Discovering Balance-Aware Polarized Communities in Signed Networks with Graph Neural Networks

(Discussion Paper)

Francesco Gullo[1], Domenico Mandaglio[2] and Andrea Tagarelli[2]

[1]*Dept. Information Engineering, Computer Science, and Mathematics, University of L'Aquila, Coppito (AQ), Italy.*
[2]*Dept. Computer Engineering, Modeling, Electronics, and Systems Engineering, University of Calabria, Rende (CS), Italy.*

Abstract

Signed graphs model interactions among users, where nodes represent individuals and edges are labeled as positive for friendly relationships and negative for antagonistic ones. The 2-Polarized-Communities (2pc) problem aims to identify two disjoint polarized communities in a signed network so as to satisfy three conditions: the majority of intra-community edges should be positive, the majority of inter-community edges should be negative, and the ratio of edges satisfying these conditions to the number of nodes in the communities should be maximized. Existing 2pc methods suffer from two key limitations: (i) they rely on a single optimal solution to a continuous relaxation of the problem, later rounded to obtain the final pair of polarized communities, and (ii) the standard 2pc objective function does not impose any constraints on the balance between community sizes.

In this paper, we discuss a method that addresses both limitations and introduce two key contributions: (i) a Graph Neural Network-based approach that systematically explores multiple suboptimal solutions to the relaxed 2pc problem, selecting the one that yields the best 2pc solution after rounding; and (ii) a generalization of the 2pc objective function which explicitly encourages size-balanced communities. Extensive experiments on real-world and synthetic signed graphs have shown the high accuracy of our approach, its superiority over existing methods, and the effectiveness of γ-polarity in producing high-quality, well-balanced polarized communities.

Keywords

polarization, signed graphs, graph neural networks

1. Introduction

The widespread use of modern social media has created a huge amount of online social interactions, fostered the formation of communities [1, 2, 3, 4] and facilitating discussions about a variety of topics. Users establish *positive* relationships such as friendships, agreements, and trust, as well as *negative* relationships such as foes, disagreements, and distrusts. The existence of such mixed interactions has led to an ever-growing *polarization* phenomenon, i.e., a division of the set of users into groups with opposite view on controversial topics (e.g., politics, religion).

Signed graphs are graphs whose edges are assigned either a positive or a negative label, denoting whether the interaction depicted by an edge is friendly or antagonistic, respectively [5]. Signed graphs are used to model a variety of data and study numerous (social) phenomena, such as emergence of polarized discussions in social media, or analysis of trust/distrust in review platforms [6, 7, 8, 9, 10]. Bonchi *et al.* [11] defined the 2-Polarized-Communities (for short,

SEBD 2025: 33rd Symposium on Advanced Database Systems, June 16-19, 2025, Ischia, Italy
gullof@acm.org (F. Gullo); d.mandaglio@dimes.unical.it (D. Mandaglio); tagarelli@dimes.unical.it (A. Tagarelli)

2PC) problem on signed graphs, aiming to find two subsets of nodes, generally referred to as *communities*, such that there are (R1) mostly positive edges within each community and (R2) mostly negative edges between the two communities, and (R3) the subgraph induced by these two communities is as much dense as possible. Also, the two communities are required to be *non-overlapping*, but they *do not necessarily need to cover the entire node set.* The rationale of the latter is to comply the most with real-world situations, where polarized communities are concealed within a body of other graph nodes which do not (yet) have a strongly formed opinion, and, as such, they are *neutral* in terms of polarization.

Motivation. The above R1–R3 requirements for the 2PC problem are jointly pursued by maximizing a single objective function, termed *polarity*. Maximizing polarity is $\mathbf{NP}$-hard [11], but a continuous relaxation of that problem is solvable in polynomial time. Existing methods rely on rounding (i.e., discretizing) the optimal solution of the relaxed problem, but this approach has two key limitations. First, deriving a solution to 2PC starting from the optimal solution of the relaxed problem may be limiting in terms of polarity as *suboptimal* solutions to the relaxed problem can lead to *better solutions* to 2PC after rounding. Second, polarity maximization does not require or foster *size-balanced* communities, often leading to one dominant and one nearly empty group, even when naturally balanced polarized communities exist. Identifying such balanced groups is crucial across various domains, from social media and politics to market research, as it fosters constructive debates, reduces echo chambers, and identifies harmful situations. Thus, methods that can detect *balanced polarized communities* are needed.

Contributions. In this paper, we discuss our recent advancement [12] in the 2PC problem that properly addresses the above limitations. Our contributions are twofold. First, to address the first limitation and leveraging the recent success of Graph Neural Networks (GNNs) in graph learning tasks [13, 14, 15], we propose a novel GNN-based approach, dubbed Neural2PC, that systematically explores multiple suboptimal solutions to the relaxed problem, ultimately selecting the one that yields the best discrete 2PC solution after rounding. Second, to overcome the second limitation, we define a generalization of the polarity function, named γ-*polarity* that is designed to produce polarized communities that, depending on the setting of γ, can be either more balanced or larger than those yielded by standard polarity.

