SSSNET: Semi-Supervised Signed Network Clustering

Yixuan He [1] Gesine Reinert [1,2]
Songchao Wang [3] Mihai Cucuringu [1,2]

[1]University of Oxford, UK
[2]The Alan Turing Institute, UK
[3]University of Science and Technology of China, China

Signed network examples:

- Users may express trust-distrust or friendship-enmity.
- Review websites as well as online news allow users to approve or denounce others.
- Correlation networks, with the empirical correlation matrix between time series, for example of different stock returns, being construed as a weighted signed network, for example with stocks as nodes.

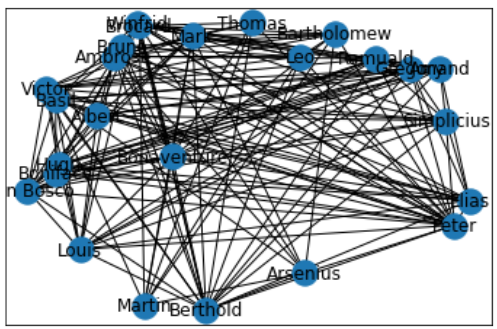


Figure: Sampson monastery data [Sampson, 1968].

Signed Clustering

Node clustering in signed networks can help reveal community structures in networks with positive and negative links, such as social networks and correlation networks.

Node embeddings could in principle help with clustering but most competitive state-of-the-art methods which generate node embeddings for signed networks focus on link sign prediction.

Those that pertain to node clustering are usually not graph neural network (GNN) methods.

GNN methods have proved powerful in other settings.

Here, we introduce a GNN framework, called SSSNET, with a *Signed Mixed-Path Aggregation* scheme, to obtain node embeddings for signed clustering.

Problem Definition

- $\mathcal{V}$: a set of n nodes
- adjacency matrix $\mathbf{A} = (A_{ij})_{i,j \in \mathcal{V}}$
- $\mathbf{X}_{\mathcal{V}} \in \mathbb{R}^{n \times d_{\text{in}}}$: node feature matrix
- $\mathbf{A}$ can be decomposed into positive and negative parts $\mathbf{A}^+$ and $\mathbf{A}^-$: $\mathbf{A}^+_{ij} = \max(\mathbf{A}_{ij}, 0)$ and $\mathbf{A}^-_{ij} = -\min(\mathbf{A}_{ij}, 0)$
- A **clustering** into K clusters: a partition of the node set into disjoint sets $\mathcal{V} = \mathcal{C}_0 \cup \mathcal{C}_1 \cup \cdots \cup \mathcal{C}_{K-1}$
- **Semi-supervised**: seed nodes for each cluster

- The standard heuristic for justifying the criteria for the embeddings hinges on the assumption that "an enemy's enemy is a friend".
- This heuristic is based on social balance theory (see for example [Sharma et al., 2021]), which asserts that in a social network, in a typical triangle either all three nodes are friends, or two friends have a common enemy; otherwise it would be viewed as *unbalanced*.
- More generally, all cycles are assumed to prefer to contain either zero or an even number of negative edges.

- Social balance theory is difficult to justify for general signed networks.
- However: the relationship between trust and distrust may not be a simple negation; the enemies of enemies are not necessarily friends;
- Example: the social network of relations between 16 tribes of the Eastern Central Highlands of New Guinea [Read, 1954].
- The main novelty of our approach is a new take on the role of social balance theory for signed network embedding, taking a **neutral** stance on whether or not the enemy of an enemy is a friend.

Path-Based Node Relationship

- For a target node v_j to be an h-hop **"friend"** neighbor of source node v_i *along a given path* from v_i to v_j of length h, all edges on this path need to be positive.
- For a target node v_j to be an h-hop **"enemy"** neighbor of source node v_i along a given path from v_i to v_j of length h, exactly one edge on this path has to be negative.
- Otherwise, v_i and v_j are **neutral** to each other on this path.
- A node can simultaneously be both a "friend" and an "enemy" to a source node.
- Aggregate these relationships by assigning different weights to different paths connecting two nodes.

Path-Based Node Relationship (cont.)

