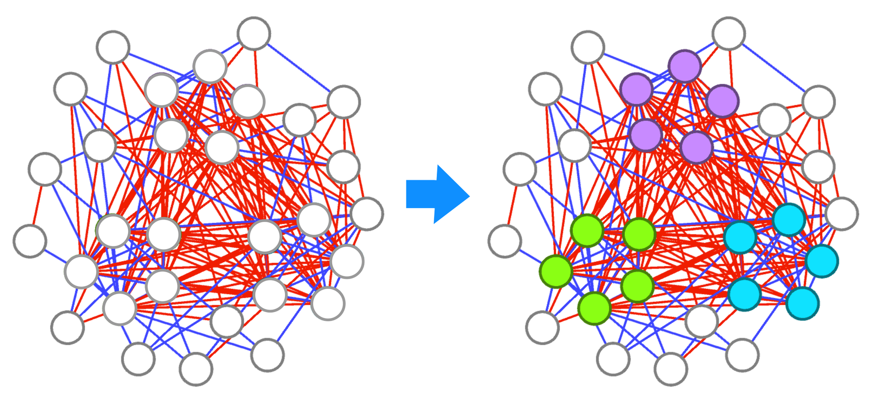
Discovering conficting groups in signed networks

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

[1]KTH Royal Institute of Technology, Sweden [2]Aalto University, Finland

Detecting conflicting groups

Given a signed network, our goal is to detect k **conflicting groups**, which are k disjoint node subsets that are mostly positively (+) linked internally and mostly negatively (−) linked to the other $k-1$ groups.



Challenge: There may exist **neutral** nodes whose interactions are **inconsistent** with the structure of the conflicting groups. Therefore, methods that partition the entire network, such as correlation clustering [1] and signed clustering [5] are inefficient.

By extending the formulation of the 2-polarized-clustering problem (2PC) [3], given an integer k as input, the k conflicting groups can be detected by

$$\max_{S_1,\ldots,S_k} \frac{\sum_{h\in[k]}(|E_+(S_h)| - |E_-(S_h)|) + \frac{1}{k-1}\sum_{h\neq \ell\in[k]}(|E_-(S_h,S_\ell)| - |E_+(S_h,S_\ell)|)}{|\cup_{h\in[k]} S_h|}, \qquad (1)$$

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

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

We introduce a group indicator matrix $\mathbf{X} \in \{0,1\}^{n\times k}$, where $\mathbf{X}_{i,:} = (\mathbf{I}_k)_{j,:}$ if node $i \in S_j$. Using the Laplacian $\mathbf{L}_k = k\mathbf{I} - \mathbf{1}_{k\times k}$ and exploiting its spectral properties, the objective in Equation (1) is equivalent to

$$\max_{\mathbf{Y}\in\mathbb{R}^{n\times(k-1)}\setminus\{\mathbf{0}\}} \frac{\mathrm{Tr}(\mathbf{Y}^T\mathbf{A}\mathbf{Y})}{\mathrm{Tr}(\mathbf{Y}^T\mathbf{Y})} \quad \text{subject to} \quad \mathbf{Y}_{i,j} = \begin{cases} c_j(k-j), & \text{if } i \in S_j \\ 0, & \text{if } i \in \cup_{h=1}^{j-1} S_h \text{ or } i \notin \cup_{h\in[k]} S_h \\ -c_j, & \text{if } i \in \cup_{h=j+1}^{k} S_h \end{cases}.$$

Main Idea: Assuming that $S_1, \ldots, S_{j-1}$ have been determined, we find S_j by solving an instance of the Max-DRQ problem:

$$\mathbf{x}^* = \operatorname*{argmax}_{\mathbf{x}\in\{k-j,0,-1\}^n} \frac{\mathbf{x}^T\mathbf{A}^{(j-1)}\mathbf{x}}{\mathbf{x}^T\mathbf{x}}. \qquad (2)$$

Let $\mathbf{A}^{(j-1)}$ be the adjacency matrix after removing $\cup_{h\in[j-1]} S_h$ from G, with $\mathbf{A}^{(0)} = \mathbf{A}$.
By solving Equation (2) we find $S_j = \{i : x_i^* = k-j\}$.
We repeat the process to find the remaining conflicting groups $S_{j+1}, \ldots, S_k$.