2. Preliminaries

Let $G = (V, E^+, E^-)$ be an *undirected signed graph*, where V is a set of nodes, and $E^+, E^- \subseteq V \times V$, $E^+ \cap E^- = \emptyset$, are sets of *positive* and *negative* edges, respectively. We assign each node in V a unique integer ID in $1, \ldots, |V|$ and use $u \in V$ interchangeably for the node and its position, simplifying matrix/vector notation. $\mathbf{A} \in \{-1, 0, 1\}^{|V| \times |V|}$ is the *signed adjacency matrix* of G, defined as $\mathbf{A}[u, v] = 1$ if $(u, v) \in E^+$, $\mathbf{A}[u, v] = -1$ if $(u, v) \in E^-$, and $\mathbf{A}[u, v] = 0$ otherwise.

Given a signed graph $G = (V, E^+, E^-)$, the 2-POLARIZED-COMMUNITIES [11] problem (for short, 2PC) finds two disjoint subsets $S_1, S_2 \subseteq V$ of nodes such that (R1) there are as many positive edges and as few negative edges as possible within S_1 and within S_2; (R2) there are as many negative edges and as few positive edges as possible across S_1 and S_2; and (R3) there should be a large number of edges complying with (R1) and (R2) within S_1 and S_2 relative to the total number of nodes in these groups.

S_1 and S_2 are regarded as *polarized communities*, i.e., groups of nodes which are cohesive in terms of both intra-group positive relationships (edges) and inter-group negative relationships. Nodes included into neither S_1 nor S_2, denoted as S_0, form the set of *neutral* nodes. A partition $\{S_0, S_1, S_2\}$ of V can alternatively be represented by a (column) vector $\mathbf{x} \in \{-1, 0, 1\}^{|V|}$, whose u-th coordinate is $\mathbf{x}_u = 0$ if $u \in S_0$, $\mathbf{x}_u = 1$ if $u \in S_1$, and $\mathbf{x}_u = -1$ if $u \in S_2$.

The above R1–R3 requirements are altogether encoded into a single function, termed *polarity*:

Definition 1 (Polarity [11]). *Given a vector* $\mathbf{x} \in \{-1, 0, 1\}^{|V|}$ *and a matrix* $\mathbf{A} \in \{-1, 0, 1\}^{|V|\times|V|}$*, the polarity* $p(\mathbf{x}, \mathbf{A})$ *of* $\mathbf{x}$ *with respect to* $\mathbf{A}$ *is defined as:*

$$p(\mathbf{x}, \mathbf{A}) = \frac{\mathbf{x}^\top \mathbf{A}\, \mathbf{x}}{\mathbf{x}^\top \mathbf{x}}. \tag{1}$$

The numerator of $p(\cdot, \cdot)$ accounts for R1 and R2, while numerator and denominator altogether model R3. In this regard, note that $\mathbf{x}^\top \mathbf{x} = |S_1 \cup S_2|$. The 2PC problem is formulated as follows:

Problem 1 (2PC [11]). *Given a signed graph* $G = (V, E^+, E^-)$ *with signed adjacency matrix* $\mathbf{A}$*, find*

$$\mathbf{x}^* = \underset{\mathbf{x}\in\{-1,0,1\}^{|V|}}{\arg\max}\; p(\mathbf{x}, \mathbf{A}).$$

Relaxing node-to-community assignments to be in $[-1, 1]$ results in the following problem:

Problem 2 (2PC-RELAXED [11]). *Given a signed graph* $G = (V, E^+, E^-)$ *with signed adjacency matrix* $\mathbf{A}$*, find*

$$\mathbf{z}^* = \underset{\mathbf{z}\in[-1,1]^{|V|}}{\arg\max}\; p(\mathbf{z}, \mathbf{A}),$$

where polarity $p(\mathbf{z}, \mathbf{A}) = \mathbf{z}^\top \mathbf{A}\, \mathbf{z} / \mathbf{z}^\top \mathbf{z}$ *of a vector* $\mathbf{z} \in [-1, 1]^{|V|}$ *is defined as in Definition 1.*

State of the art in 2PC. 2PC is shown to be $\mathbf{NP}$-hard, while 2PC-RELAXED can be solved in polynomial time by finding the eigenvector of the signed adjacency matrix corresponding to the largest eigenvalue [11]. Bonchi *et al.* [11] exploit the latter to devise two approximation algorithms for 2PC. The first (deterministic) algorithm simply rounds the optimal solution $\mathbf{z}^*$ to 2PC-RELAXED as $\mathbf{x}^*_u = \mathrm{sgn}(\mathbf{z}^*_u)$, for all $u \in V$, where $\mathrm{sgn}(\cdot)$ is the sign function. The second (randomized) algorithm sets, for all $u \in V$, $\mathbf{x}^*_u = \mathrm{sgn}(\mathbf{z}^*_u)$ if a Bernoulli experiment with success probability $|\mathbf{z}^*_u|$ succeeds, otherwise $\mathbf{x}^*_u = 0$.