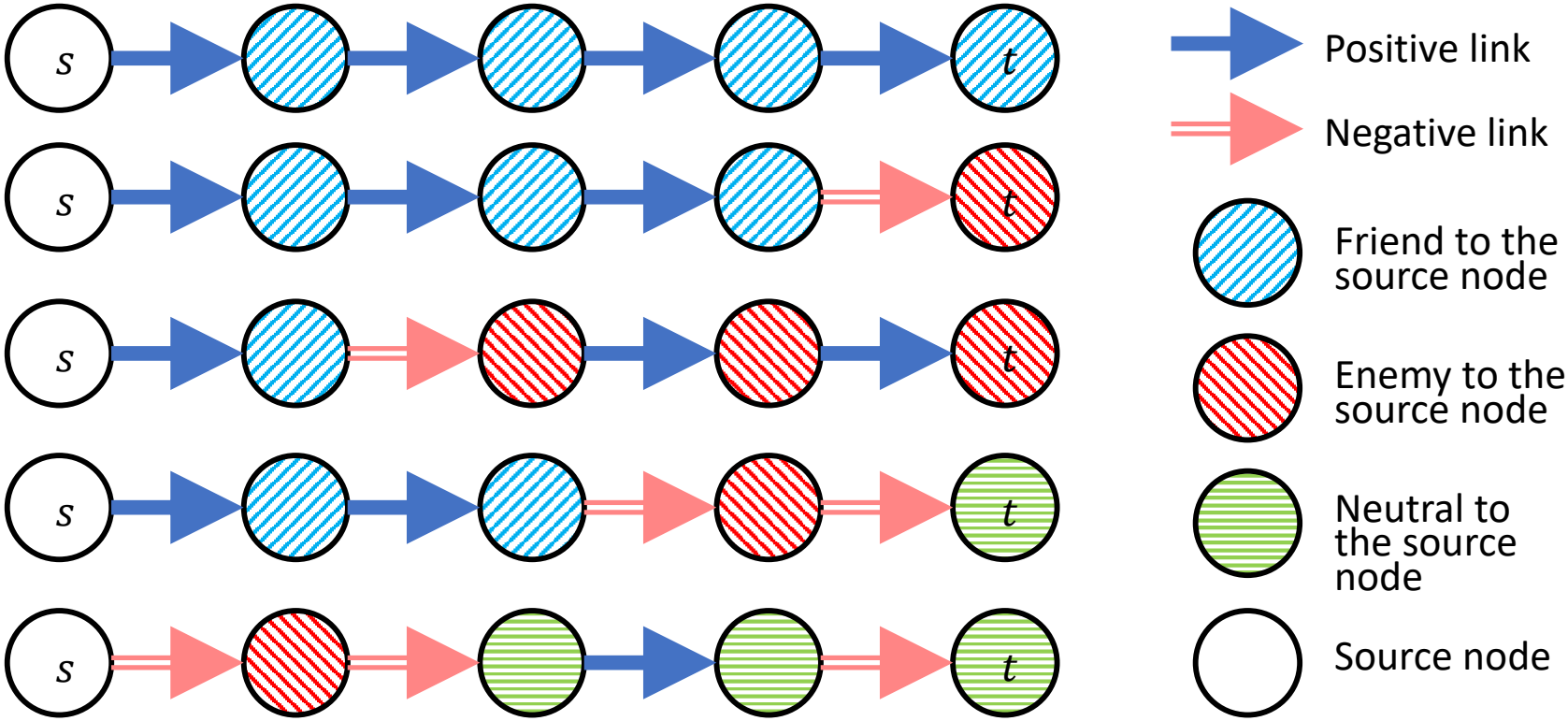


Figure: Example: five paths between the source (s) and target (t) nodes, and resulting relationships. While we assume a neutral relationship on the last two paths, social balance theory treats them as "friend" and "enemy", respectively.

