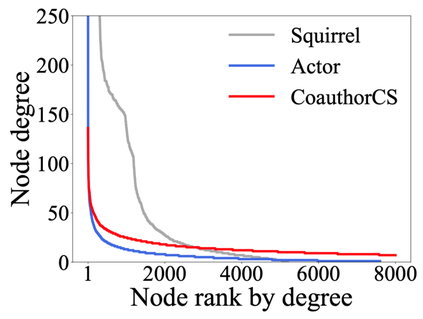
SMU SINGAPORE MANAGEMENT UNIVERSITY
School of Information Systems

Tail-GNN: Tail-Node Graph Neural Networks

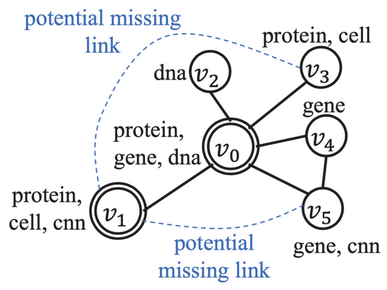
Zemin Liu, Trung-Kien Nguyen, Yuan Fang

In Proceedings of the 27th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD-21)
14th -18th August, 2021

Motivation



(a) Long-tailed node distribution



Tail node v_1 **Head node v_0**

(b) Toy citation network

Problem: Robust tail node embedding.

Tail nodes
- Small neighborhood
- Potentially suffer from missing information

Challenges:

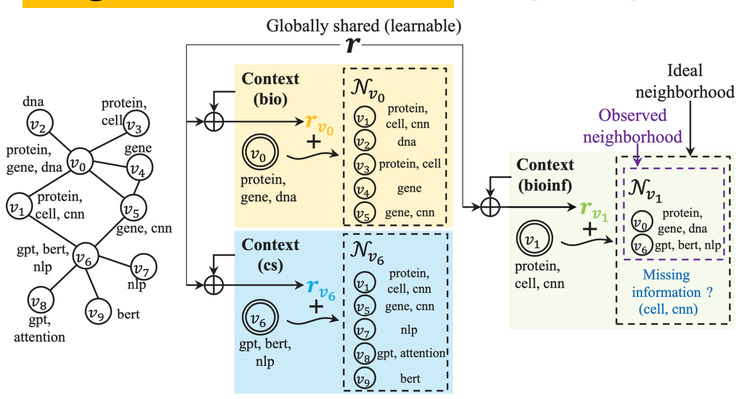
C1: How to uncover the missing neighborhood information for tail nodes?

C2: How to localize the missing information for each tail node while maintaining the generality across nodes?

The proposed model: Tail-GNN

Neighborhood translation

$$\mathbf{h}_v + \mathbf{r}_v \approx \mathbf{h}_{\mathcal{N}_v}$$



(a) Toy network (b) Neighborhood translation for head nodes (c) Neighborhood translation for tail nodes

- **Key idea**
  - Neighborhood translation
- **First challenge**
  - predict the missing neighborhood information for tail nodes by exploiting a transferable neighborhood translation
- **Second challenge**
  - tailor the shared neighborhood translation to each target node w.r.t. its local context.

- **Neighborhood of head nodes**
  - Observed neighborhood: complete and representative
  - no missing information

$$\mathbf{m}_v = \mathbf{h}_{\mathcal{N}_v^*} - \mathbf{h}_{\mathcal{N}_v} = \mathbf{0}$$

- **Neighborhood of tail nodes**
  - Observed neighborhood: not representative enough
  - Imperative: uncover the missing information

