UNIVERSITÀ
DEGLI STUDI
DELL'AQUILA

UNIVERSITÀ
DELLA CALABRIA

AI & Data Science Lab
@DIMES Dept.
MACHINES, LANGUAGES &
NETWORKS team

Polarized Communities meet Densest Subgraph: Efficient and Effective Polarization Detection in Signed Networks

ACM Transactions on Knowledge Discovery from Data (TKDD), 2025

Francesco Gullo
University of L'Aquila, Italy

Domenico Mandaglio
University of Calabria, Italy

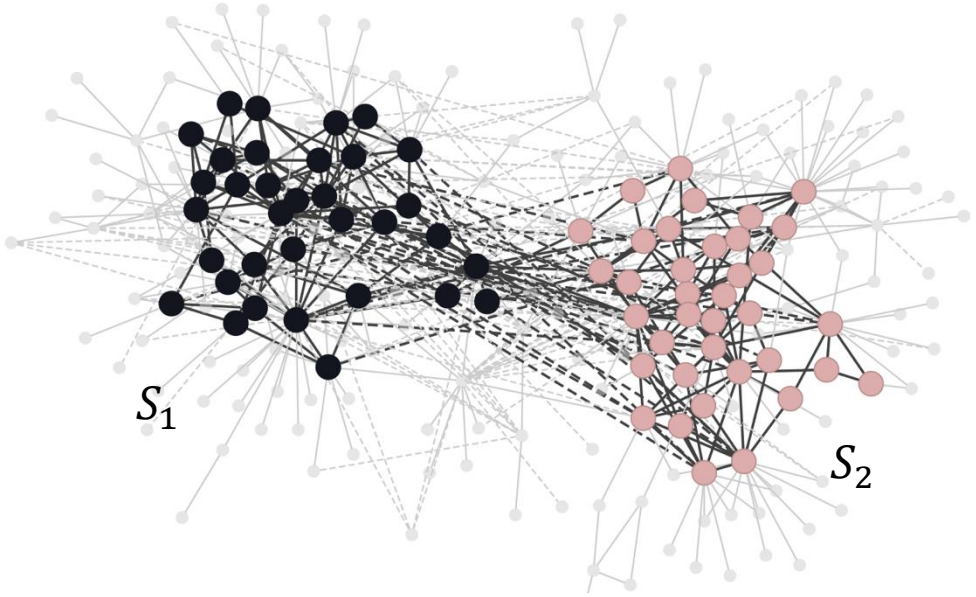
Andrea Tagarelli
University of Calabria, Italy

SIGKDD
SINGAPORE

20th January 2026

Polarization Detection in Social Media

- In real-world (social) graphs interactions can be *friendly* or *antagonistic* (e.g. friend/foe, trust/distrust, agree/disagree).
- Increase of *polarization* around controversial issues is a growing concern, with important societal fallouts.
- Signed networks naturally model this setting: nodes represents users, and edges are labeled positive (+) or negative (-).



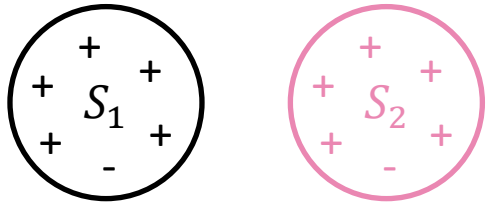
An example of two polarized communities in the Congress network. Solid edges are positive, while dashed edges are negative.

Key question: How can we identify the core groups driving the polarization within a big network?

2-Polarized-Communities (2PC) Problem

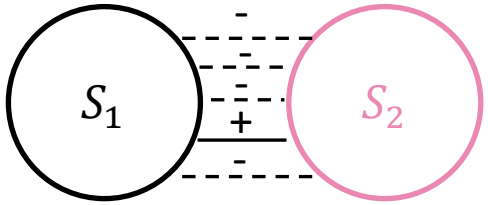
Find two non-overlapping communities of nodes, S_1 and S_2, that satisfy three key requirements:

(**R1**) Internal Cohesion



The majority of edges *within* S_1 and S_2 are positive

(**R2**) External Opposition



The majority of edges *across* S_1 and S_2 are negative

(**R3**) Significant Density

A large number of edges satisfying R1-R2 relative to the total number of nodes in S_1 and S_2

Bonchi Francesco, et al. "Discovering polarized communities in signed networks." CIKM 2019.