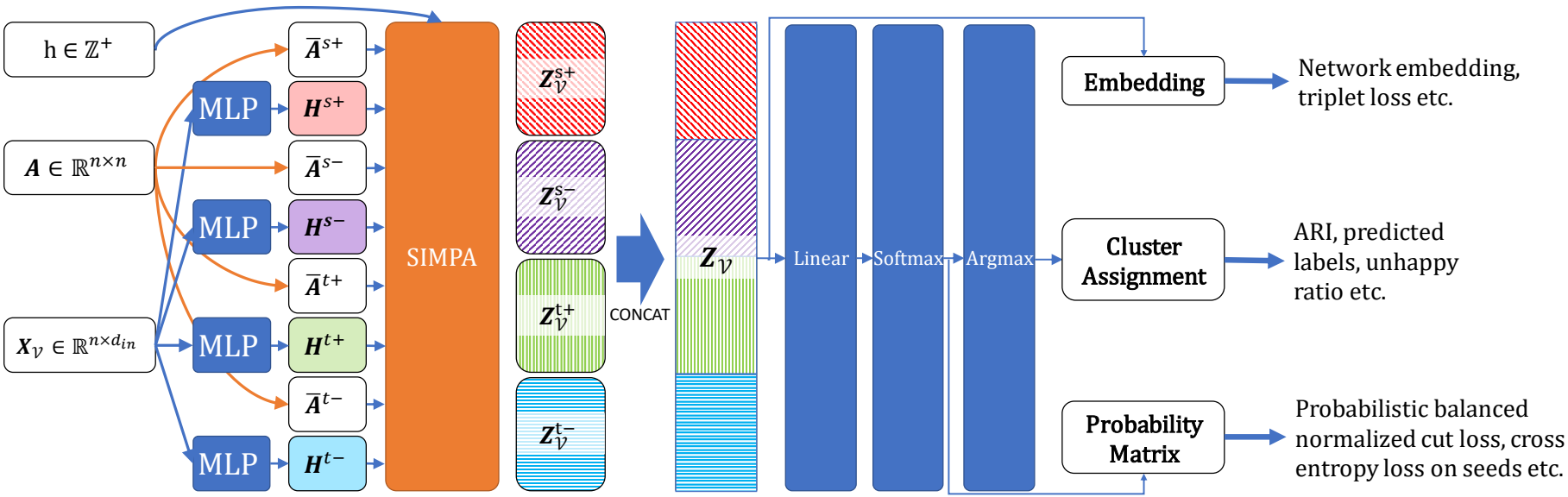
The **self-supervised** *Probabilistic Balanced Normalized Cut (PBNC) loss* function in SSSNET is

$$\mathcal{L}_{\text{PBNC}} = \sum_{k=1}^{K} \frac{(\mathbf{P}_{(:,k)})^T(\mathbf{D}^+ - \mathbf{A})\mathbf{P}_{(:,k)}}{(\mathbf{P}_{(:,k)})^T\overline{\mathbf{D}}\mathbf{P}_{(:,k)}}$$

- $\mathbf{P}$: assignment probability matrix; its i-th row contains predicted probabilities for different classes for node v_i.
- $\mathbf{P}_{(:,k)}$: the k^{th} column of the probability matrix $\mathbf{P}$
- $\overline{\mathbf{D}}_{ii} = \sum_{j=1}^{n} |A_{ij}|$ and $\mathbf{D}^+_{ii} = \sum_{j=1}^{n} A^+_{ij}$

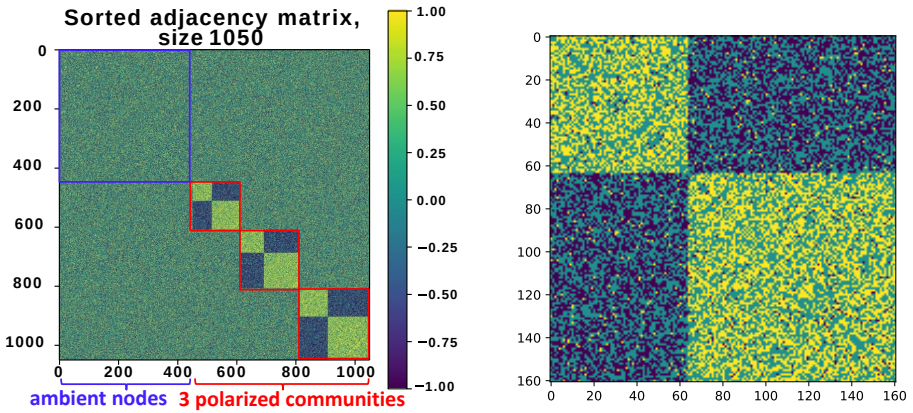
This is related to the (non-differentiable) Balanced Normalized Cut (BNC) [Chiang et al., 2012]. When some seed nodes have known labels, a **supervised** loss can be added to the loss function, similar to that in [Tian et al., 2019].

Framework Overview



Signed Clustering Data

- Signed Stochastic Block Models (SSBMs) and Polarized SSBMs generalized from [Cucuringu et al., 2019] and [Xiao et al., 2020], respectively, where a polarized SSBM model has several SSBMs planted in a random graph.