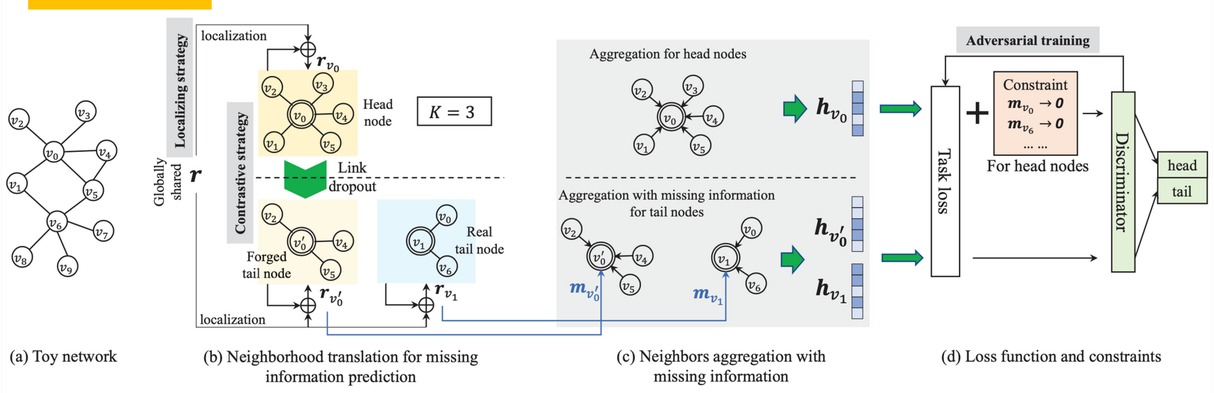
$$\mathbf{m}_v = \mathbf{h}_{\mathcal{N}_v^*} - \mathbf{h}_{\mathcal{N}_v} \neq \mathbf{0}$$

Predicting missing information for tail node v

$$\mathbf{h}_{\mathcal{N}_v^*} = \mathbf{h}_v + \mathbf{r}_v$$

$$\mathbf{m}_v = \mathbf{h}_v + \mathbf{r}_v - \mathbf{h}_{\mathcal{N}_v}$$

Tail-GNN



(a) Toy network (b) Neighborhood translation for missing information prediction (c) Neighbors aggregation with missing information (d) Loss function and constraints

Contrastive strategy

- **Head nodes**

$$\mathbf{h}_v^l + \mathbf{r}_v^l - \mathbf{h}_{\mathcal{N}_v}^l \rightarrow \mathbf{0}$$

(Embedding of observed neighborhood)

- **Tail nodes**
  - Real and forged tail nodes
  - Robust tail node embedding: uncover the missing information

$$\mathbf{m}_v^l = \mathbf{h}_{\mathcal{N}_v^*}^l - \mathbf{h}_{\mathcal{N}_v}^l = \mathbf{h}_v^l + \mathbf{r}_v^l - \mathbf{h}_{\mathcal{N}_v}^l$$

Localizing strategy

$$\mathbf{r}_v^l = \phi(\mathbf{h}_v^l, \mathbf{h}_{\mathcal{N}_v}^l, \mathbf{r}^l; \theta_\phi^l) = (\gamma_v^l + \mathbf{1}) \odot \mathbf{r}^l + \beta_v^l$$

(Local context; Scaling vector; Shifting vector)

Neighborhood aggregation

- **Head nodes**

$$\mathbf{h}_v^{l+1} = \mathcal{M}(\mathbf{h}_v^l, \{\mathbf{h}_i^l : i \in \mathcal{N}_v\}; \theta^{l+1})$$

(Message passing function)

- **Tail nodes**

$$\mathbf{h}_v^{l+1} = \mathcal{M}(\mathbf{h}_v^l, \{\mathbf{m}_v^l\} \cup \{\mathbf{h}_i^l : i \in \mathcal{N}_v\}; \theta^{l+1})$$

(missing neighborhood information; observed neighborhood)

Overall objective

Task loss $\mathcal{L}_t = \sum_{v \in \mathcal{V}_{\text{tr}}} \text{CrossEnt}(\mathbf{h}_v^\ell, \mathbf{y}_v) + \lambda_t \|\Theta\|_2^2$

Loss for head missing information constraint $\mathcal{L}_m = \sum_{v \in \mathcal{V}_{\text{tr}}} I_v \sum_{l=1}^{\ell} \|\mathbf{m}_v^{l-1}\|_2^2$

Loss for adversarial constraint (Discriminator) $\mathcal{L}_d = \sum_{v \in \mathcal{V}_{\text{tr}}} \text{CrossEnt}(I_v, D(\mathbf{h}_v^\ell; \theta_d)) + \lambda_d \|\theta_d\|_2^2$

Overall objective

$$\min_\Theta \max_{\theta_d} \; \mathcal{L}_t + \mu \mathcal{L}_m - \eta \mathcal{L}_d$$

