IP901/20 Project
Fall 2022
Neetha Reddy
IMT2018050

Node Co-association Prediction in Communities in Dynamic Sparse Directed Networks

Abstract

This paper attempts to validate the CAP-DSDN algorithm proposed in [1], use alternative pre-processing techniques to those suggested in the paper, and compare their accuracies.

1 Problem statement

Predicting the future community structure of a real-world graph is a tricky problem. This can be attributed to the limited availability of training data, the directional nature of the graph and edge sparsity due to inconsistencies in the collected data. So we can generalise the problem as predicting the community structure in a dynamic sparse directed network (DSDN). Now that we have generalised the problem, we can proceed to calculate the communities for the existing data across timesteps. However, the absence of benchmark data or ground truths combined with the limited literature available on community detection in directed and sparse networks renders this computation rather challenging.

In other to deal with these limitations, [1] proposes a three-step algorithm, CAP-DSDN, for predicting the likelihood of any two nodes belonging to the same community (CAP) in such a network. This algorithm uses the persistent community behaviour of nodes in the time series to predict its community behaviour in the future.

2 Preliminaries

Here, we define the Co-association Matrix (CAM) and persistent community behaviour.

Definition 1. A Co-association Matrix (CAM) is a symmetric matrix with rows and columns representing the same set of data items, say countries in the migration flow network, in the same order, and each cell (i, j) indicates if the item in the ith row and that in the jth column is associated with the same cluster, i.e., belong to the same community in the case of networks[1].

Definition 2. Persistent Community Behavior is a property of a dynamic network node with a high likelihood of non-zero co-association with other nodes in a network across several snapshots[1].

Definition 3. Node Significance is the temporal significance of the node and is based on the persistence of its co-associations in the network.

3 Approach

The CAP-DSDN algorithm has 3 steps:

Step-1: Performing Community Detection(CD) across each timestep and computing aggregated CAM (we will refer to this aggregated CAM as PCAM going forward). Here CD algorithms used are Speaker Listener Label Propagation Algorithm(SLPA)[2], Infomap (IMAP) algorithm[3], Modularity Optimization for Directed Networks (MODN) algorithm[4].

Step-2: Using node significance to calculate the entropy of all nodes across timesteps, identify the threshold as a sharp increase in these sorted entropies and filter out nodes(and their corresponding edges) above this threshold.

Step-3: Predicting using Autoregressive Models (here VAR and ARMA). Validated results using Normalized Mutual Information Score (NMI) and Rand Index (RI).

4 Experiments

4.1 Pre-processing

In [1], the DSDN has been passed directly to the community detection(CD) algorithms without any prior node or edge filtering. Owing to the sparsity of the network, my hypothesis is that filtering out "insignificant" nodes before passing them to the CD algorithms will ensure better community structures. After researching several centrality measures of a network, I have decided to use the node percolation centrality proposed in [5] for this filtering step. Node percolation centrality quantifies the relative impact of nodes based on their topological connectivity, as well as their percolation states. On passing the migration data, I obtained the node-wise scores for each timestep. Across all time steps, there are about 49 nodes with zero node percolation centrality and these 49 nodes and their corresponding edges have been filtered out. Post this pre-processing step, the DSDN has 150 nodes(24.6% decrease from the original) and 63565 edges(15.8% decrease from the original).

