DIGRAC: Digraph Clustering Based on Flow Imbalance

Yixuan He [1] Gesine Reinert [1,2] Mihai Cucuringu [1,2]

[1]University of Oxford, UK
[2]The Alan Turing Institute, UK

Motivation for Directed Networks

- Most existing methods that could be applied to directed clustering use local edge densities as main signal and directionality as additional signal
- We argue that even in the absence of any edge density differences, directionality can play a vital role in directed clustering as it can reveal latent properties of network flows.
- Therefore, instead of finding relatively dense groups of nodes in digraphs which have a relatively small amount of flow between the groups, our main goal is to recover clusters with **strongly imbalanced flow** among them, in the spirit of [Cucuringu et al., 2020], where directionality (i.e, edge orientation) is the main signal.

Flow Imbalance

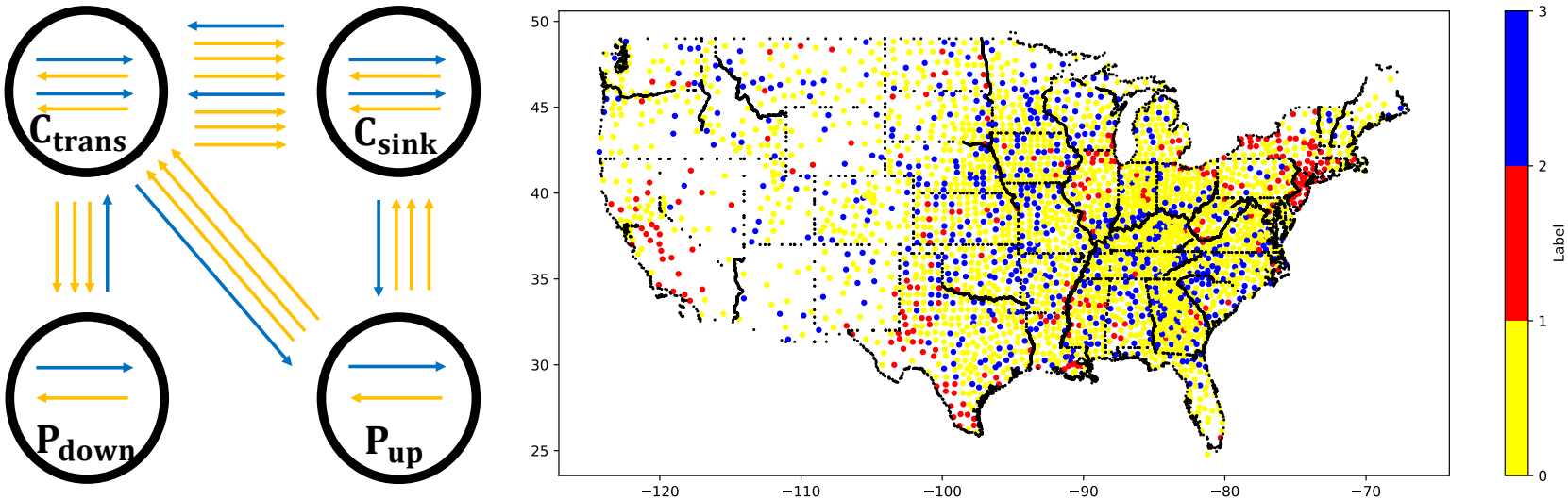


Figure: Visualization of directed flow imbalance: (a) A meta-graph which we hypothesize to be present on *Telegram* [Bovet and Grindrod, 2020]. More edge weights flow from $\mathcal{C}_{\text{trans}}$ to $\mathcal{C}_{\text{sink}}$ than in the other direction; (b) top pair imbalanced flow on *Migration* data [Perry, 2003], along with the geographic locations of the counties and state boundaries (in black): most edges flow from red (1) to blue (2).

Previous GNNs on Directed Networks

- DGCN [Tong et al., 2020b] uses first and second order proximity, constructs three Laplacians, but the method is space and speed inefficient.
- DiGCN [Tong et al., 2020a] simplifies DGCN, builds a directed Laplacian based on PageRank, and aggregates information dependent on higher-order proximity.
- MagNet [Zhang et al., 2021] builds upon [Cucuringu et al., 2020, Mohar, 2020] and introduces a complex Hermitian matrix that encodes undirected geometric structure in the magnitude of its entries, and directional information in their phase.
- The above methods all require known labels, which are not generally available for real-world data.

