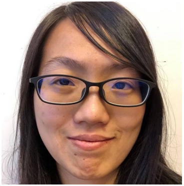
Discovering conflicting groups in signed networks

Ruo-Chun Tzeng[1] Bruno Ordozgoiti[2] Aristides Gionis[1,2]

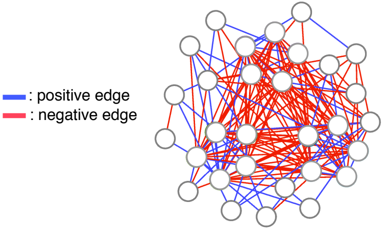


[1]KTH Royal Institute of Technology [2]Aalto University

34th Conference on Neural Information Processing Systems

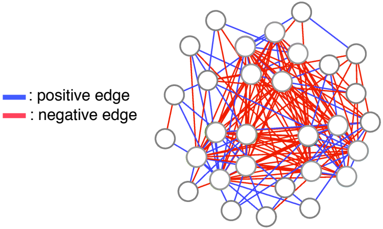
Problem: finding the conflicting groups

- Given a signed network, e.g., social networks with edge sign indicating agree/disagree.



Problem: finding the conflicting groups

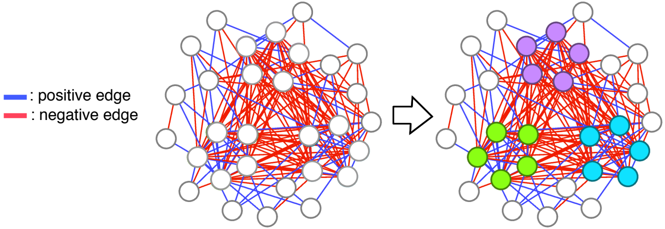
- Given a signed network, e.g., social networks with edge sign indicating agree/disagree.



- People form into groups with likely-minded or common enemies.

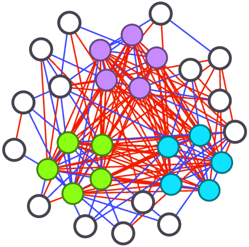
Problem: finding the conflicting groups

- Given a signed network, e.g., social networks with edge sign indicating agree/disagree.



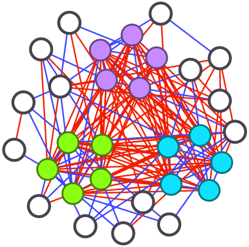
- People form into groups with likely-minded or common enemies.
- Our goal: find the conflicting groups with mostly $+$ intra-group edges and mostly $-$ inter-group edges.

Challenge: existence of neutral nodes



- Reason: e.g., not all people have strong opinions or firm stances.
- Neutral nodes might behave differently to both the conflicting groups or to themselves.

Challenge: existence of neutral nodes



- Reason: e.g., not all people have strong opinions or firm stances.
- Neutral nodes might behave differently to both the conflicting groups or to themselves.
- Methods partitioning the entire network such as signed clustering [4, 5] and correlation clustering [1] are not efficient.

Our approach: **S**pectral **C**onflicting **G**roup

- By extending the formulation of 2-PC [3] to $k \geq 2$ cases,

Our approach: **S**pectral **C**onflicting **G**roup

- By extending the formulation of 2-PC [3] to $k \geq 2$ cases,
- **Our contribution:** a framework SCG so that each group S_j can be found by solving

$$x^* = \underset{x \in \{k-j, 0, -1\}^n}{\operatorname{argmax}} \frac{x^T A^{(j-1)} x}{x^T x}, \tag{1}$$

where $A^{(j-1)}$ is the adjacency matrix of the graph after removing $\cup_{h<j} S_h$.

Our approach: **S**pectral **C**onflicting **G**roup

- By extending the formulation of 2-PC [3] to $k \geq 2$ cases,
- **Our contribution:** a framework SCG so that each group S_j can be found by solving

$$x^* = \underset{x \in \{k-j, 0, -1\}^n}{\operatorname{argmax}} \frac{x^T A^{(j-1)} x}{x^T x}, \tag{1}$$

where $A^{(j-1)}$ is the adjacency matrix of the graph after removing $\cup_{h<j} S_h$.

  - The j-th group $S_j = \{i : x_i^* = k - j\}$.

Our approach: **S**pectral **C**onflicting **G**roup

- By extending the formulation of 2-PC [3] to $k \geq 2$ cases,
- **Our contribution:** a framework SCG so that each group S_j can be found by solving

$$x^* = \underset{x \in \{k-j, 0, -1\}^n}{\operatorname{argmax}} \frac{x^T A^{(j-1)} x}{x^T x}, \tag{1}$$

where $A^{(j-1)}$ is the adjacency matrix of the graph after removing $\cup_{h<j} S_h$.