One interesting thing observed from Fig. 1 is that popular migrant countries like the USA, the UK, the UAE, etc. are included in the zero node percolation list determined by the centrality measure. One possible explanation for this could be

```
49
array(['Bahamas', 'Botswana', 'Brunei Darussalam', 'Cayman Islands',
       'Curacao ', 'Denmark', 'Dominica', 'Finland', 'France', 'Germany',
       'Iceland', 'Ireland', 'Japan', 'Kiribati', 'Luxembourg',
       'Madagascar', 'Malta', 'Monaco', 'Mozambique', 'Niue', 'Norway',
       'Panama', 'Papua New Guinea', 'Qatar', 'Rep. of Korea',
       'Saint Lucia', 'San Marino', 'Seychelles', 'Tonga',
       'Trinidad and Tobago', 'Tunisia', 'Turkmenistan',
       'Turks and Caicos Islands', 'Tuvalu', 'Türkiye', 'Uganda',
       'Ukraine', 'United Arab Emirates',
       'United Kingdom of Great Britain and Northern Ireland',
       'United Rep. of Tanzania', 'United States of America', 'Uruguay',
       'Uzbekistan', 'Venezuela (Bolivarian Republic of)', 'Viet Nam',
       'Western Sahara', 'Yemen', 'Zambia', 'Zimbabwe'], dtype=object)
```

Figure 1: Filtered nodes during node percolation-based pre-processing step

the low outflow and high inflow associated with these nodes which makes them "endpoints" of the network(away from the central structure of the DSDN). The effect of filtering out these nodes, in the beginning, is yet to be determined.

4.2 Step-1: Community Detection

In [1], the ensemble community detection algorithms used are SLPA, IMAP and MODN. Here, SLPA identifies communities based on directional propagation of labels, IMAP uses random walks and MODN uses high and low densities of intra-community and inter-community edges. Thus, these algorithms fall under different CD models based on dynamic processes on graphs, on a flow model, and on a null model, respectively. The quality of the communities resulting from these algorithms is computed using link modularity (LM), internal edge density (IED), average internal degree (AID), cut ratio (CR), and Z-modularity (ZM).

Apart from these algorithms, I have explored Spectral and Stochastic Block Model(SBM) based CD algorithms like SpectralClustering from sklearn library, Motif[6] and SparseBM[7]. The community quality metrics have been shown in Fig. 2.

Algorithm	LM		IED		AID		CR		ZM	
	Original	Filtered	Original	Filtered	Original	Filtered	Original	Filtered	Original	Filtered
SLPA	0.05	0.06	0.20	0.34	1.40	4.40	0.08	0.13	(0.01)	0.00
IMAP	**0.06**	**0.07**	0.32	0.40	7.09	**22.50**	0.08	0.03	0.01	0.01
MODN	0.02	0.02	0.38	0.44	4.18	6.56	0.08	0.14	**0.08**	0.04
Spectral	0.06	0.04	**0.38**	0.36	7.68	8.62	0.05	0.10	0.03	**0.06**
Motif	0.03	0.04	0.21	**0.58**	5.47	5.66	0.08	0.09	0.01	(0.00)
SparseBM	0.02	0.02	0.12	0.29	**7.74**	12.13	**0.08**	**0.17**	(0.17)	(0.15)

Figure 2: Community Quality metrics for the CD algorithms for the year 2016

One problem arises with using the spectral and stochastic block model-based CD algorithms specified above. These new CD algorithms require us to specify the number of clusters apart from some additional algorithm-specific attributes which are yet to be properly experimented upon. In this case, we are considering the number of clusters that the IMAP map algorithm gives to be this number. IMAP was used to determine the number of clusters for these new algorithms due to its ability to form compact yet fragmented communities. This can be observed across timesteps as shown in Fig. 3.

Algorithm	SLPA		MODN		IMAP	
	Original	Filtered	Original	Filtered	Original	Filtered
2000	52	68	17	58	20	68
2001	52	67	19	60	25	67
2002	56	71	17	60	15	64
2003	55	70	16	60	7	51
2004	54	70	20	63	18	71
2005	55	71	19	61	24	67
2006	53	70	13	57	11	61
2007	50	68	13	57	14	62
2008	48	67	14	57	8	59
2009	46	66	13	56	7	62
2010	44	64	13	57	11	58
2011	40	63	14	58	6	57
2012	43	65	14	57	6	57
2013	39	62	13	58	6	57
2014	35	60	12	57	7	56
2015	34	58	13	57	6	53
2016	35	60	13	57	6	52
2017	35	60	12	57	6	53
2018	35	60	11	57	6	54

Figure 3: Number of Clusters formed across timesteps

I also observed that the initial pre-processing step led to the formation of a large number of overly fragmented communities which might give undesired results going forward. So we will be using the original DSDN for now.
Now that we have our communities from various algorithms, we can proceed to compute CAMs and their corresponding Probability of Co-association Matrices (PCAMs) as specified in [1].