Solve-Max-DRQ

The Max-DRQ problem is APX-Hard [2]. The problem has been studied only for $k=2$. There exist an SDP-based $\tilde{\mathcal{O}}(n^{1/3})$-approximation algorithm [2] and a more practical $\mathcal{O}(n^{1/2})$-approximation algorithm [3]. For $k > 2$, no algorithm for Max-DRQ is known.

Our approach is based on rounding the leading eigenvector $\mathbf{v}$ of $\mathbf{A}^{(j)}$ to a vector in $\mathbf{r} \in \{k-j, 0, -1\}^n$. We use the following rounding schemes:

MinimumAngle (MA) Set $\mathbf{r} = \mathrm{argmin}_{\mathbf{u}\in\{k-j,0,-1\}^n} \sin\theta(\mathbf{v},\mathbf{u})$.

- The minimizer can be found in time $\mathcal{O}(n^2)$.
- For practical considerations in the experiments we use a $\mathcal{O}(n)$ heuristic method.

RandomRound (R) Each entry of $\mathbf{r}$ is rounded by

$$\mathbf{r}_i = \begin{cases} (k-j), & \text{with prob. } |\mathbf{v}_i|/(k-j) \text{ if } v_i > 0 \\ -1, & \text{with prob. } |\mathbf{v}_i| \text{ if } v_i < 0 \end{cases}.$$

This randomized algorithm gives an $(k-j)n^{1/2}$-approximation algorithm to the Max-DRQ problem, which generalizes the previous result for the 2PC problem [3].

Theorem Let $(\mathbf{A}, \mathbf{v}, q)$ be an instance of Max-DRQ. The RandomRound algorithm returns a solution with approximation guarantee $qn^{1/2}$.

Theorem Let OPT be the optimum to the Max-DRQ problem. There exists a problem instance such that $\lambda_1(\mathbf{A}) \geq \mathrm{OPT}\cdot\Omega(n^{1/2})$.

Corollary The integrality gap of RandomRound algorithm is $\Omega(n^{1/2})$ and it is tight up to a factor of q.

Pseudocode: SCG framework

```
Algorithm 1: SCG (A, k)                    Spectral Conflicting Group detection
Input  : A is the adjacency matrix of the signed network; k is the number of groups.
Output : Groups S_1, ..., S_k.
A^(0) ← A;
for t = 1, ..., k−1 do
  | r^(t) ← Solve-Max-DRQ (A^(t−1), k − t) ;
  | if t < k − 1 then
  |   | S_t ← {i ∉ ∪_{j=1}^{t−1} S_j : |r_i^(t)| = (k − t)};
  |   | A^(t) ← A^(t−1);
  |   | A^(t)_{i,:} ← 0_{1×n} and A^(t)_{:,i} ← 0_{n×1} for all i ∈ S_t ;   // Remove edges E(S_t, V)
  | else S_{k−1} ← {i ∉ ∪_{j=1}^{t−1} S_j : r_i^(t) = 1} and S_k ← {i ∉ ∪_{j=1}^{t−1} S_j : r_i^(t) = −1} ;
end
return S_1, ..., S_k;
```

Experiments

The proposed SCG algorithm finds high-quality conflicting groups in both real-world networks and networks generated by the *modified signed stochastic block model* (m-SSBM). The signed-network clustering method SPONGE [5] performs pretty well in m-SSBM networks but fails in real-world networks. KOCG [4] finds groups of very small size.