3. Related Work

Representation learning for signed graphs. *Graph representation learning* is the problem of assigning elements of a graph (e.g., nodes, edges, subgraphs) to numerical vectors (*embeddings*) such that the similarity between those elements in the graph is preserved in the embedding space. This field spans shallow methods, which optimize specific criteria (e.g., d-hop reachability, random-walk co-occurrence) and deep approaches based on *graph neural networks* (GNNs) [16, 14]. Representation learning has been studied for signed graphs as well, both undirected [17, 18, 19, 20, 21] and directed [22, 23]. In this work, we regard signed graph representation learning as a building block of the proposed framework. Note that our approach is versatile w.r.t. the choice of graph representation learning model.

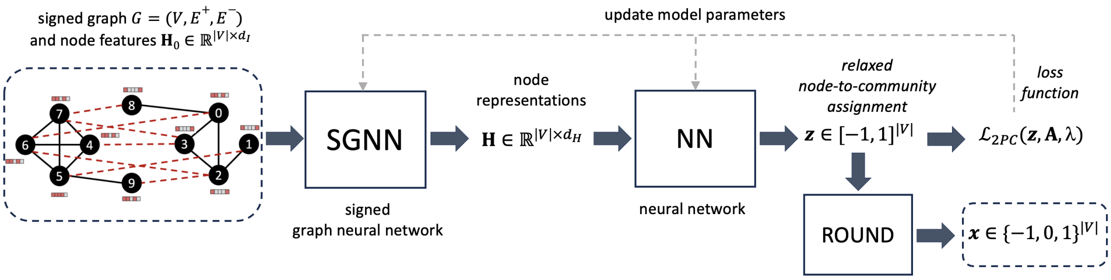

Figure 1: Overview of the Neural2PC approach [12].

Clustering signed graphs has also received attention in the literature [24, 25, 26, 27, 28]. However, those methods require every node to be part of an output cluster, hence they are not designed to detect neutral nodes and left them out of evaluation, unlike our approach. Also, signed graph clustering methods optimize criteria other than polarity.

4. The Neural2PC approach

Overview. Unlike existing methods [11] which find the optimal solution $\mathbf{z}^*$ to 2PC-RELAXED (Problem 2) directly, we let a *neural-network* model f_θ – with parameters θ – produce a set $\{\mathbf{z}_e \mid e = 1, \ldots, e_{max}\}$ of feasible solutions to 2PC-RELAXED during multiple epochs $1, \ldots, e_{max}$ of training. All the various $\mathbf{z}_e$ are rounded in order to yield feasible discrete solutions $\mathbf{x}_e$ to 2PC. The best (in terms of polarity, Definition 1) of such $\mathbf{x}_e$ solutions is the definitive output.

The rationale of our approach is that it allows for exploring a variety of suboptimal solutions to 2PC-RELAXED. This favors obtaining ultimate discrete solutions (after rounding) which exhibit higher polarity than the one derived by rounding the optimal solution to 2PC-RELAXED. The goal is to find the model parameters θ that maximize the polarity of the (relaxed) solutions computed via f_θ (or, equivalently, minimize a loss defined based on the negative polarity). As parameter learning goes on, it is expected to get a deeper exploration of the space of relaxed solutions, and hence a higher likelihood of getting an effective discrete solution after rounding.

The proposed neural approach is named Neural2PC. A graphical illustration of its main components is shown in Figure 1. Next, we delve into its technical details.

Neural model. Our f_θ model takes as input a signed graph $G = (V, E^+, E^-)$, and a matrix $\mathbf{H}_0 \in \mathbb{R}^{|V| \times d_I}$ containing a d_I-dimensional (real-valued) vector of features for every node. If such features are not available, $\mathbf{H}_0$ can be initialized by considering structural information derived from G [17]. The first block of f_θ is a (m-layer) signed GNN [17, 18, 19, 20, 21] SGNN$(\cdot)$, with parameters θ_{SGNN}. SGNN$(\cdot)$ properly processes G's topology and (possibly) node features $\mathbf{H}_0$, and outputs a matrix $\mathbf{H} \in \mathbb{R}^{|V| \times d_H} = [\mathbf{h}_u \in \mathbb{R}^{d_H}]_{u \in V}$ containing a hidden vector representation $\mathbf{h}_u$ of every node $u \in V$. This operation can be described as $\mathbf{H} = \text{SGNN}(G, \mathbf{H}_0)$. Then, vector representations produced by SGNN$(\cdot)$ feed into fully-connected neural-network linear layers NN$(\cdot)$, with parameters θ_{NN}. Ultimately, a $\tanh$ activation function is used to cast the (node-to-community assignment) scores for every node to the desired $[-1, 1]$ range (cf. Problem 2). This operation can be written as $\mathbf{z} = \tanh(\text{NN}(\mathbf{H}))$. As a result, the overall f_θ