DIGRAC Approach

- We introduce a graph neural network framework to obtain node embeddings for directed networks in a self-supervised manner, including a novel probabilistic imbalance loss for node clustering.
- We propose **directed flow imbalance** measures, which are tightly related to directionality, to reveal clusters in the network even when there is no density difference between clusters.

- $\mathcal{V}$: 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
- For a *directed* network, $\mathbf{A}$ is usually *asymmetric*.
- A **clustering** into K clusters: a partition of the node set into disjoint sets $\mathcal{V} = \mathcal{C}_0 \cup \mathcal{C}_1 \cup \cdots \cup \mathcal{C}_{K-1}$
- Self-supervised: no label supervision

Probabilistic Cut

Our **self-supervised** loss function is inspired by [Cucuringu et al., 2020], aiming to cluster the nodes by maximizing a normalized form of cut imbalance across clusters.

For K clusters, the *assignment probability matrix* $\mathbf{P} \in \mathbb{R}^{n \times K}$ has as row i the probability vector $\mathbf{P}_{(\mathbf{i},:)} \in \mathbb{R}^K$ with entries denoting the probabilities of each node to belong to each cluster; its k^{th} column is denoted by $\mathbf{P}_{(:,k)}$.

The probabilistic cut from cluster $\mathcal{C}_k$ to $\mathcal{C}_l$ is defined as

$$W(\mathcal{C}_k, \mathcal{C}_l) = \sum_{i,j \in \{1,\ldots,n\}} \mathbf{A}_{i,j} \cdot \mathbf{P}_{i,k} \cdot \mathbf{P}_{j,l} = (\mathbf{P}_{(:,k)})^T \mathbf{A} \mathbf{P}_{(:,l)}.$$

The imbalance flow between clusters $\mathcal{C}_k$ and $\mathcal{C}_l$ is defined as

$$|W(\mathcal{C}_k, \mathcal{C}_l) - W(\mathcal{C}_l, \mathcal{C}_k)|, \quad \forall k, l \in \{0, \ldots, K-1\}.$$

For interpretability and ease of comparison, we normalize the imbalance flows to obtain an imbalance score with values in $[0, 1]$.
The probabilistic volume for cluster $\mathcal{C}_k$ is defined as

$$VOL(\mathcal{C}_k) = VOL^{(\text{out})}(\mathcal{C}_k) + VOL^{(\text{in})}(\mathcal{C}_k) = \sum_{i,j} (\mathbf{A}_{i,j} + \mathbf{A}_{j,i}) \cdot \mathbf{P}_{j,k}.$$

Then $VOL(\mathcal{C}_k) \geq W(\mathcal{C}_k, \mathcal{C}_l)$ for all $l \in \{0, \ldots, K-1\}$ and

$$\min(VOL(\mathcal{C}_k), VOL(\mathcal{C}_l)) \geq |W(\mathcal{C}_k, \mathcal{C}_l) - W(\mathcal{C}_l, \mathcal{C}_k)|. \quad (1)$$

Imbalance Score

The imbalance term, which is used in most of our experiments, denoted $\mathrm{CI}^{\mathrm{vol\_sum}}$, is defined as

$$\mathrm{CI}^{\mathrm{vol\_sum}}(k,l) = 2\frac{|W(\mathcal{C}_k,\mathcal{C}_l) - W(\mathcal{C}_l,\mathcal{C}_k)|}{VOL(\mathcal{C}_k) + VOL(\mathcal{C}_l)} \in [0,1]. \qquad (2)$$