Quantifying Polarization: The Polarity Objective Function

Given a signed graph $G = (V, E^+, E^-)$, find two non-overlapping communities, $S_1, S_2 \subseteq V$, that **maximizes** the *polarity* objective function:

$$p(S_1, S_2; G) = \frac{\sum_{i \in \{1,2\}} (|E^+(S_i)| - |E^-(S_i)|) + |E^-(S_1, S_2)| - |E^+(S_1, S_2)|}{|S_1 \cup S_2|}$$

where $E^{\pm}(S_i, S_j) = \{(u, v) \in E^{\pm} : u \in S_i, v \in S_j\}$ and $E^{\pm}(S_i) = E^{\pm}(S_i, S_i)$

Idea: prefer the S_1, S_2 that:

- have many **compliant** and few **noncompliant** edges (**R1-R2**)
- the size of $S_1 \cup S_2$ is as small as possible (**R3**)

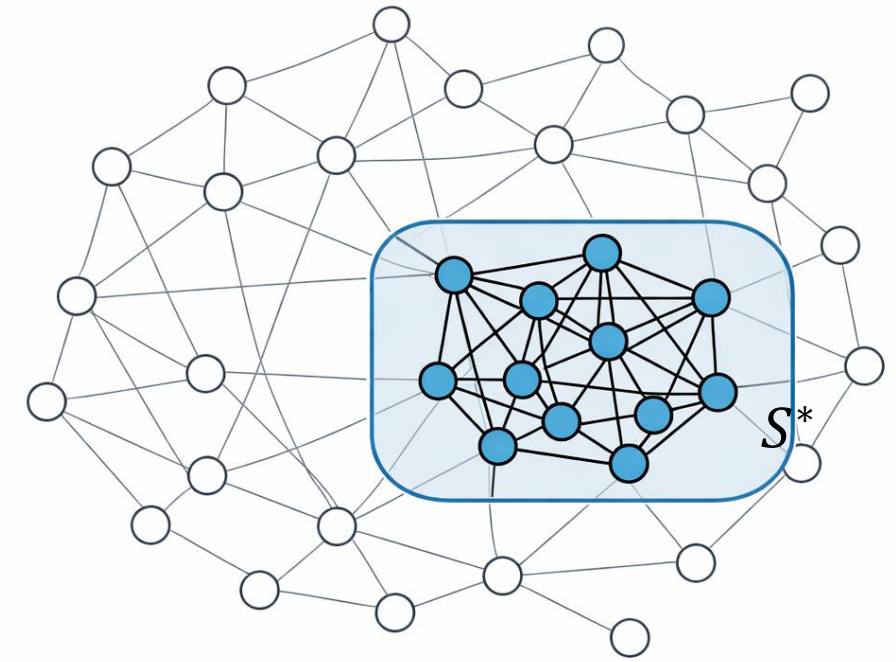
Bonchi Francesco, et al. "Discovering polarized communities in signed networks." CIKM 2019.

The Densest Subgraph (DS) Problem

Given an undirected graph $G = (V, E)$,
find $S^* \subseteq V$ such that:

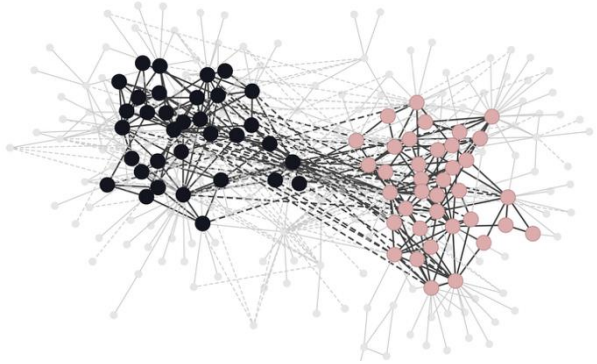
$$S^* = \text{argmax}_{S \subseteq V} \frac{|E(S)|}{|S|} = \text{argmax}_{S \subseteq V} \frac{\sum_{v \in S} d_S(v)}{|S|}$$

- Unlike 2PC, the DS problem can be solved exactly in polynomial time
- An effective and efficient approximation solution is the **greedy peeling algorithm**: it iteratively removes the node with the lowest degree and returns the intermediate subgraph with the highest density.



