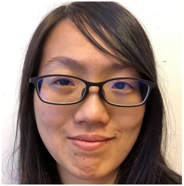
Discovering conflicting groups in signed networks

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

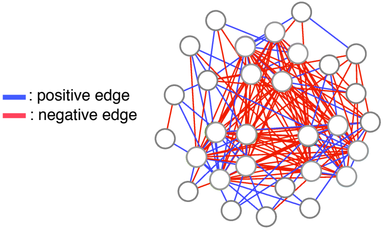


[1]KTH Royal Institute of Technology [2]Aalto University

34th Conference on Neural Information Processing Systems

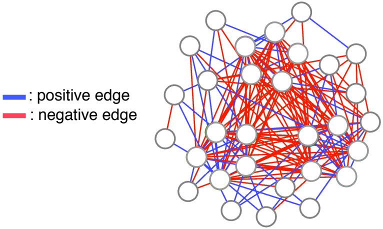
Motivation

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



Motivation

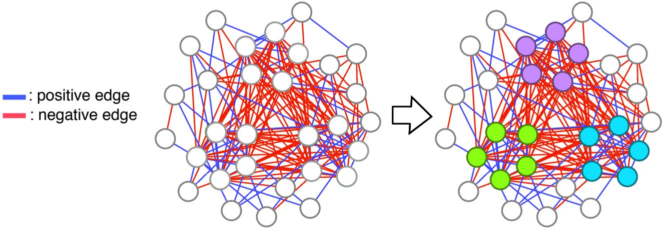
- Given a signed network, e.g., social networks with edge sign indicating agree/disagree.



- People form groups with like-minded or those with common enemies.

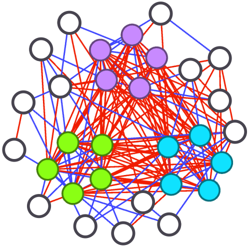
Motivation

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



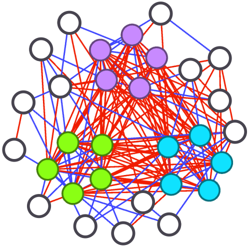
- People form groups with like-minded or those with common enemies.
- Our goal: find the conflicting groups with mostly $+$ intra-group edges and mostly $-$ inter-group edges.

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 [5] and correlation clustering [1] are not efficient.

Related Work: 2PC [3]

- Given a signed network $G = (V, E_+ \cup E_-)$, the goal is to find $k = 2$ conflicting groups

Related Work: 2PC [3]

- Given a signed network $G = (V, E_+ \cup E_-)$, the goal is to find $k = 2$ conflicting groups
- Find $S_1, \cdots, S_k \subset V$ that maximize

$$\frac{\sum_{h\in[k]}(|E_+(S_h)| - |E_-(S_h)|) + \sum_{h\neq l\in[k]}(|E_-(S_h, S_\ell)| - |E_+(S_h, S_\ell)|)}{|\cup_{h\in[k]} S_h|}, \tag{1}$$

where $E(S_h, S_\ell) = \{(i,j) \in E : i \in S_h, j \in S_\ell\}$ and $E(S_h) = E(S_h, S_h)$.

  - Many balanced edges and few imbalanced edges.
  - Normalized by the total group size.

Related Work: 2PC [3]

- Given a signed network $G = (V, E_+ \cup E_-)$, the goal is to find $k = 2$ conflicting groups
- Find $S_1, \cdots, S_k \subset V$ that maximize

$$\frac{\sum_{h\in[k]}(|E_+(S_h)| - |E_-(S_h)|) + \sum_{h\neq l\in[k]}(|E_-(S_h, S_\ell)| - |E_+(S_h, S_\ell)|)}{|\cup_{h\in[k]} S_h|}, \tag{1}$$

where $E(S_h, S_\ell) = \{(i,j) \in E : i \in S_h, j \in S_\ell\}$ and $E(S_h) = E(S_h, S_h)$.

  - Many balanced edges and few imbalanced edges.
  - Normalized by the total group size.
- Express Eq (1) as

$$\max_{S_1\cap S_2=\emptyset} \frac{\sum_{h\in[k]} \sum_{(i,j)\in E(S_h)} A_{i,j} + \sum_{h\neq\ell\in[k]} \sum_{(i,j)\in E(S_h,S_\ell)}(-A_{i,j})}{|\cup_{h\in[k]} S_h|} \tag{2}$$

Related Work: 2PC [3]

- Then, rewrite Eq (2) as

$$\max_{x \in \{-1,0,1\}^n} \frac{x^T A x}{x^T x} \tag{3}$$

  - APX-Hard [2] and best $\mathcal{O}(n^{1/3})$-approx exists but less practical.