Table: Real-world networks. Polarity objective (Equation (1)) achieved by the proposed method (SCG) and baselines. Dashes (—) indicate that a method exceeds the memory limit.

	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	—	—	—	—

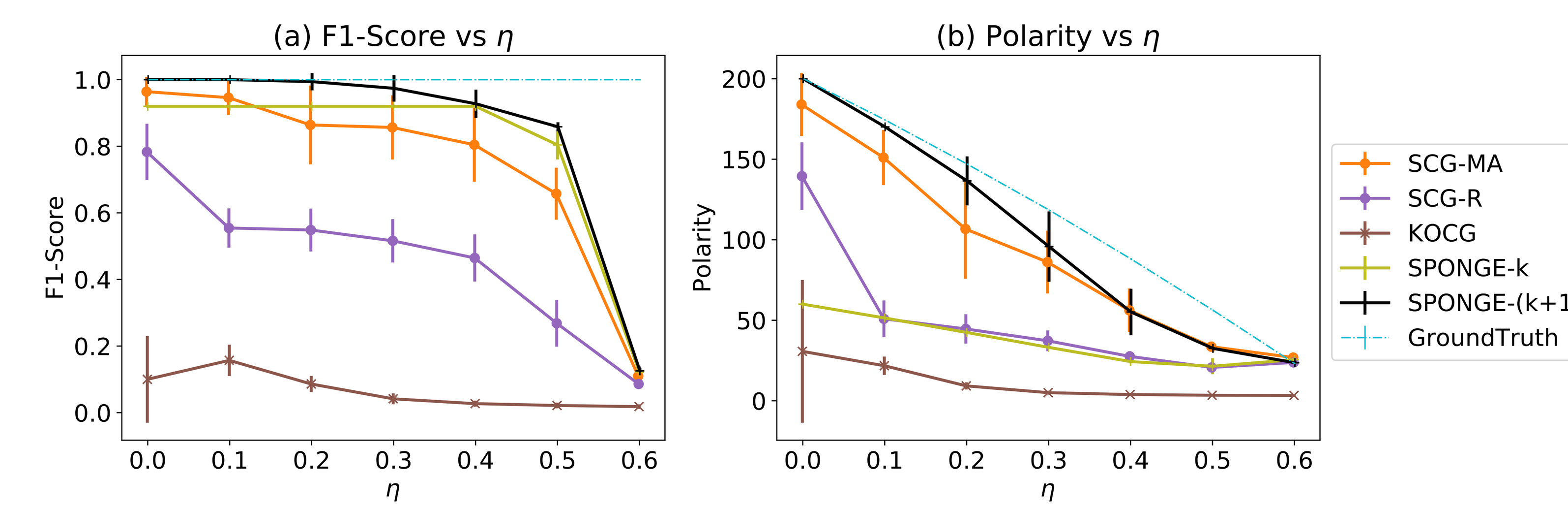


Figure: Results on modified signed stochastic block model (m-SSBM). We show F_1-score and Polarity as a function of a noise parameter η.

References

[1] N. Bansal, A. Blum, and S. Chawla. Correlation clustering. *Machine learning*, 2004.

[2] A. Bhaskara, N. Charikar, R. Manokaran, and A. Vijayaraghavan. On quadratic programming with a ratio objective. In *Proc. of ICALP*. Springer, 2012.

[3] F. Bonchi, E. Galimberti, A. Gionis, B. Ordozgoiti, and G. Ruffo. Discovering polarized communities in signed networks. In *Proc. of CIKM*, 2019.

[4] L. Chu, Z. Wang, J. Pei, J. Wang, Z. Zhao, and E. Chen. Finding gangs in war from signed networks. In *Proc. of KDD*, 2016.

[5] M. Cucuringu, P. Davies, A. Glielmo, and H. Tyagi. Sponge: A generalized eigenproblem for clustering signed networks. In *Proc. of AISTATS*, 2019.

Social Mining & Big Data Ecosystem