Andrew V Goldberg. 1984. Finding a maximum density subgraph. (1984).

Bridging 2PC and DS



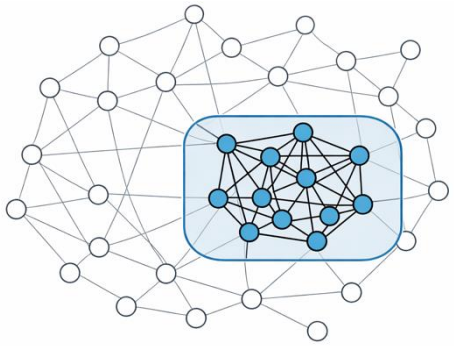
2PC polarity (reframed)

$$p(S_1, S_2; G) = \frac{\sum_{u \in S_1 \cup S_2} d_S^{\pm}(u)}{|S_1 \cup S_2|}$$

(average net degree balances)

Key concept: *Net Degree Balance*
For any node u, its net degree balance w.r.t. a pair of polarized communities $S = \{S_1, S_2\}$ is

$$d_S^{\pm}(u) = \underbrace{\#\text{compliant edges (incident to } u)}_{} - \underbrace{\#\text{noncompliant edges (incident to } u)}_{}$$



DS density

$$\frac{\sum_{v \in S} d_S(v)}{|S|}$$

(average simple degrees)

Key insight: The 2PC problem is a generalization of the DS problem.
This connection allows us to adapt powerful DS algorithms to solve the 2PC problem.

Greedy-2PC: a greedy peeling algorithm for 2PC

Algorithm 2 Greedy-2PC

Input: Signed graph $G = (V, E^+, E^-)$
Output: A pair $\widehat{S} = \{S_1, S_2\}$ of polarized communities

1: $S^{\text{SEED}} \leftarrow$ EIGEN-FULL(G) {Algorithm 1}
2: $\widehat{S} \leftarrow S^{\text{SEED}}$, $S^n \leftarrow S^{\text{SEED}}$, $i \leftarrow n$
3: **while** $|S_1^i \cup S_2^i| > 1$ **do**
4: $u = \arg\min_{v \in S_1^i \cup S_2^i} d^{\pm}_{S^i}(v)$
5: remove u from S_1^i or S_2^i in S^i
6: **if** $p(S_1^i, S_2^i; G) > p(\widehat{S}_1, \widehat{S}_2; G)$ **then**
7: $\widehat{S} \leftarrow S^i$
8: **end if**
9: $i \leftarrow i - 1$
10: **end while**
11: **return** $\widehat{S}$

Algorithm 1 EIGEN-FULL

Input: Signed graph $G = (V, E^+, E^-)$
Output: A pair $S = \{S_1, S_2\}$ of polarized communities

1: Compute $\mathbf{z}^*$, the eigenvector corresponding to the largest eigenvalue λ_1 of the signed adjacency matrix $\mathbf{A}$ of G
2: $S_1 = \{u \in V : \mathbf{z}^*_u \geq 0\}, S_2 = \{u \in V : \mathbf{z}^*_u < 0\}$

Observation. The eigenvector z^* corresponding to the largest eigenvalue λ_1 of $\mathbf{A}$ is an optimal solution of the relaxed problem of 2PC.

Moses Charikar. "Greedy approximation algorithms for finding dense components in a graph." APPROX 2003.