Related Work: 2PC [3]

- Then, rewrite Eq (2) as

$$\max_{x \in \{-1,0,1\}^n} \frac{x^T A x}{x^T x} \tag{3}$$

  - APX-Hard [2] and best $\mathcal{O}(n^{1/3})$-approx exists but less practical.
- Provided with a $O(n^{1/2})$-approx randomized algorithm.

Related Work: 2PC [3]

- Then, rewrite Eq (2) as

$$\max_{x \in \{-1,0,1\}^n} \frac{x^T A x}{x^T x} \tag{3}$$

  - APX-Hard [2] and best $\mathcal{O}(n^{1/3})$-approx exists but less practical.
- Provided with a $O(n^{1/2})$-approx randomized algorithm.
- Cons: $x \in \{-1, 0, 1\}^n$ can only represent $k = 2$ conflicting groups.

Our approach: detecting $k \geq 2$ conflicting groups

- Extend Eq (2) as

$$\max_{S_1,\cdots,S_k} \frac{\sum_{h\in[k]} \sum_{(i,j)\in E(S_h)} A_{i,j} + \frac{1}{k-1} \sum_{h\neq\ell\in[k]} \sum_{(i,j)\in E(S_h,S_\ell)} (-A_{i,j})}{|\cup_{h\in[k]} S_h|} \quad (4)$$

  - The weighting is to prevent inter-group edges from dominating Eq (4).

Our approach: detecting $k \geq 2$ conflicting groups

- Extend Eq (2) as

$$\max_{S_1,\cdots,S_k} \frac{\sum_{h\in[k]} \sum_{(i,j)\in E(S_h)} A_{i,j} + \frac{1}{k-1} \sum_{h\neq\ell\in[k]} \sum_{(i,j)\in E(S_h,S_\ell)} (-A_{i,j})}{|\cup_{h\in[k]} S_h|} \quad (4)$$

  - The weighting is to prevent inter-group edges from dominating Eq (4).
- Notice the numerator of Eq (4) can be rewritten as $\langle A, XL_kX^T\rangle_F$,
  - where $L_k = kI_k - \mathbf{1}_{k\times k}$ is the Laplacian of a clique of size k and
  - $X \in \{0,1\}^{n\times k}$ is the group indicator with $X_{i,:} = (I_k)_{j,:}$ if $i \in S_j$.

Our approach: detecting $k \geq 2$ conflicting groups

- Extend Eq (2) as

$$\max_{S_1,\cdots,S_k} \frac{\sum_{h\in[k]} \sum_{(i,j)\in E(S_h)} A_{i,j} + \frac{1}{k-1} \sum_{h\neq\ell\in[k]} \sum_{(i,j)\in E(S_h,S_\ell)} (-A_{i,j})}{|\cup_{h\in[k]} S_h|} \quad (4)$$

  - The weighting is to prevent inter-group edges from dominating Eq (4).
- Notice the numerator of Eq (4) can be rewritten as $\langle A, X L_k X^T \rangle_F$,
  - where $L_k = kI_k - \mathbf{1}_{k\times k}$ is the Laplacian of a clique of size k and
  - $X \in \{0,1\}^{n\times k}$ is the group indicator with $X_{i,:} = (I_k)_{j,:}$ if $i \in S_j$.
- By expressing Eq (4) in terms of the eigendecomposition of L_k, ...

Our approach: detecting $k \geq 2$ conflicting groups

- Rewrite Eq (4) as $\displaystyle \max_{Y \in \mathbb{R}^{n \times (k-1)} \setminus \{\mathbf{0}\}} \frac{Tr(Y^T A Y)}{Tr(Y^T Y)}$

$$\text{subject to } Y_{i,j} = \begin{cases} c_j(k-j), & \text{if } i \in S_j \\ 0, & \text{if } i \in \cup_{h=1}^{j-1} S_h \text{ or } i \notin \cup_{h \in [k]} S_h \\ -c_j, & \text{if } i \in \cup_{h=j+1}^{k} S_h \end{cases} . \tag{5}$$