Experiments

Experimental setup

	# Nodes	# Edges	# Features	# Classes	# Tail ($K = 5$)
Email	1,005	25,571	128	42	235
Squirrel	5,201	217,073	2,089	5	942
Actor	7,600	33,391	931	5	4,823
CoauthorCS	18,333	327,576	6,805	15	8,037
Amazon	937,349	12,455,925	100	44	248,125

- **Base GNN models**
  - GCN [1]
  - GAT [2]
  - GraphSAGE [3]
- **Baselines** (based on GCN)
  - Conventional: DeepWalk [4], GCN [1]
  - Refinement: Additive [5], a la carte [6], meta-tail2vec [7]
  - Robust models: SDNE [8], ARGA [9], DDGCN
  - Degree-aware models: Demo-Net [11], role2vec

Node classification

Table 2: Evaluation on tail node classification using GCN as the base model.

Henceforth, tabular results are in percent; the best result is **bolded** and the runner-up is underlined; a dash (-) denotes no result reported for failing to work on a large dataset.

Methods	Email		Squirrel		Actor		CoauthorCS		Amazon	
	Accuracy	Micro-F	Accuracy	Micro-F	Accuracy	Micro-F	Accuracy	Micro-F	Accuracy	Micro-F
DeepWalk	54.4 ± 0.3	51.3 ± 0.3	28.8 ± 1.6	28.0 ± 2.3	21.8 ± 0.6	18.2 ± 0.9	84.1 ± 0.7	81.5 ± 0.7	83.7 ± 0.1	74.3 ± 0.6
GCN	57.9 ± 1.2	57.7 ± 1.3	24.8 ± 1.3	23.2 ± 1.8	29.7 ± 0.2	15.0 ± 0.9	88.4 ± 0.1	86.1 ± 0.1	82.3 ± 0.2	70.6 ± 0.1
Additive	55.4 ± 0.4	52.5 ± 0.2	27.0 ± 1.7	22.9 ± 1.6	28.1 ± 0.3	15.1 ± 1.3	89.5 ± 0.1	87.8 ± 0.1	84.2 ± 0.2	73.2 ± 0.6
a la carte	21.1 ± 0.4	17.9 ± 0.5	22.5 ± 1.1	22.5 ± 0.7	28.0 ± 0.5	14.8 ± 1.4	88.7 ± 0.2	86.7 ± 0.3	81.1 ± 0.1	69.7 ± 0.7
meta-tail2vec	57.1 ± 0.1	55.3 ± 0.2	25.1 ± 0.5	21.5 ± 0.3	29.7 ± 0.4	20.1 ± 0.7	89.3 ± 0.1	87.4 ± 0.1	81.9 ± 0.1	71.4 ± 0.4
SDNE	32.9 ± 0.6	29.8 ± 0.5	23.8 ± 3.2	16.6 ± 6.2	24.4 ± 0.8	12.6 ± 5.6	70.6 ± 0.9	64.5 ± 1.1	-	-
ARGA	45.1 ± 0.9	41.2 ± 1.0	22.4 ± 1.0	22.8 ± 1.9	25.9 ± 0.3	8.2 ± 0.6	74.6 ± 1.8	67.9 ± 2.5	-	-
DDGCN	39.8 ± 0.6	38.9 ± 0.7	26.3 ± 2.1	26.4 ± 3.3	24.0 ± 0.4	11.7 ± 0.7	73.6 ± 0.9	68.8 ± 1.0	-	-
DEMO-Net	56.9 ± 0.6	56.5 ± 0.7	28.3 ± 0.5	22.5 ± 2.2	28.4 ± 0.8	22.0 ± 1.3	90.8 ± 0.5	88.9 ± 0.6	83.1 ± 0.1	72.0 ± 0.4
role2vec	44.9 ± 1.6	43.8 ± 2.4	26.3 ± 0.8	27.5 ± 1.7	23.1 ± 0.1	18.3 ± 0.6	62.7 ± 0.3	56.3 ± 0.3	77.1 ± 0.2	61.5 ± 0.5
Tail-GCN	**59.2** ± 0.8	**58.5** ± 1.3	**30.2** ± 1.1	**31.1** ± 1.1	**34.9** ± 0.5	**25.2** ± 0.6	**93.6** ± 0.1	**92.7** ± 0.1	**87.0** ± 0.1	**78.2** ± 0.2

Table 3: Evaluation on tail node classification using other GNNs as the base model.