  - The j-th group $S_j = \{i : x_i^* = k - j\}$.
  - Nodes in other groups: $\{i : x_i^* = -1\}$.

Our approach: **S**pectral **C**onflicting **G**roup

- By extending the formulation of 2-PC [3] to $k \geq 2$ cases,
- **Our contribution:** a framework SCG so that each group S_j can be found by solving

$$x^* = \operatorname*{argmax}_{x \in \{k-j, 0, -1\}^n} \frac{x^T A^{(j-1)} x}{x^T x}, \tag{1}$$

where $A^{(j-1)}$ is the adjacency matrix of the graph after removing $\cup_{h<j} S_h$.

  - The j-th group $S_j = \{i : x_i^* = k - j\}$.
  - Nodes in other groups: $\{i : x_i^* = -1\}$.
  - Neutral nodes: $\{i : x_i^* = 0\}$.

Our approach: **S**pectral **C**onflicting **G**roup

- By extending the formulation of 2-PC [3] to $k \geq 2$ cases,
- **Our contribution:** a framework SCG so that each group S_j can be found by solving

$$x^* = \underset{x \in \{k-j, 0, -1\}^n}{\operatorname{argmax}} \frac{x^T A^{(j-1)} x}{x^T x}, \quad (1)$$

  where $A^{(j-1)}$ is the adjacency matrix of the graph after removing $\cup_{h<j} S_h$.

- It is APX-hard [2] to solve Eq (1) for $k = 2$.

Our approach: **S**pectral **C**onflicting **G**roup

- By extending the formulation of 2-PC [3] to $k \geq 2$ cases,
- **Our contribution:** a framework SCG so that each group S_j can be found by solving

$$x^* = \underset{x \in \{k-j, 0, -1\}^n}{\operatorname{argmax}} \frac{x^T A^{(j-1)} x}{x^T x}, \tag{1}$$

  where $A^{(j-1)}$ is the adjacency matrix of the graph after removing $\cup_{h<j} S_h$.
- It is APX-hard [2] to solve Eq (1) for $k = 2$.
- **Our contribution:** approximation algorithms by rounding the leading eigenvector of $A^{(j-1)}$.

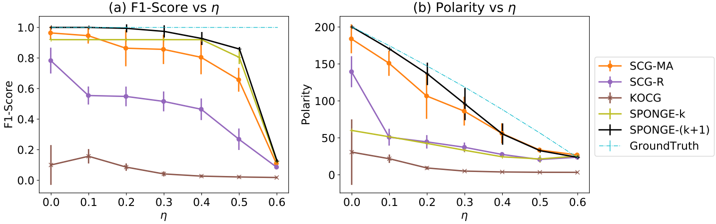
  (Minimum Angle) competitive in practice but hard to be analyzed.
  (Randomized) tight $\mathcal{O}(\sqrt{n})$ integrality gap when $k = 2$.

Experiment Results

- Real-world networks:

	Bitcoin	WikiVote	Referendum	Slashdot	WikiConflict	Epinions	Wikipolitics
$\|V\|$	5 881	7 115	10 884	82 140	116 717	131 580	138 587
$\|E\|$	21 492	100 693	251 406	500 481	2 026 646	711 210	715 883
$\|E_-\|/\|E\|$	0.2	0.2	0.1	0.2	0.6	0.2	0.1
SCG-MA	**14.6**	**45.5**	**84.9**	**37.8**	**102.6**	**88.8**	**57.5**
SCG-R	5.0	9.7	39.8	7.3	16.2	39.4	5.5
KOCG	4.4	5.5	8.8	2.6	4.5	8.7	4.8
SPONGE-k	5.0	15.8	41.5	—	—	—	—
SPONGE-(k+1)	0.8	1.0	1.0	—	—	—	—

- Synthetic:



Reference I

Nikhil Bansal, Avrim Blum, and Shuchi Chawla.
Correlation clustering.
Machine learning, 2004.

Aditya Bhaskara, Moses Charikar, Rajsekar Manokaran, and Aravindan Vijayaraghavan.
On quadratic programming with a ratio objective.
In *Proc. of ICALP*. Springer, 2012.

Francesco Bonchi, Edoardo Galimberti, Aristides Gionis, Bruno Ordozgoiti, and Giancarlo Ruffo.
Discovering polarized communities in signed networks.
In *Proc. of CIKM*, 2019.

Kai-Yang Chiang, Joyce Jiyoung Whang, and Inderjit S Dhillon.
Scalable clustering of signed networks using balance normalized cut.
In *Proc. of CIKM*, 2012.

Mihai Cucuringu, Peter Davies, Aldo Glielmo, and Hemant Tyagi.
Sponge: A generalized eigenproblem for clustering signed networks.
In *Proc. of AISTATS*, 2019.