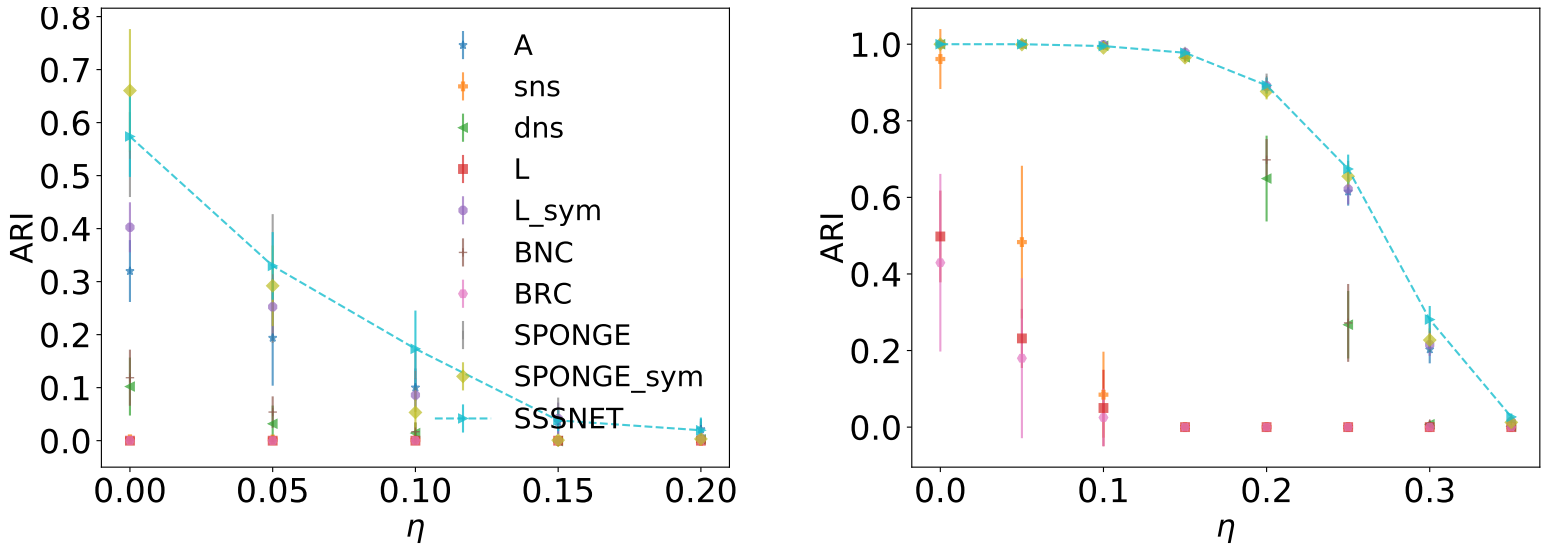


(a) Sorted adjacency matrix. (b) Polarized community #1.

Figure: A polarized SSBM model with 1050 nodes, $r = 3$ polarized communities of sizes 161, 197, and 242 (size ratio $\rho = 1.5$), and each SSBM has $K_1 = K_2 = K_3 = 2$ blocks, rendering $K = 7$.

- Real-world data sets.