model, with parameters $\theta = \{\theta_{\text{SGNN}}, \theta_{\text{NN}}\}$, is as follows:

$$f_\theta(G, \mathbf{H}_0) = \tanh(\textsc{nn}(\textsc{sgnn}(G, \mathbf{H}_0))), \quad (2)$$

Loss function. To optimize model parameters θ, we employ a loss function $\mathcal{L}_{2\text{PC}}$ defined as a combination of (the negative of) polarity $p(\cdot,\cdot)$ and a regularization term. The latter enforces the model produce continuous scores that are as close as possible to the ultimately desired discrete $\{-1, 0, 1\}$ scores. Specifically, we define the regularization term as the $||\cdot||_2$ L2-norm of a vector $\rho \in \mathbb{R}^{|V|}$, whose entries $\rho[u]$, for all $u \in V$, are set to the difference $\min\{|\mathbf{z}[u]|, 1 - |\mathbf{z}[u]|\}$ between $\mathbf{z}[u]$ and the closest valid discrete score. The intuition is that minimizing the norm of ρ (together with the other loss component) is expected to produce the desired effect of yielding output continuous $\mathbf{z}$ scores not too far from the valid discrete ones.

Given $\mathbf{z} = f_\theta(G, \mathbf{H}_0)$, the signed adjacency matrix $\mathbf{A}$ of G, and a hyperparameter $\lambda \in \mathbb{R}$ which weighs the importance of the regularization term, the $\mathcal{L}_{2\text{PC}}$ loss function is defined as:

$$\mathcal{L}_{2\text{PC}}(\mathbf{z}, \mathbf{A}, \lambda) \; = \; -p(\mathbf{z}, \mathbf{A}) + \lambda||\rho||_2^2 \quad (3)$$

Rounding. To round a continuous solution $\mathbf{z} \in [-1, 1]^{|V|}$ onto a valid discrete $\mathbf{x} \in \{-1, 0, 1\}^{|V|}$ solution to 2PC, we borrow the procedure adopted by Bonchi *et al.* [11]. Specifically, given a threshold $\tau \in [0, 1]$, for all $u \in V$, $\mathbf{x}[u] = \text{sgn}(\mathbf{z}[u])$ if $|z[u]| \geq \tau$, $\mathbf{x}[u] = 0$ otherwise. Let $Z_i = \{\lceil \mathbf{z}[u] \rfloor_i \mid u \in V\}$ be a set of candidate thresholds, where $\lceil \cdot \rfloor_i$ denotes approximating a real number at the i-th decimal digit (we use $i = 3$). In order to avoid sticking to a single τ, we follow [11], and try all the thresholds $\tau \in Z_i$. Given a threshold $\tau \in Z_i$, we yield a discrete solution $\mathbf{x}_\tau$ as follows: $\forall u \in V,\ \mathbf{x}_\tau[u] = \text{sgn}(\mathbf{z}[u])$ if $|\mathbf{z}[u]| \geq \tau$; 0 otherwise. The final solution corresponds to the discrete solution with highest polarity:

$$\textsc{round}(\mathbf{z}) \; = \; \underset{\mathbf{x} \in \{\mathbf{x}_\tau \,|\, \tau \in Z_i\}}{\arg\max} \; p(\mathbf{x}, \mathbf{A}). \quad (4)$$

Algorithm. The algorithm we employ to produce a solution to 2PC simply consists in optimizing the $\theta = \{\theta_{\text{SGNN}}, \theta_{\text{NN}}\}$ parameters of the f_θ neural model end-to-end, via standard gradient descent, for a number e_{max} of training epochs. Specifically, the algorithm alternates a forward phase, which produces a continuous solution $\mathbf{z}$ given the current θ parameters, and a backward phase, where parameters θ are updated via gradient descent, using the $\mathcal{L}_{2\text{PC}}$ loss function, with a certain learning rate α. The continuous solution $\mathbf{z}$ yielded in every epoch is rounded according to the ROUND$(\cdot)$ procedure described above. The discrete rounded solution with the highest polarity score out of all the ones produced in the various epochs is ultimately output. Rounding and evaluating polarity in every epoch is necessary because the best discrete solution's epoch is hard to predict in advance.