Methods	Email		Squirrel		Actor		CoauthorCS		Amazon	
	Accuracy	Micro-F	Accuracy	Micro-F	Accuracy	Micro-F	Accuracy	Micro-F	Accuracy	Micro-F
GAT	57.9 ± 0.4	57.3 ± 0.2	24.1 ± 2.4	23.1 ± 2.6	29.8 ± 0.6	13.2 ± 2.7	88.6 ± 0.2	86.2 ± 0.2	-	-
Tail-GAT	**59.4** ± 0.9	**58.2** ± 1.2	**28.8** ± 2.1	**30.4** ± 2.6	**34.5** ± 1.3	**24.7** ± 2.0	**92.5** ± 0.1	**90.8** ± 0.1	-	-
GraphSAGE	52.0 ± 1.6	51.3 ± 1.7	27.1 ± 2.7	26.4 ± 4.9	33.1 ± 1.1	23.2 ± 2.4	89.8 ± 2.4	87.7 ± 1.1	79.1 ± 0.4	62.8 ± 0.6
Tail-GraphSAGE	**55.7** ± 0.6	**54.9** ± 0.7	**28.5** ± 1.6	**28.2** ± 2.4	**34.1** ± 1.7	**26.8** ± 1.8	**93.8** ± 0.7	**92.4** ± 1.4	**85.1** ± 0.2	**75.5** ± 0.3

- **GCN as base model**
  - **DEMO-Net, role2vec**: can distinguish nodes of different degrees, not specifically for enhancing the tail nodes.
  - **SDNE, ARGA and DDGCN**: improve the robustness of graph learning, not specifically target the tail nodes.
  - **Refinement models**: in two stages, the embedding stage cannot benefit from the refinement stage.
- **GAT and GraphSAGE as base models**
  - Tail-GNN still outperforms the baselines, showing its flexibility.

Ablation study

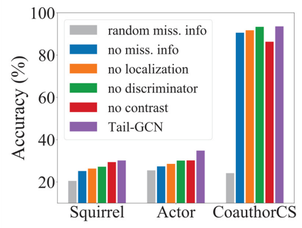


Figure 4: Ablation study.

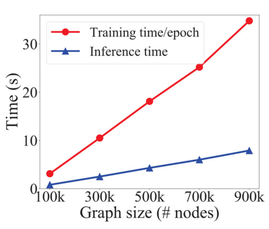


Figure 5: Scalability study.

- **Ablation study**
  - Random/no missing info: impairs the performance
  - Without localization: hurts the performance
  - Discriminator: contributes to the performance
  - Without contrastive strategy: performance becomes worse
- **Scalability**
  - Increase linearly w.r.t. graph size

Conclusions

Problem
- **Tail node embedding** in graph neural networks

Proposed model: Tail-GNN
- A new concept of **transferable neighborhood translation**
  - to capture the **relational tie** between a node and its neighboring nodes
- A novel model **Tail-GNN**
  - to narrow the gap between head and tail nodes for robust tail node embedding

Experiments
- Extensive experiments demonstrate that Tail-GNN significantly outperforms state-of-the-art models.

Reference

[1] Kipf, T. N., et al. 2017. Semi-supervised classification with graph convolutional networks. ICLR.
[2] Veličković, P., et al. 2018. Graph attention networks. ICLR.
[3] Hamilton W L., et al. 2017. Inductive representation learning on large graphs. NeurIPS.
[4] Perozzi B., et al. 2014. Deepwalk: Online learning of social representations. KDD.
[5] Lazaridou A, et al. 2017. Multimodal word meaning induction from minimal exposure to natural text. Cognitive science.
[6] Khodak M, et al. 2018. A la carte embedding: Cheap but effective induction of semantic feature vectors. ACL.
[7] Liu Z, et al. 2020. Towards locality-aware meta-learning of tail node embeddings on networks. CIKM.
[8] Wang D, et al. 2016. Structural deep network embedding. KDD.
[9] Pan S, et al. 2018. Adversarially regularized graph autoencoder for graph embedding. IJCAI.
[10] Cai R, et al. 2020. Dual-dropout graph convolutional network for predicting synthetic lethality in human cancers. Bioinformatics.
[11] Wu J, et al. 2019. Demo-Net: Degree-specific graph neural networks for node and graph classification. KDD.
[12] Ahmed N, et al. 2020. Role-based graph embeddings. TKDE.