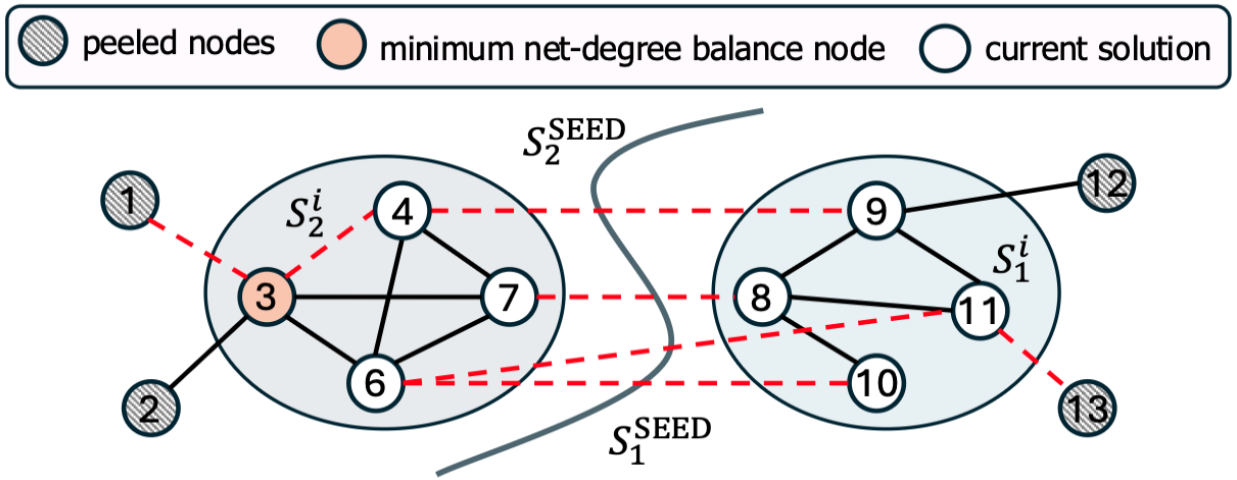
Greedy-2PC: a greedy peeling algorithm for 2PC

Algorithm 2 Greedy-2PC

Input: Signed graph $G = (V, E^+, E^-)$

Output: A pair $\widehat{S} = \{S_1, S_2\}$ of polarized communities

$S^{\text{SEED}} \leftarrow$ EIGEN-FULL(G) {Algorithm 1}
$\widehat{S} \leftarrow S^{\text{SEED}}$, $S^n \leftarrow S^{\text{SEED}}$, $i \leftarrow n$
while $|S_1^i \cup S_2^i| > 1$ **do**
  $u = \arg\min_{v \in S_1^i \cup S_2^i} d^{\pm}_{S^i}(v)$
  remove u from S_1^i or S_2^i in S^i
  if $p(S_1^i, S_2^i; G) > p(\widehat{S}_1, \widehat{S}_2; G)$ **then**
    $\widehat{S} \leftarrow S^i$
  end if
  $i \leftarrow i - 1$
end while
return $\widehat{S}$



Overview of Greedy-2PC at the *i*-th iteration

Key Benefits of Greedy-2PC

Highly efficient.
Runs in linear time, $O(|V| + |E|)$.

Effective in practice.
Consistently outperforms SOTA methods.

Theoretical (additive) guarantees.
Under condition $S_1^* \subseteq S_1^{\text{seed}}, S_2^* \subseteq S_2^{\text{seed}}$, $p(\widehat{S_1}, \widehat{S_2}; G) \geq OPT - c$, where c is a term related to the (maximum) #noncompliant edges of the peeled nodes.

Simple to implement.

Experimental Evaluation: Datasets

Real-World Datasets

11 real-world networks + 2 augmented (dummy vertices with random edges matching average degree, preserving negative-edge ratio)

Synthetic Datasets

We used the modified *Signed Stochastic Block Model* (m-SSBM) to generate synthetic graphs, controlling over ground-truth community size and noise level (η).

dataset	$\|V\|$	$\|E\|$	$\|E^-\|/\|E\|$
Bitcoin	5 881	21 492	0.152
Cloister	18	125	0.552
Congress	219	521	0.205
Epinions	131 580	711 210	0.171
HTribes	16	58	0.500
Slashdot	82 140	500 481	0.239
TwitterRef	10 884	251 406	0.051
WikiCon	116 717	2 026 646	0.628
WikiEle	7 115	100 693	0.221
WikiPol	138 587	715 883	0.123
Word	4 962	47 088	0.199
Epinions-8|V|	1 052 640	12 290 250	0.171
Wikiconflict-8|V|	933 736	34 707 406	0.628

Experimental Evaluation: Competing Methods and Evaluation Criteria

Competing methods

- *SOTA for 2PC:* Neural2PC[1] (neural), RH[2] (heuristic), Eigen[3] & R-Eigen[3] (spectral).
- *Related Baselines:* SPONGE[4], BNC[5] & SSSNet[6] (signed graph clustering), Timbal[7] (balanced subgraph), Pivot[8] (correlation clustering).