Our approach: detecting $k \geq 2$ conflicting groups

- Rewrite Eq (4) as $\displaystyle \max_{Y \in \mathbb{R}^{n\times(k-1)} \setminus \{\mathbf{0}\}} \frac{Tr(Y^T A Y)}{Tr(Y^T Y)}$

$$\text{subject to } Y_{i,j} = \begin{cases} c_j(k-j), & \text{if } i \in S_j \\ 0, & \text{if } i \in \cup_{h=1}^{j-1} S_h \text{ or } i \notin \cup_{h\in[k]} S_h \\ -c_j, & \text{if } i \in \cup_{h=j+1}^{k} S_h \end{cases}. \tag{5}$$

Our approach: detecting $k \geq 2$ conflicting groups

- Rewrite Eq (4) as $$\max_{Y \in \mathbb{R}^{n \times (k-1)} \setminus \{\mathbf{0}\}} \frac{Tr(Y^T A Y)}{Tr(Y^T Y)}$$

$$\text{subject to } Y_{i,j} = \begin{cases} c_j(k-j), & \text{if } i \in S_j \\ 0, & \text{if } i \in \cup_{h=1}^{j-1} S_h \text{ or } i \notin \cup_{h \in [k]} S_h \\ -c_j, & \text{if } i \in \cup_{h=j+1}^{k} S_h \end{cases} . \quad (5)$$

Our approach: detecting $k \geq 2$ conflicting groups

- Rewrite Eq (4) as $\max\limits_{Y \in \mathbb{R}^{n\times(k-1)} \setminus \{\mathbf{0}\}} \dfrac{Tr(Y^T A Y)}{Tr(Y^T Y)}$

$$\text{subject to } Y_{i,j} = \begin{cases} c_j(k-j), & \text{if } i \in S_j \\ 0, & \text{if } i \in \cup_{h=1}^{j-1} S_h \text{ or } i \notin \cup_{h\in[k]} S_h \\ -c_j, & \text{if } i \in \cup_{h=j+1}^{k} S_h \end{cases}. \tag{5}$$

Our approach: detecting $k \geq 2$ conflicting groups

- Rewrite Eq (4) as $$\max_{Y \in \mathbb{R}^{n \times (k-1)} \setminus \{\mathbf{0}\}} \frac{Tr(Y^T A Y)}{Tr(Y^T Y)}$$

$$\text{subject to } Y_{i,j} = \begin{cases} c_j(k-j), & \text{if } i \in S_j \\ 0, & \text{if } i \in \cup_{h=1}^{j-1} S_h \text{ or } i \notin \cup_{h \in [k]} S_h \\ -c_j, & \text{if } i \in \cup_{h=j+1}^{k} S_h \end{cases}. \quad (5)$$

- Our idea: suppose $S_1, \cdots, S_{j-1}$ are determined, find S_j by solving

$$x^* = \underset{x \in \{k-j, 0, -1\}^n}{\operatorname{argmax}} \frac{x^T A^{(j-1)} x}{x^T x}. \quad (6)$$

  - Let $A^{(0)} = A$ and $A^{(j-1)}$ results from removing $\cup_{h \in [j-1]} S_h$ from G.
  - After equation (6) is solved, we know $S_j = \{i : x_i^* = k - j\}$.
  - Repeat the same process to decide the remaining $S_{j+1}, \cdots, S_k$.

Our approach: detecting $k \geq 2$ conflicting groups

- Rewrite Eq (4) as $\max_{Y \in \mathbb{R}^{n \times (k-1)} \setminus \{\mathbf{0}\}} \frac{Tr(Y^T A Y)}{Tr(Y^T Y)}$

$$\text{subject to } Y_{i,j} = \begin{cases} c_j(k-j), & \text{if } i \in S_j \\ 0, & \text{if } i \in \cup_{h=1}^{j-1} S_h \text{ or } i \notin \cup_{h \in [k]} S_h \\ -c_j, & \text{if } i \in \cup_{h=j+1}^{k} S_h \end{cases}. \quad (5)$$

- Our idea: suppose $S_1, \cdots, S_{j-1}$ are determined, find S_j by solving

$$x^* = \operatorname*{argmax}_{x \in \{k-j, 0, -1\}^n} \frac{x^T A^{(j-1)} x}{x^T x}. \quad (6)$$

  - Let $A^{(0)} = A$ and $A^{(j-1)}$ results from removing $\cup_{h \in [j-1]} S_h$ from G.
  - After equation (6) is solved, we know $S_j = \{i : x_i^* = k - j\}$.
  - Repeat the same process to decide the remaining $S_{j+1}, \cdots, S_k$.