5. Balancing the size of the communities

A known issue with the polarity measure (Definition 1) is its bias toward size-imbalanced communities, sometimes leading to one dominant community and the other empty [11]. To address this, we propose γ-*polarity*, a generalized measure that promotes more balanced polarized communities by adjusting the parameter γ. We define γ-polarity by modifying the denominator of the polarity measure while keeping the numerator unchanged. Given a node-to-community assignment vector $\mathbf{x} \in \{-1, 0, 1\}^{|V|}$, let $s_1 = \sum_{u \in V, \mathbf{x}[u]<0} |\mathbf{x}[u]|$ and $s_2 = \sum_{u \in V, \mathbf{x}[u]>0} \mathbf{x}[u]$ be the size of the two communities, with $s_{max} = \max\{s_1, s_2\}$, $s_{min} = \min\{s_1, s_2\}$. The denominator of the polarity measure is $\mathbf{x}^\top \mathbf{x} = s_{max} + s_{min}$, which can be rewritten as $(s_{max} - s_{min}) + 2s_{min}$. The key idea behind γ-polarity is to weight the size imbalance term $(s_{max} - s_{min})$ by a factor $\gamma > 0$, leading to the following definition:

Definition 2 (γ-polarity). *Given a vector $\mathbf{x} \in \{-1, 0, 1\}^{|V|}$, a matrix $\mathbf{A} \in \{-1, 0, 1\}^{|V| \times |V|}$, and a real number $\gamma > 0$, the γ-polarity $p_\gamma(\mathbf{x}, \mathbf{A})$ of $\mathbf{x}$ with respect to $\mathbf{A}$ is defined as:*

$$p_\gamma(\mathbf{x}, \mathbf{A}) = \frac{\mathbf{x}^\top \mathbf{A}\, \mathbf{x}}{(s_{max} - s_{min})\gamma + 2\, s_{min}}. \tag{5}$$

For $\gamma > 1$, the size-difference $(s_{max} - s_{min})$ term is amplified: maximizing p_γ enforces such a term to be small, encouraging balanced communities. For $\gamma \in (0, 1)$, the effect is reversed, while $\gamma = 1$ recovers standard polarity.

The relaxed version of γ-polarity replaces $\mathbf{x}$ with a continuous vector $\mathbf{z} \in [-1, 1]^{|V|}$ in Equation (5). It can be integrated into Neural2PC by substituting $p(\mathbf{z}, \mathbf{A})$ with $p_\gamma(\mathbf{z}, \mathbf{A})$ in the $\mathcal{L}_{2\text{PC}}$ loss (Equation (3)).

6. Experimental Methodology

Evaluation goals. We evaluated Neural2PC and competitors/baselines on (1) real datasets, and (2) synthetic datasets; (3) impact of different signed GNNs in Neural2PC; (4) runtimes of the considered methods; (5) an ablation study on the Neural2PC components; (6) effectiveness of the γ-polarity measure in yielding communities that are both size-balanced and high-quality.

Real datasets. We selected publicly-available real-world signed graphs of varying sizes and types. *Bitcoin* [29] (5.9k nodes, 21.5k edges), *Epinions* [29] (131.6k nodes, 711.2k edges) are trust-distrust networks. *Cloister* [30] (18 nodes, 125 edges), *Congress* [30] (219 nodes, 521 edges), and *HTribes* [30] (16 nodes, 58 edges) are social networks. Larger networks include *Slashdot* [29] (82.1k nodes, 500.5k edges), a friend-foe network, *TwitterRef* [31] (10.9k nodes, 251.4k edges), a stance network, *WikiCon* [29] (116.7k nodes, 2.03M edges), an edit-conflict network, *WikiEle* [30] (7.1k nodes, 100.7k edges), a voting network, and *WikiPol* [31] (138.6k nodes, 715.9k edges), a political discussion network.

Synthetic datasets. We used synthetic signed graphs to test methods in recovering ground-truth polarized communities, generated by the *modified signed stochastic block model* (m-SSBM) [32]. The model has three parameters: the total number of nodes n, the size $n_c = |S_1| = |S_2|$ of the polarized communities, and a noise parameter $\eta \in [0, 1]$, with polarized communities emerging when $\eta \leq 2/3$.

We used different synthetic graphs by varying number of nodes (n), community size n_c, and $\eta \in \{0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6\}$. For each configuration, we generated 10 different graphs.