Evaluation criteria

- *Polarity (pol.)*: 2PC objective function
- *Agreement ratio (a.r.)*: fraction of compliant edges in the subgraph induced by the detected communities.
- *F1-score* w.r.t. the ground-truth communities (synthetic networks)

1. Gullo Francesco, et al. ''Neural discovery of balance-aware polarized communities'' Machine Learning 2024
2. Jingbang Chen, et. al. '' Scalable Algorithm for Finding Balanced Subgraphs with Tolerance in Signed Networks'' KDD 2024
3. Bonchi Francesco, et al. ''Discovering polarized communities in signed networks'' CIKM 2019.
4. Mihai Cucuringu, et. al. '' SPONGE: A generalized eigenproblem for clustering signed networks.'' AISTATS 2019
5. Kai-Yang Chiang, et al. '' Scalable clustering of signed networks using balance normalized cut'' CIKM 2012
6. Yixuan He, et al. '' SSSNET: semi-supervised signed network clustering'' SDM 2022
7. Bruno Ordozgoiti, et al. ''Finding large balanced subgraphs in signed networks'' WWW 2020
8. Nikhil Bansal, et al. ''Correlation Clustering'' Machine Learning 2004

Results on Real-World Data

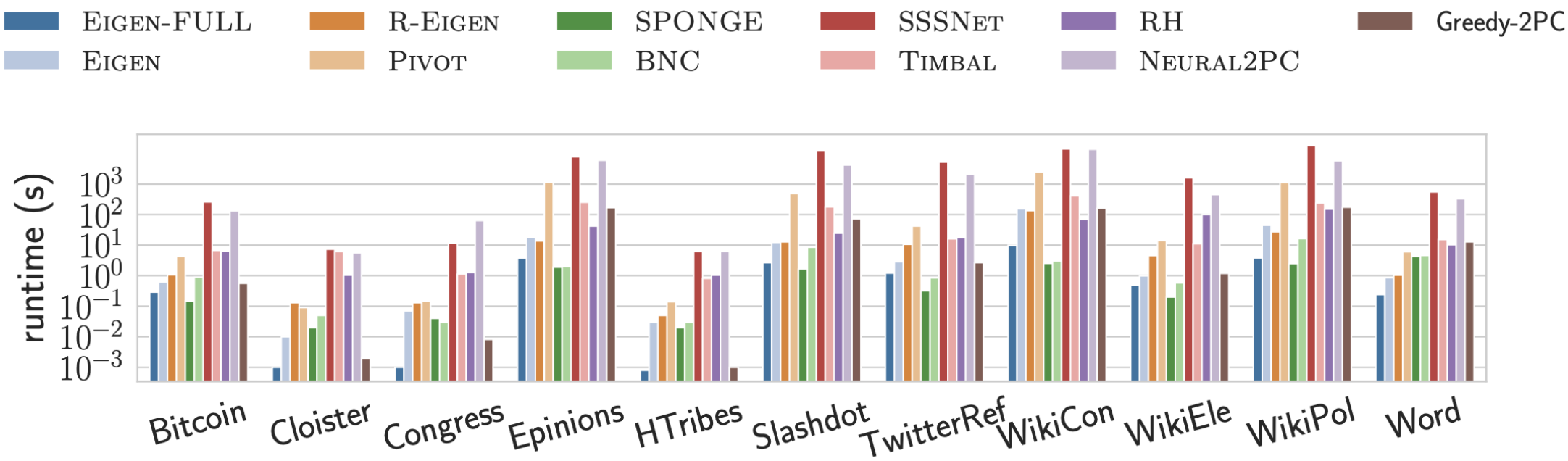
1. we report only the strongest competitors for effectiveness; full tables are in the paper.

