.79	19.79	21.97	20.45
H+d_e^{H-2}	4.20	0.43	**0.33**	**0.00**	1.53	**0.10**	2.93	0.06	7.05	2.22	7.02	3.98	6.47	6.47	7.01	2.12
InH	**1.69**	**0.00**	0.61	0.22	**1.22**	0.62	**1.40**	**0.04**	**4.82**	**0.69**	**2.46**	**0.60**	**4.23**	**4.23**	**4.06**	**0.65**

References

[1] N. R. Devanur, S. Dughmi, R. Schwartz, A. Sharma, and M. Singh, "On the approximation of submodular functions," *arXiv preprint arXiv:1304.4948*, 2013.

[2] J. Huang, C. Guestrin *et al.*, "Uncovering the riffled independence structure of ranked data," *Electronic Journal of Statistics*, vol. 6, pp. 199–230, 2012.