Competing methods. We compared our Neural2PC against state-of-the-art methods for polarized community detection and relevant baselines from related problems. Our primary

Table 1
Polarity (“pol.”, Def. 1) and solution size ($|S_1|; |S_2|$) of the proposed Neural2PC vs. competing methods on real datasets [12]. Best polarity results in bold, second-best underlined.

method	criteria	*Bitcoin*	*Cloister*	*Congress*	*Epinions*	*HTribes*	*Slashdot*	*TwitterRef*	*WikiCon*	*WikiEle*	*WikiPol*
Eigen	pol.	29.52	**7.45**	6.58	128.72	**6.18**	79.7	174.1	175.65	71.73	88.44
	size	136;2	8;3	28;24	999;18	7;4	233;1	669;4	1993;449	745;3	646;2
R-Eigen	pol.	14.12	6.23	5.38	71.36	5.82	29.21	118.81	99.93	55.91	35.72
	size	725;103	14;3	50;50	4057;407	7;4	2827;125	1487;20	9778;3109	1054;67	6838;579
Pivot	pol.	21.65	4.17	3.1	156.38	3.5	61.0	116.25	129.33	37.59	46.52
	size	21;19	9;3	6;5	248;5	5;3	283;6	1142;16	368;134	407;7	598;11
Greedy	pol.	29.01	6.11	5.77	170.3	5.5	**82.72**	173.94	127.96	**72.67**	**90.02**
	size	140;0	15;3	36;33	269;0	12;4	200;0	685;0	1151;0	730;0	543;0
SPONGE	pol.	8.36	6.11	4.43	7.12	5.5	6.36	28.03	-8.92	15.79	7.79
	size	26;2	15;3	115;104	131578;2	12;4	82138;2	8274;2610	116712;5	7113;2	138585;2
BNC	pol.	5.27	1.0	2.75	7.12	-0.5	6.36	41.49	8.92	15.79	7.79
	size	5834;47	17;1	216;3	131225;355	10;6	82138;2	10882;2	115730;987	7102;13	138556;31
SSSNet	pol.	9.06	6.93	4.43	73.19	5.0	7.26	41.49	28.56	17.09	7.82
	size	713;8	6;3	115;104	953;0	11;5	72915;9225	10864;20	50835;5401	6402;713	132566;6021
Neural2PC	pol.	**30.28**	**7.45**	**6.64**	**171.1**	**6.18**	82.25	**174.35**	**187.29**	72.17	88.89
	size	158;32	8;3	29;24	268;1	7;4	203;0	677;4	1788;559	742;2	618;2

competitors are Eigen and its randomized variant R-Eigen [11], as they address the same 2pc optimization problem.

We included Pivot, a baseline inspired by a correlation clustering algorithm [33, 34, 35] and Greedy [11], a method based on a 2-approximation algorithm for densest subgraph [36]. We also considered the signed graph clustering algorithms BNC [24], SPONGE [25], and SSSNet [26].

Experimental setting. We used SGDNET [18], SNEA [19] and SGNN [17] as GNN models for our Neural2PC. All signed GNN models were trained on CPUs with uniform settings: node embeddings size $d_H = 64$, $m = 2$ layers, the final embedding of a signed spectral embedding model [27] as the input feature matrix ($d_I = 64$) and default values for other parameters. Model training was carried out with the Adam optimizer for $e_{max} = 300$ epochs, using grid search for the learning rate $\alpha \in \{0.01, 0.005, 0.001\}$ and regularization factor $\lambda \in \{0.1, 0.01, 0.001\}$. Results are averaged over 30 runs.

7. Results

Results on real datasets. Table 1 reports the polarity values, along with the sizes of the two resulting polarized communities. Concerning Neural2PC, we only report the results obtained by the best-performing (in terms of polarity) graph representation learning method.

Neural2PC is generally the most competitive in polarity. The exceptions (Slashdot, WikiEle, WikiPol) occur when Greedy selects an overly dense subgraph as one polarized community, leaving the other empty—an undesirable outcome. In contrast, our method returns both non-empty communities with high polarity in WikiEle and WikiPol. Among competitors, Eigen and R-Eigen, achieve strong polarity, with Eigen outperforming R-Eigen. Pivot as well as BNC and SPONGE performs poorly, with the latter two often producing very imbalanced communities. SSSNet performs slightly better but still lags behind.

Results on synthetic datasets. We analyzed the average F_1-scores and polarity scores over 10 synthetic graphs for each noise level η, considering varying network sizes ($n \in 250, 500, 1000, 2000$) and community sizes ($n_c \in 25, 50, 100, 200$). Our experiments (results not shown) revealed that Neural2PC remains robust to increasing noise, consistently outperforming competitors in both F_1-score and polarity.

Impact of different signed GNNs. We analyzed polarity and community size values yielded by Neural2PC using different signed GNN models (results not shown). Our experiments revealed that the polarity of the solutions provided by Neural2PC does not significantly change across the various GNNs, which indicates robustness of the approach in terms of this main component.

Execution times. The average runtime performance of the methods was measured across the different runs (results not shown). The learning-based methods, SSSNet and Neural2PC, have the highest runtimes, primarily due to the number of training epochs ($e_{\max} = 300$). Nonetheless, Neural2PC 's per-epoch time is comparable to the fastest methods. Among the other methods, SPONGE performs best, followed by BNC and Eigen. R-Eigen is slightly slower than Eigen due to its randomized nature, while Pivot and Greedy are inefficient.