4.3 Step-2: Node-Filtering the Network

In [1], the authors have defined a node significance-based entropy. Later, they identified an entropy threshold as the point of a steep increase in the line plot of sorted entropies of all nodes. All nodes which had an entropy higher than this threshold along with the edges associated with these nodes were then filtered out. This filtering was done as real-world graphs have been observed to exhibit two groups of nodes that correspond to low and high entropy rates. Here, the presence of any node from the high entropy group tends to sharply increase the size of the node-filtered network. This will help densify the DSDN. But this method completely fails to account for high entropy nodes which might contain useful information.

Rather than using this entropy-based node filtering, I have used edge filtering using Variance thresholding on the PCAMs to filter out constant(low variance) columns. This way, we do not lose out on (potential) important high entropy nodes. Further, we can also reconstruct the original format of the PCAM by refactoring these filtered

columns back into the final predicted PCAM.
After experimentation, I observed that a variance threshold of 0.0035 was giving the highest accuracy as shown in Fig. 4.

Variance =>	0.003(5884 edges filtered)			0.005(5922 edges filtered)			0.01(7703 edges filtered)		
	MAE	RMSE	R2	MAE	RMSE	R2	MAE	RMSE	R2
2019	0.05	0.01	**0.92**	0.05	0.01	0.92	0.06	0.01	0.91
2020	0.12	0.03	**0.64**	0.12	0.03	0.63	0.14	0.04	0.58
2021	0.18	0.06	**0.02**	0.18	0.06	-0.02	0.21	0.07	-0.14

Figure 4: Variance Threshold

Note: here is that the scores shown above are only representative of the threshold-filtered PCAM.

4.4 Step-3: CAP Using Autoregressive Models

As described in [1], I fitted the threshold filtered PCAM to the VAR model and predicted the PCAMs for future timesteps.

However, both models considered are Autoregressive which makes the prediction an auto-regressive solution. Here NMI and RI are cluster metrics and therefore will not give useful information about the accuracy of our proposed method. So I have used Mean Absolute Error(MAE), Root Mean Squared Error RMSE and R2 scores as the accuracy metrics for my predictions.

5 Results

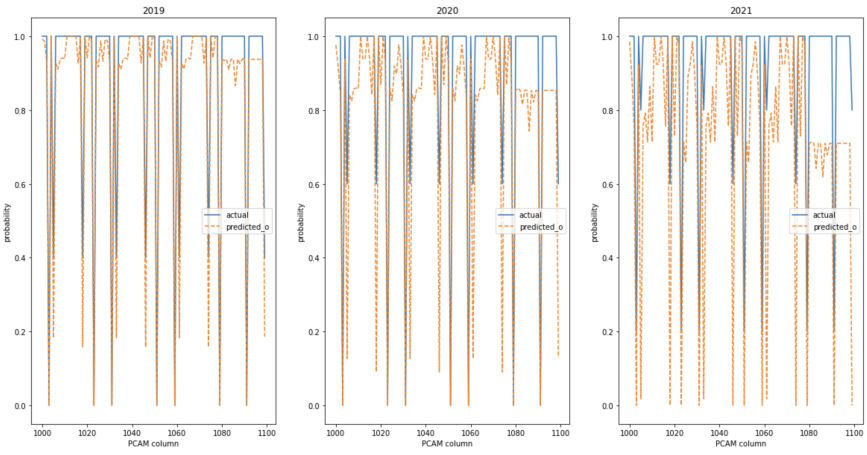

Figure 5: Accuracy line plots for 2019,2020 and 2021 using IMAP

For the scope of this paper, we have analysed pre-processing and predicting future community structure of DSDNs using case study of international refugee migration flow between countries. I used the UNHCR migration dataset[8] which has information on refugees dating back to 1951, although the first census of asylum applicants