method	criteria	*Bitcoin*	*Cloister*	*Congress*	*Epinions*	*HTribes*	*Slashdot*	*TwitterRef*	*WikiCon*	*WikiEle*	*WikiPol*	*Word*
EIGEN-FULL	pol.	6.23	6.11	4.37	8.89	5.50	8.08	39.03	28.78	19.58	8.57	9.87
	a.r.	0.93	0.72	0.96	0.91	0.88	0.83	0.92	0.91	0.85	0.91	0.76
EIGEN	pol.	29.52	**7.45**	6.58	128.72	**6.18**	79.7	174.1	175.65	71.73	88.44	24.02
	a.r.	0.95	0.94	0.98	0.95	1.00	0.99	0.99	0.95	0.93	0.96	0.98
RH	pol.	29.30	7.39	6.50	170.54	**6.18**	82.39	174.40	190.52	72.64	89.50	24.28
	a.r.	0.96	0.91	0.98	1.00	1.00	0.99	1.00	0.96	0.92	0.97	0.97
NEURAL2PC	pol.	30.28	**7.45**	6.64	171.1	**6.18**	82.25	174.35	187.29	72.17	88.89	24.32
	a.r.	0.95	0.94	0.98	1.00	1.00	0.99	0.99	0.95	0.92	0.96	0.97
Greedy-2PC	pol.	**30.57**	**7.45**	**6.70**	**171.17**	**6.18**	**82.80**	**174.66**	**196.73**	**72.79**	**90.07**	**25.03**
	a.r.	0.96	0.94	1.00	1.00	1.00	0.99	1.00	0.97	0.92	0.97	0.98

Table. Polarity («pol.») and edge-agreement ratio («a.r.») of the proposed method vs. competing methods[1] on real datasets. Best polarity results in bold, second-best underlined.

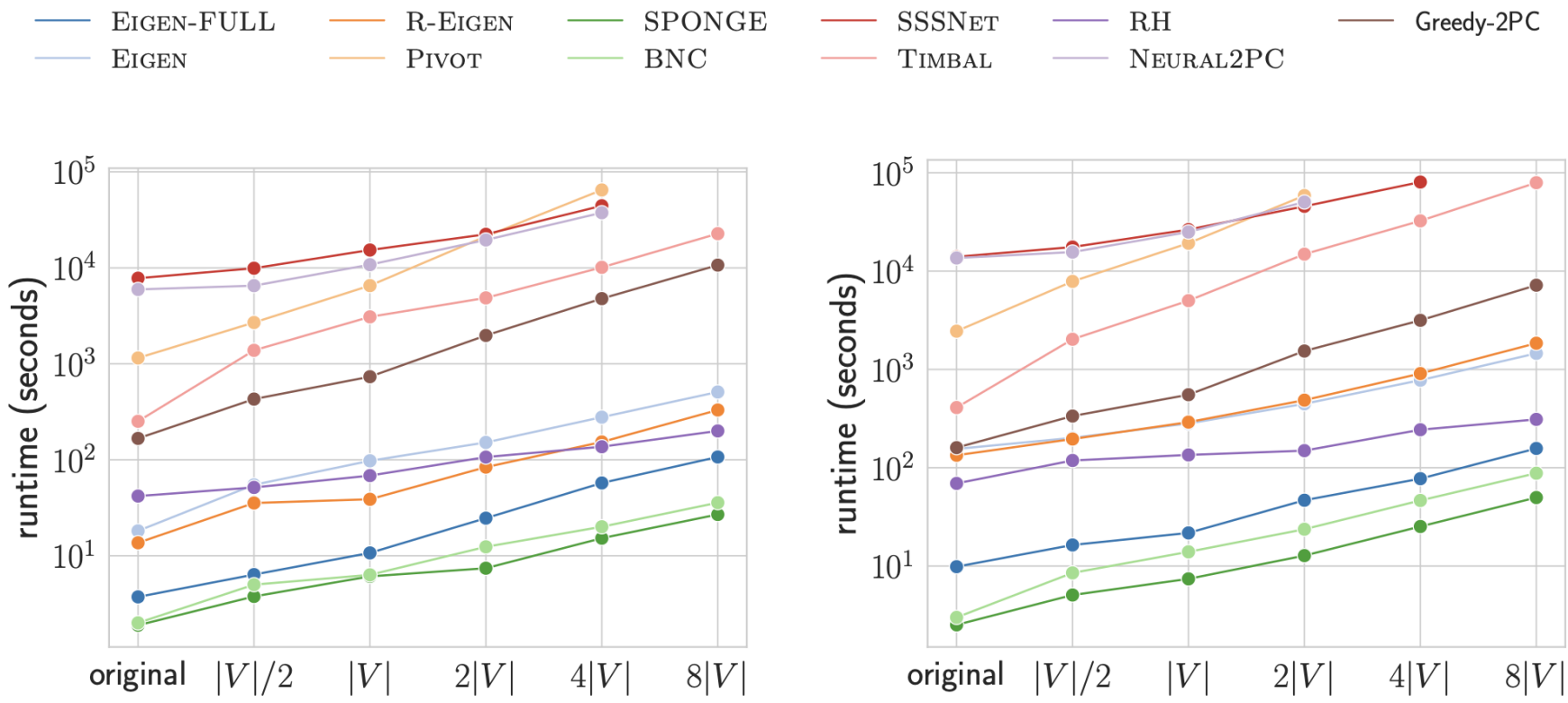
- Greedy-2PC achieves the best polarity: $+\sim 1\%$ vs Neural2PC on 5 datasets and $+\sim 5\%$ on WikiCon; $+\sim 4\%$ vs RH on Bitcoin and $+\sim 3\%$ on Cloister/WikiCon/Word.