Ablation study. To assess the impact of Neural2PC components, we conducted an ablation study on two simplified versions of Neural2PC: (i) NN, which removes the sgnn$(\cdot)$ block, retaining only nn$(\cdot)$, and (ii) Direct, which optimizes the $\mathbf{z}$ assignments by minimizing the $\mathcal{L}_{2\text{PC}}$ loss via projected gradient descent. For each variant, we measured (results not shown) polarity, solution size, and execution time. The full Neural2PC is crucial for optimal polarity across all datasets or at least matching Direct (TwitterRef, WikiEle). However, Direct is less efficient, requiring significantly more epochs (at least twice as many) than Neural2PC (and NN, too), as the latter leverages sgnn$(\cdot)$ to assigns more similar scores within communities, reducing the number of thresholds tested in rounding and improving efficiency. Also, Neural2PC and Direct yield larger communities than NN. Overall, the outcomes of this ablation study justify the need for all components of the Neural2PC framework.

γ**-polarity results.** We analyzed the impact of γ on the size and quality of solutions found by Neural2PC using the γ-polarity loss. We tested multiple γ values above 1 (up to 20) and their reciprocals to explore a symmetric range below 1. Neural2PC consistently achieves (results not shown) the best γ-polarity. Higher γ leads to more balanced communities, while lower γ creates imbalance, sometimes leaving one community empty. Competing methods often yield highly unbalanced solutions. Overall, γ-polarity proves useful, allowing users to inspect and select the most suitable communities for their needs.

8. Conclusion

We discussed a recent advancement in 2pc [12], which relies on a GNN-based neural approach and introduces the notion of γ-polarity to improve the balance in the size of the polarized communities. Future work includes detecting more than two communities [32], leveraging clustering ensemble techniques [37, 38] and improving the training efficiency [39].

Acknowledgements: D. Mandaglio and A. Tagarelli are partly supported by the PNRR Future AI Research (FAIR) project (H23C22000860006, M4C21.3 spoke 9).

References

[1] D. Mandaglio, A. Tagarelli, F. Gullo, In and out: Optimizing overall interaction in probabilistic graphs under clustering constraints, in: Proc. KDD Conf., 2020, pp. 1371–1381. doi:10.1145/3394486.3403190.

[2] A. K. Ghoshal, N. Das, S. Das, Disjoint and overlapping community detection in small-world networks leveraging mean path length, TCSS 9 (2021) 406–418.

[3] D. Mandaglio, A. Tagarelli, Dynamic consensus community detection and combinatorial multi-armed bandit, in: Proc. ASONAM Conf., 2019, pp. 184–187. doi:10.1145/3341161.3342910.

[4] D. Mandaglio, A. Tagarelli, A combinatorial multi-armed bandit based method for dynamic consensus community detection in temporal networks, in: Proc. DS Conf., 2019, pp. 412–427. doi:10.1007/978-3-030-33778-0_31.

[5] F. Harary, On the notion of balance of a signed graph, Michigan Mathematical Journal 2 (1953) 143–146.

[6] G. Beigi, J. Tang, H. Liu, Signed link analysis in social media networks, in: Proc. ICWSM Conf., 2016, pp. 539–542.

[7] T. Derr, C. C. Aggarwal, J. Tang, Signed network modeling based on structural balance theory, in: Proc. CIKM Conf., 2018, pp. 557–566.

[8] K. Yao, L. Chang, L. Qin, Computing maximum structural balanced cliques in signed graphs, in: Proc. ICDE Conf., 2022, pp. 1004–1016.

[9] X. Zheng, D. D. Zeng, F. Wang, Social balance in signed networks, Inf. Syst. Frontiers 17 (2015) 1077–1095.

[10] L. L. Cava, D. Mandaglio, A. Tagarelli, Polarization in decentralized online social networks, in: Proc. WebSci Conf., 2024. doi:10.1145/3614419.3644013.

[11] F. Bonchi, E. Galimberti, A. Gionis, B. Ordozgoiti, G. Ruffo, Discovering polarized communities in signed networks, in: Proc. CIKM Conf., 2019, pp. 961–970.

[12] F. Gullo, D. Mandaglio, A. Tagarelli, Neural discovery of balance-aware polarized communities, Mach. Learn. 113 (2024) 6611–6644. doi:10.1007/S10994-024-06581-4.

[13] A. Tsitsulin, J. Palowitch, B. Perozzi, E. Müller, Graph clustering with graph neural networks, JMLR 24 (2023) 1–21.

[14] L. Wu, P. Cui, J. Pei, L. Zhao, Graph Neural Networks: Foundations, Frontiers, and Applications, Springer Singapore, Singapore, 2022.