Our approach: detecting $k \geq 2$ conflicting groups

- Rewrite Eq (4) as $$\max_{Y \in \mathbb{R}^{n \times (k-1)} \setminus \{\mathbf{0}\}} \frac{Tr(Y^T A Y)}{Tr(Y^T Y)}$$

$$\text{subject to } Y_{i,j} = \begin{cases} c_j(k-j), & \text{if } i \in S_j \\ 0, & \text{if } i \in \cup_{h=1}^{j-1} S_h \text{ or } i \notin \cup_{h \in [k]} S_h \\ -c_j, & \text{if } i \in \cup_{h=j+1}^{k} S_h \end{cases}. \tag{5}$$

- Our idea: suppose $S_1, \cdots, S_{j-1}$ are determined, find S_j by solving

$$x^* = \operatorname*{argmax}_{x \in \{k-j, 0, -1\}^n} \frac{x^T A^{(j-1)} x}{x^T x}. \tag{6}$$

  - Let $A^{(0)} = A$ and $A^{(j-1)}$ results from removing $\cup_{h \in [j-1]} S_h$ from G.
  - After equation (6) is solved, we know $S_j = \{i : x_i^* = k - j\}$.
  - Repeat the same process to decide the remaining $S_{j+1}, \cdots, S_k$.

Our approach: detecting $k \geq 2$ conflicting groups

- Rewrite Eq (4) as $\max_{Y \in \mathbb{R}^{n \times (k-1)} \setminus \{\mathbf{0}\}} \frac{Tr(Y^T A Y)}{Tr(Y^T Y)}$

$$\text{subject to } Y_{i,j} = \begin{cases} c_j(k-j), & \text{if } i \in S_j \\ 0, & \text{if } i \in \cup_{h=1}^{j-1} S_h \text{ or } i \notin \cup_{h \in [k]} S_h \\ -c_j, & \text{if } i \in \cup_{h=j+1}^{k} S_h \end{cases}. \tag{5}$$

- Our idea: suppose $S_1, \cdots, S_{j-1}$ are determined, find S_j by solving

$$x^* = \operatorname*{argmax}_{x \in \{k-j, 0, -1\}^n} \frac{x^T A^{(j-1)} x}{x^T x}. \tag{6}$$

  - Let $A^{(0)} = A$ and $A^{(j-1)}$ results from removing $\cup_{h \in [j-1]} S_h$ from G.
  - After equation (6) is solved, we know $S_j = \{i : x_i^* = k - j\}$.
  - Repeat the same process to decide the remaining $S_{j+1}, \cdots, S_k$.

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

Algorithm 1: SCG(A, k)

$A^{(0)} \leftarrow A$;

for $t = 1, \cdots, k-1$ **do**

  $r^{(t)} \leftarrow$ Solve-Max-DRQ($A^{(t-1)}, k-t$) **if** $t < k-1$ **then**

    $S_t \leftarrow \{i \notin \cup_{j=1}^{t-1} S_j : |r_i^{(t)}| = (k-t)\}$;

    $A^{(t)} \leftarrow A^{(t-1)}$;

    $A_{i,:}^{(t)} \leftarrow 0_{1\times n}$ and $A_{:,i}^{(t)} \leftarrow 0_{n\times 1}$ for all $i \in S_t$

  else $S_{k-1} \leftarrow \{i \notin \cup_{j=1}^{t-1} S_j : r_i^{(t)} = 1\}$ and

   $S_k \leftarrow \{i \notin \cup_{j=1}^{t-1} S_j : r_i^{(t)} = -1\}$;

end

return $S_1, \ldots, S_k$;

Our approach: Solve-Max-DRQ

$$x^* = \operatorname*{argmax}_{x \in \{k-j, 0, -1\}^n} \frac{x^T A^{(j-1)} x}{x^T x}. \tag{6}$$

Our approach: Solve-Max-DRQ

$$x^* = \underset{x \in \{k-j, 0, -1\}^n}{\operatorname{argmax}} \frac{x^T A^{(j-1)} x}{x^T x}. \tag{6}$$

- APX-Hard [2] for $k = 2$ and practical $\mathcal{O}(n^{1/2})$-approx by 2PC [3].

Our approach: Solve-Max-DRQ

$$x^* = \underset{x \in \{k-j,0,-1\}^n}{\operatorname{argmax}} \frac{x^T A^{(j-1)} x}{x^T x}. \tag{6}$$

- APX-Hard [2] for $k = 2$ and practical $\mathcal{O}(n^{1/2})$-approx by 2PC [3].
- Our approach is based on rounding the leading eigenvector of $A^{(j-1)}$ to a vector in $\{k - j, 0, -1\}^n$.