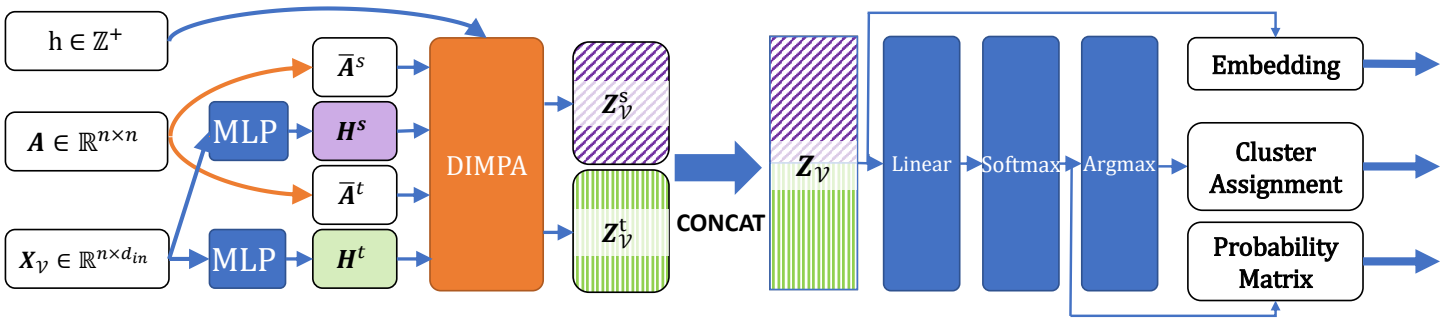
The aim is to find a partition which maximizes the imbalance flow under the constraint that the partition has at least two sets, to capture groups of nodes which could be viewed as representing clusters in the meta-graph. The normalization by the volumes penalizes partitions that put most nodes into a single cluster. The range $[0,1]$ follows from Eq. (1).

Global Imbalance Objective

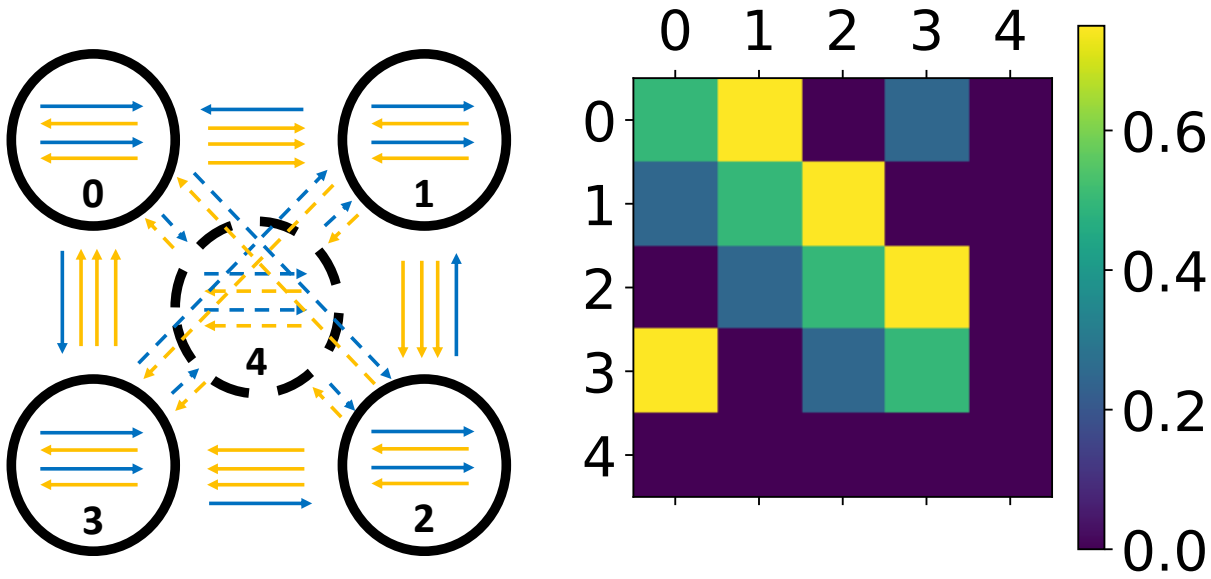
To obtain a **global probabilistic imbalance score**, based on $\mathrm{CI}^{\text{vol\_sum}}$ from Eq. (2), we average over pairwise imbalance scores of different pairs of clusters. Since the scores discussed are symmetric and the cut difference before taking absolute value is skew-symmetric, we only need to consider the pairs $\mathcal{T} = \{(\mathcal{C}_k, \mathcal{C}_l) : 0 \leq k < l \leq K-1, k, l \in \mathbb{Z}\}$. We consider a "*sort*" variant to select these pairs. With $\mathcal{T}(\beta) = \{(\mathcal{C}_k, \mathcal{C}_l) \in \mathcal{T} : \mathrm{CI}^{\text{vol\_sum}}(k,l)$ is among the top β values$\}$, we set

$$\mathcal{O}^{\text{sort}}_{\text{vol\_sum}} = \frac{1}{\beta} \sum_{(\mathcal{C}_k, \mathcal{C}_l) \in \mathcal{T}(\beta)} \mathrm{CI}^{\text{vol\_sum}}(k,l), \quad \text{and} \quad \mathcal{L}^{\text{sort}}_{\text{vol\_sum}} = 1 - \mathcal{O}^{\text{sort}}_{\text{vol\_sum}} \tag{3}$$

as the corresponding loss function.