was made in 2000. We thus focus on the years 2000 through 2021 by using yearly data beginning in 2000. There are 179 countries of origin and 199 nations offering refuge during this time. Thus, there are 199 nodes in our DSDN for 22 snapshots. The time interval length was chosen as 5 during the PCAM computation. You can find the source code of the work discussed in this paper on GitHub[9].

6 Conclusion and Future Work

This paper builds on the methods proposed in [1] and employed additional preprocessing steps like node percolation-based filtering and variance threshold-based filtering, explored the usage of spectral-based CD algorithms like SpectralClustering and Motif based clustering, as well as SBM-based algorithms like SparseBM clustering. However, there is still scope for additional work on the subject. Some of them are listed below:

1. Currently PCAMs are constructed as moving averages over some time intervals (interval sizes - 3 or 5). Using weighted CAMs while computing PCAMs and then normalising this PCAM might give better results since we are giving a higher weightage to the most recent timestep rather than equally dividing it among the time interval. This might prove significantly useful when the number of timesteps is higher.
2. Basic implementations of the newly introduced CD algorithms were implemented due to the complexity of the attributes involved. Further hyperparameter tuning can be done to increase accuracy, especially for motif-based clustering.
3. Additional experimentation is required to ensure the seamless incorporation of previously filtered nodes/edges into the final result.
4. Reconstructing the predicted PCAMs into community structures. Here PCAMs show the likelihood of 2 countries belonging to the same community based on past data. What do the semantics of our results mean? Does this mean there is more predicted refugee/migrant flow between countries with a high PCAM cell value? These questions are yet to be answered.

7 Acknowledgements

I would like to thank Prof. Jaya Sreevalsan Nair for guiding me through the knowledge discovery process of community detection algorithms and Dynamic Sparse Directed Networks (DSDNs).

8 References

[1] Sreevalsan-Nair, J. and Jakher, A. (2022). CAP-DSDN: Node Co-association Prediction in Communities in Dynamic Sparse Directed Networks and a Case Study

of Migration Flow. In Proceedings of the 14th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management - KDIR, ISBN 978-989-758-614-9; ISSN 2184-3228, pages 63-74. DOI: 10.5220/0011537600003335

[2] Xie, J., Szymanski, B. K., and Liu, X. (2011). SLPA: Uncovering overlapping communities in social networks via a speaker-listener interaction dy- namic process. In IEEE 11th Intl. Conf. on Data Mining Workshops, pages 344–349. IEEE.

[3] Rosvall, M. and Bergstrom, C. T. (2008). Maps of random walks on complex networks reveal com- munity structure. PNAS, 105(4):1118–1123.

[4] Leicht, E. A. and Newman, M. E. (2008). Community structure in directed networks. Phys. Rev. Letters, 100(11):118703.

[5] Piraveenan M, Prokopenko M, Hossain L (2013) Percolation Centrality: Quantifying Graph-Theoretic Impact of Nodes during Percolation in Networks. PLOS ONE 8(1): e53095. https://doi.org/10.1371/journal.pone.0053095

[6] Underwood, William G., Andrew Elliott, and Mihai Cucuringu. "Motif-based spectral clustering of weighted directed networks." Applied Network Science 5.1 (2020): 1-41.

[7] Frisch, Gabriel, Jean-Benoist Leger, and Yves Grandvalet. "SparseBM: A Python Module for Handling Sparse Graphs with Block Models." (2021).

[8] UNHCR Refugee Data Finder

[9] Community detection in DSDNs