Efficiency Results



- Greedy-2PC is $1-4\times$ faster than Neural2PC and beats RH on 6/11 datasets—Bitcoin, Cloister, Congress, HTribes, TwitterRef, WikiEle—while remaining comparable on larger networks.

Scalability Results



(a) Epinions

(b) WikiConflict

- On augmented Epinions/WikiCon, Greedy-2PC scales linearly and finishes in 7.13h in the worst case, while Pivot/SSSNet/Neural2PC do not complete within the 24h timeout on the largest instances.

Results on Synthetic Data

method	criteria	η 0	0.1	0.2	0.3	0.4	0.5	0.6
EIGEN-FULL	F_1	.333	.334	.333	.333	.334	.333	.264
	pol.	39.80	44.86	44.55	40.51	37.94	30.23	28.82
EIGEN	F_1	**1.0**	.998	.998	.998	**.995**	.972	.307
	pol.	**199**	168.04	140.31	110.5	**81.44**	50.02	35.52
RH	F_1	**1.0**	.99	**1.0**	**1.0**	**.995**	0.81	0.25
	pol.	**199**	167.89	**140.65**	**110.69**	**81.44**	45.83	35.76
NEURAL2PC	F_1	**1.0**	**1.0**	**1.0**	**1.0**	**.995**	**.997**	.341
	pol.	**199**	**168.62**	**140.65**	**110.69**	**81.44**	50.27	36.16
Greedy-2PC	F_1	**1.0**	**1.0**	**1.0**	**1.0**	**.995**	.988	.293
	pol.	**199**	**168.62**	**140.65**	**110.69**	**81.44**	**50.28**	**38.1**

Table. Greedy-2PC vs. baselines on synthetic datasets (averaged over 10 graphs per setting), varying noise η while keeping network and community size to 1000 and 100, respectively.

- On synthetic graphs with increasing noise (η), Greedy-2PC is robust and typically best (with Neural2PC) in both F1 and polarity

Conclusions & Future Work

Summary

- We established a link between the 2PC problem in signed networks and the DS problem.
- We proposed Greedy-2PC, a linear-time algorithm inspired by DS methods, and showed its superior effectiveness and efficiency on real-world and synthetic datasets compared to existing approaches.
- This work opens the door to applying the rich densest-subgraph literature to solve other challenging problems in signed network analysis.

Future Work

- Extend the approach to handle more than two communities, relax theoretical assumption, and explore applications in weighted and directed graphs.

Thank you!
Questions?