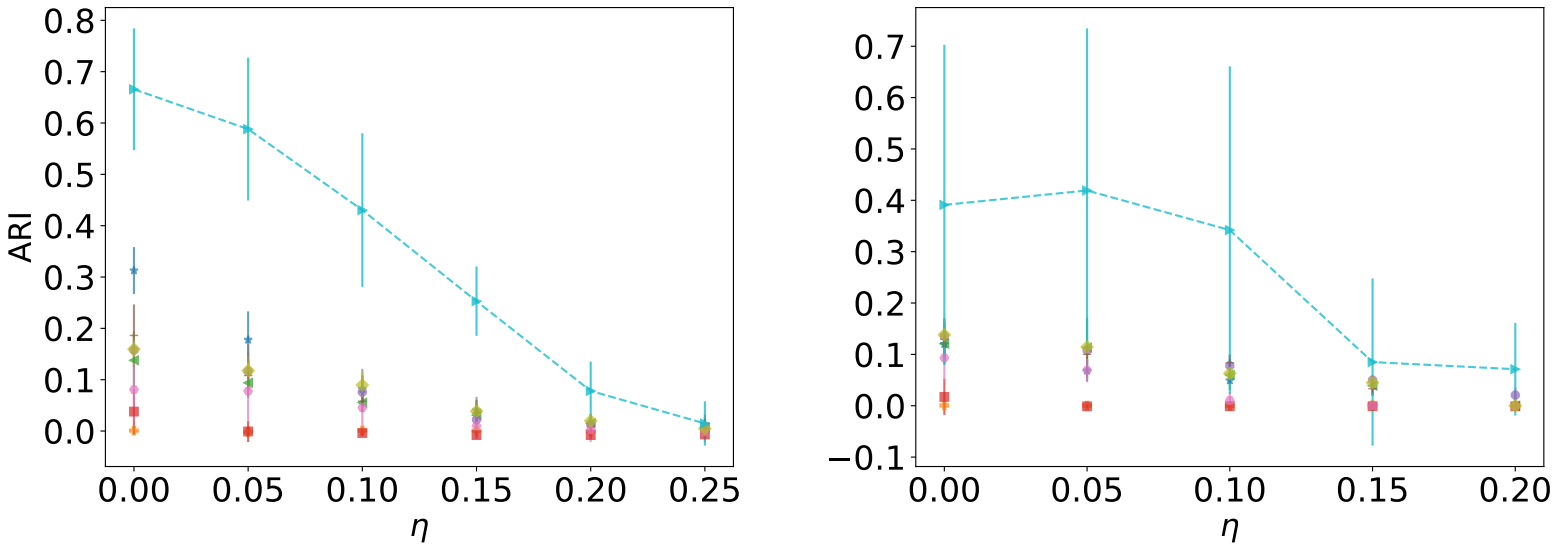
SSBM Results



(a) SSBM($n = 1000, p = 0.01$) (b) SSBM($n = 5000, p = 0.01$)

Figure: Test Adjusted Rand Index (ARI) comparison on SSBMs, averaged over ten runs. Dashed lines highlight SSSNET's performance. Error bars indicate one standard error. These are SSBM results with $K = 5$ and size ratio $\rho = 1.5$.

Polarized SSBM Results



(a) $n = 1050, r = 2, p = 0.1, \rho = 1$ (b) $n = 5000, r = 3, p = 0.1, \rho = 1.5$

Figure: Test ARI comparison on polarized SSBMs, averaged over ten runs. Dashed lines highlight SSSNET's performance. Error bars indicate one standard error.

Real-World Data Results

Clustering performance on real-world data sets; best is in **bold red**, and 2nd in underline blue. The first 3 rows are test ARIs, the 4th "ARI distance to best", the rest are unhappy ratios (%).

Data set	A	sns	dns	L	L_sym	BNC	BRC	SPONGE	SPONGE_sym	SSSNET
Sampson	0.32±0.10	0.15±0.09	0.33±0.10	0.16±0.05	0.35±0.09	0.32±0.12	0.21±0.11	0.36±0.11	0.34±0.11	**0.55±0.07**
Rainfall	0.61±0.08	0.28±0.03	0.65±0.04	0.46±0.06	0.58±0.07	0.62±0.05	0.47±0.05	N/A	0.75±0.09	**0.76±0.13**
S&P 1500	0.21±0.00	0.00±0.00	0.05±0.01	0.06±0.00	0.24±0.00	0.04±0.00	0.00±0.00	0.30±0.00	0.34±0.00	**0.66±0.00**
Fin-YNet	0.22±0.09	0.37±0.12	0.32±0.10	0.33±0.10	0.22±0.09	0.32±0.09	0.33±0.11	0.20±0.08	0.16±0.07	**0.00±0.00**
PPI	57.59±0.55	46.82±0.01	46.79±0.04	46.91±0.03	47.05±0.04	46.63±0.04	52.11±0.42	47.57±0.00	46.39±0.10	**17.64±0.84**
Wiki-Rfa	50.05±0.03	23.28±0.00	23.28±0.00	23.28±0.00	36.95±0.01	23.28±0.00	23.49±0.00	29.63±0.01	**23.26±0.00**	23.27±0.14

Conclusion and Future Work

SSSNET provides an end-to-end pipeline to create node embeddings and carry out signed clustering, with or without available additional node features, and with an emphasis on polarization. Future directions include:

- Apply the method to more networks without ground truth.
- Detect the number of clusters automatically.
- Address the performance in the very sparse regime.
- Apply signed clustering to cluster multivariate time series and leverage the clusters for time series prediction.
- Adapt our pipeline for constrained clustering.

Code:
`https://github.com/SherylHYX/SSSNET_Signed_Clustering`
Full paper (SDM 2022):
`https://arxiv.org/pdf/2110.06623.pdf`