h ∈ ℤ+
A ∈ ℝ^{n×n}
X_V ∈ ℝ^{n×d_in}
MLP
MLP
Ā^s
H^s
Ā^t
H^t
DIMPA
Z_V^s
Z_V^t
CONCAT
Z_V
Linear
Softmax
Argmax
Embedding
Cluster Assignment
Probability Matrix
Network embedding etc.
Pairwise/global cut imbalance scores etc.
ARI, predicted labels etc.

Data: Directed Stochastic Block Models (DSBMs)



(a) Meta-graph.

(b) Meta-graph adjacency matrix.

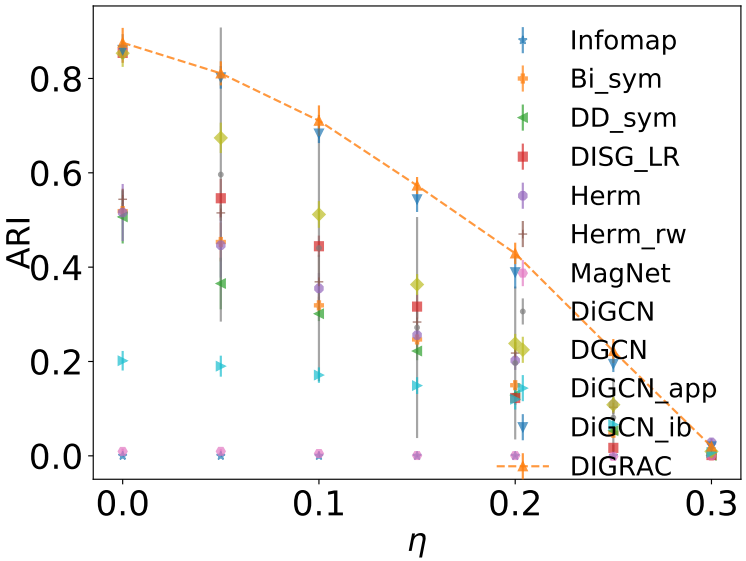
Figure: Visualization of a DSBM with a cycle meta-graph with ambient nodes, for a total of 5 clusters. 75% of the edges flow in direction $0 \to 1 \to 2 \to 3 \to 0$, while 25% flow in the opposite direction. Cluster 4 is the ambient cluster. In (a), dashed lines indicate flows with random equally likely directions; these flows do not exist in the meta-graph adjacency matrix. For (b), the lighter the color, the stronger the flow.

Data: Real-World Data Sets

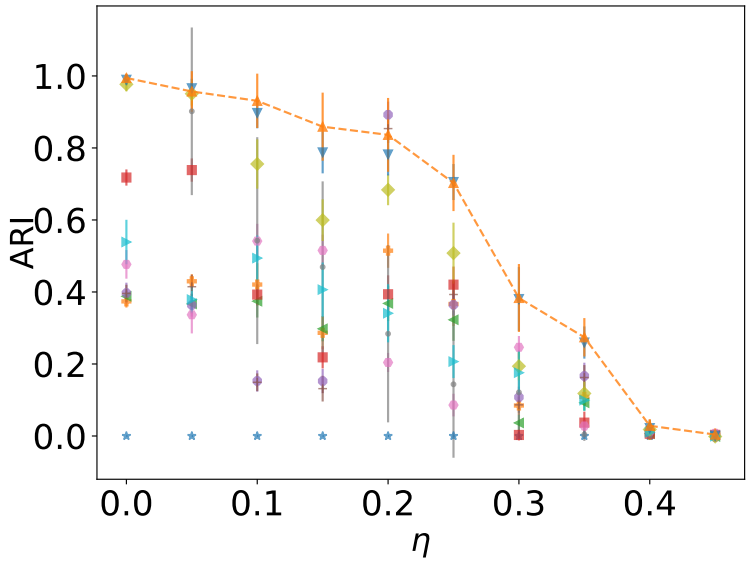
Table: Summary statistics for the real-world networks.