Algorithm 1: Solve-Max-DRQ(A, q)

Input : Square and symmetric matrix A, and positive integer q.
Output: The rounded vector $r \in \{0, -1, q\}^n$.
$v \leftarrow$ the leading eigenvector of A;
$(d_1, r_1) \leftarrow \text{Round}(v, q)$; // $d_1 = \sin\theta(v, r_1)$
$(d_2, r_2) \leftarrow \text{Round}(-v, q)$; // $d_2 = \sin\theta(v, r_2)$
if $d_1 \leq d_2$ **then** $r \leftarrow r_1$;
else $r \leftarrow r_2$;
return r ;

Deterministic Rounding: Minimum Angle (MA)

- Rounded $r = \operatorname{argmin}_{u \in \{q,0,-1\}^n} \sin\theta(v,u)$.
- Guaranteed to finish in $\mathcal{O}(n^2)$.
- For practical consideration, implement an $\mathcal{O}(n)$ algorithm.

Algorithm 2: MA(v, q)

$\{i_k\}_{k=1}^n \leftarrow$ Sort v and return the indexes such that $v_{i_1} \geq \cdots \geq v_{i_n}$;
$(d, u^*) \leftarrow (\infty, 0)$;
$(k_1, k_2) \leftarrow (0, n+1)$;
while $k_1 < k_2$ **do**
  $u_1 \leftarrow$ set the i_{k_1+1}-th element of u^* to q;
  $u_2 \leftarrow$ set the i_{k_2-1}-th element of u^* to -1;
  if $\min\{\sin\theta(v,u_1), \sin\theta(v,u_2)\} \geq d$ **then** break;
  if $\sin\theta(v,u_1) < \sin\theta(v,u_2)$ **then**
   $(k_1, d, u^*) \leftarrow (k_1+1, \sin\theta(v,u_1), u_1$;
  else $(k_2, d, u^*) \leftarrow (k_2-1, \sin\theta(v,u_2), u_2)$;
end
return (d, u^*);

Randomized Rounding (R)

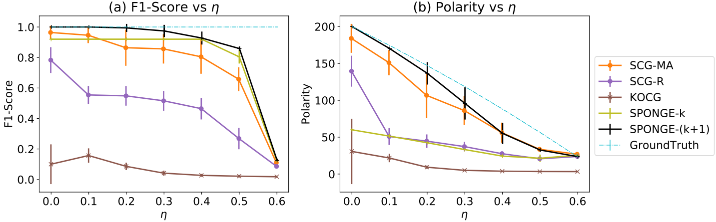
- Generalize the randomized approach of 2PC [3].
- Round v to r by setting $r_i = \begin{cases} q, & \text{w.p. } |v_i|/q \\ -1, & \text{w.p. } |v_i| \end{cases}$.
- It gives a $\mathcal{O}(qn^{1/2})$-approx to the Max-DRQ problem, which is tight upto a factor of q.

Experiment Results

- Real-world networks:

	Bitcoin	WikiVote	Referendum	Slashdot	WikiConflict	Epinions	Wikipolitics
$\lvert V\rvert$	5 881	7 115	10 884	82 140	116 717	131 580	138 587
$\lvert E\rvert$	21 492	100 693	251 406	500 481	2 026 646	711 210	715 883
$\lvert E_{-}\rvert/\lvert E\rvert$	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]	4.4	5.5	8.8	2.6	4.5	8.7	4.8
SPONGE-k [5]	5.0	15.8	41.5	—	—	—	—
SPONGE-(k+1) [5]	0.8	1.0	1.0	—	—	—	—

- Synthetic:



Summary

- An efficient optimization framework to find conflicting groups.
  - By rewriting the objective and analyzing the eigenspaces of the Laplacian of a clique of size k, finding each conflicting group reduces to solving a discrete optimization problem.
  - Present approximation algorithms with provable guarantee.
- Future works:
  - Is it possible to improve $\mathcal{O}(n^{1/2})$-approx by other approach?
  - What causes the empirical difference in real and synthetic graphs?

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.

Lingyang Chu, Zhefeng Wang, Jian Pei, Jiannan Wang, Zijin Zhao, and Enhong Chen.
Finding gangs in war from signed networks.
In *Proc. of KDD*, 2016.

Reference II

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