[15] L. Zangari, D. Mandaglio, A. Tagarelli, Link prediction on multilayer networks through learning of within-layer and across-layer node-pair structural features and node embedding similarity, in: Proc. WWW Conf., 2024, pp. 924–935. doi:10.1145/3589334.3645646.

[16] H. Cai, V. W. Zheng, K. C. Chang, A comprehensive survey of graph embedding: Problems, techniques, and applications, TKDE 30 (2018) 1616–1637.

[17] T. Derr, Y. Ma, J. Tang, Signed graph convolutional networks, in: Proc. ICDM Conf., 2018, pp. 929–934.

[18] H. Liu, Z. Zhang, P. Cui, Y. Zhang, Q. Cui, J. Liu, W. Zhu, Signed graph neural network with latent groups, in: Proc. KDD Conf., 2021, pp. 1066–1075.

[19] Y. Li, Y. Tian, J. Zhang, Y. Chang, Learning signed network embedding via graph attention, in: Proc. AAAI Conf., 2020, pp. 4772–4779.

[20] S. Wang, J. Tang, C. Aggarwal, Y. Chang, H. Liu, Signed network embedding in social media, in: Proc. SDM Conf., 2017, pp. 327–335.
[21] J. Kim, H. Park, J.-E. Lee, U. Kang, Side: representation learning in signed directed networks, in: Proc. WWW Conf., 2018, pp. 509–518.
[22] J. Huang, H. Shen, L. Hou, X. Cheng, SDGNN: learning node representation for signed directed networks, in: Proc. AAAI Conf., 2021, pp. 196–203.
[23] J. Huang, H. Shen, L. Hou, X. Cheng, Signed graph attention networks, in: Proc. ICANN Work., 2019, pp. 566–577.
[24] K.-Y. Chiang, J. J. Whang, I. S. Dhillon, Scalable clustering of signed networks using balance normalized cut, in: Proc. CIKM Conf., 2012, pp. 615–624.
[25] M. Cucuringu, P. Davies, A. Glielmo, H. Tyagi, SPONGE: A generalized eigenproblem for clustering signed networks, in: Proc. AISTATS Conf., 2019, pp. 1088–1098.
[26] Y. He, G. Reinert, S. Wang, M. Cucuringu, SSSNET: semi-supervised signed network clustering, in: Proc. SDM Conf., 2022, pp. 244–252.
[27] J. Kunegis, S. Schmidt, A. Lommatzsch, J. Lerner, E. W. D. Luca, S. Albayrak, Spectral analysis of signed graphs for clustering, prediction and visualization, in: Proc. SDM Conf., 2010, pp. 559–570.
[28] P. Mercado, F. Tudisco, M. Hein, Clustering signed networks with the geometric mean of laplacians, in: Proc. NIPS Conf., 2016, pp. 4421–4429.
[29] J. Leskovec, A. Krevl, SNAP Datasets: Stanford large network dataset collection, http://snap.stanford.edu/data, 2014.
[30] J. Kunegis, KONECT – The Koblenz Network Collection, in: Proc. WWW Conf., 2013, pp. 1343–1350. http://konect.cc.
[31] M. Lai, V. Patti, G. Ruffo, P. Rosso, Stance evolution and twitter interactions in an Italian political debate, in: Proc. NLDB Conf., 2018, pp. 15–27.
[32] R.-C. Tzeng, B. Ordozgoiti, A. Gionis, Discovering conflicting groups in signed networks, Proc. NIPS Conf. 33 (2020) 10974–10985.
[33] N. Bansal, A. Blum, S. Chawla, Correlation clustering, Machine Learning 56 (2004) 89–113.
[34] D. Mandaglio, A. Tagarelli, F. Gullo, Correlation clustering with global weight bounds, in: Proc. ECML PKDD Conf., volume 12976, 2021, pp. 499–515. doi:10.1007/978-3-030-86520-7_31.
[35] F. Gullo, D. Mandaglio, A. Tagarelli, A combinatorial multi-armed bandit approach to correlation clustering, DAMI 37 (2023) 1630–1691. doi:10.1007/S10618-023-00937-5.
[36] M. Charikar, Greedy approximation algorithms for finding dense components in a graph, in: Proc. APPROX Work., 2003, pp. 84–95.
[37] F. Gullo, A. Tagarelli, S. Greco, Diversity-based weighting schemes for clustering ensembles, in: Proc. SDM Conf., SIAM, 2009, pp. 437–448. doi:10.1137/1.9781611972795.38.
[38] F. Gullo, C. Domeniconi, A. Tagarelli, Projective clustering ensembles, DAMI 26 (2013) 452–511. doi:10.1007/S10618-012-0266-X.
[39] F. Scala, S. Flesca, L. Pontieri, Play it straight: An intelligent data pruning technique for green-ai, in: Proc. DS Conf., volume 15243, 2024, pp. 69–85. doi:10.1007/978-3-031-78977-9_5.