data set	n	$\lvert\mathcal{E}\rvert$	density	weighted
Telegram	245	8,912	$1.28 \cdot 10^{-2}$	True
Blog	1,222	19,024	$1.49 \cdot 10^{-1}$	True
Migration	3,075	721,432	$7.63 \cdot 10^{-2}$	True
WikiTalk	2,388,953	5,018,445	$8.79 \cdot 10^{-7}$	False
Lead-Lag	269	29,159	$4.04 \cdot 10^{-1}$	True

DSBM Results



(a) DSBM("cycle", $n = 5000, p = 0.01$)



(b) DSBM("complete", $n = 1000, p = 0.1$)

Figure: Test Adjusted Rand Index comparison on DSBMs, averaged over 50 runs. Dashed lines highlight DIGRAC's performance. Error bars indicate one standard error. These are DSBM results with $K = 5$ clusters and size ratio $\rho = 1.5$. Both networks contain ambient nodes, and p is the edge density.

Real-World Data Results

Performance comparison on real-world data sets. The best is marked in **bold red** and the second best is marked in underline blue.

Metric	Data set	InfoMap	Bi_sym	DD_sym	DISG_LR	Herm	Herm_rw	DIGRAC
$\mathcal{O}^{\text{sort}}_{\text{vol\_sum}}$	*Telegram*	0.04±0.00	0.21±0.0	0.21±0.0	0.21±0.01	0.2±0.01	0.14±0.0	**0.32±0.01**
	Blog	0.07±0.00	0.07±0.0	0.0±0.0	0.05±0.0	0.37±0.0	0.0±0.0	**0.44±0.0**
	Migration	N/A	0.03±0.00	0.01±0.00	0.02±0.00	0.04±0.00	0.02±0.00	**0.05±0.00**
	WikiTalk	N/A	N/A	N/A	0.18±0.03	0.15±0.02	0.0±0.0	**0.24±0.05**
	Lead-Lag	N/A	0.07±0.01	0.07±0.01	0.07±0.01	0.07±0.02	0.07±0.02	**0.15±0.03**
$\mathcal{O}^{\text{naive}}_{\text{vol\_sum}}$	*Telegram*	0.01±0.00	0.26±0.0	0.26±0.0	0.26±0.01	0.25±0.02	0.23±0.0	**0.27±0.01**
	Blog	0.00±0.00	0.07±0.0	0.0±0.0	0.05±0.0	0.37±0.0	0.0±0.0	**0.44±0.0**
	Migration	N/A	0.01±0.00	0.01±0.00	0.01±0.00	0.02±0.00	0.01±0.00	**0.04±0.01**
	WikiTalk	N/A	N/A	N/A	0.1±0.02	0.04±0.0	0.0±0.0	**0.12±0.01**
	Lead-Lag	N/A	0.30±0.06	0.28±0.06	0.27±0.06	0.29±0.05	0.29±0.05	**0.32±0.11**

Conclusion and Future Work

For directed networks, DIGRAC provides an end-to-end pipeline to create node embeddings and perform directed clustering, with or without available additional node features or cluster labels. The main novelty is an objective based on flow imbalance.

Future directions:

- Additional experiments in the semi-supervised setting, when there exist seed nodes with known cluster labels, or when additional information is available in the form of *must-link* and *cannot-link* constraints, popular in the *constrained clustering* literature [Cucuringu et al., 2016].
- Extend our framework to also detect the number of clusters [Riolo et al., 2017], instead of specifying it a-priori, as this is typically not available in real-world applications.

Conclusion and Future Work (cont.)

- Automatically detect the value β used in the current model, to select the subset of influential pairs of imbalances.
- Address the performance in the sparse regime, where spectral methods are known to underperform, and various regularizations have been proven to be effective both on the theoretical and experimental fronts.
- Adapt our pipeline for directed clustering in extremely large networks, possibly combined with sampling methods or mini-batch [Hamilton et al., 2017]

Full paper (LoG 2022):
`https://proceedings.mlr.press/v198/he22b.html`
Code: `https://github.com/SherylHYX/DIGRAC_Directed_Clustering`
More about me: `https://sherylhyx.github.io/`