Link Prediction Based on Graph Neural Networks (SEAL)

2024-10-11

presenter : Sooho Moon

DMAIS

Muhan Zhang, Yixin Chen
Department of CSE
Washington University in St. Louis

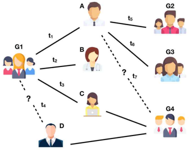
NeurIPS 2018

INDEX

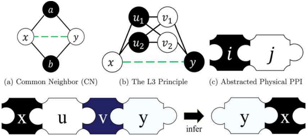
Main interest

- **Link prediction on graph networks**

  - Find whether there's a link between two nodes



▲ Social network recommendation



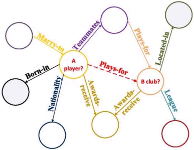
(a) Common Neighbor (CN)

(b) The L3 Principle

(c) Abstracted Physical PPI

(d) Biological Motivation of the L3 Principle

▲PPI link prediction



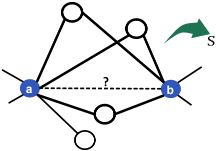
▲ Knowledge graph link prediction

Problem with previous models

▪ Previous approaches

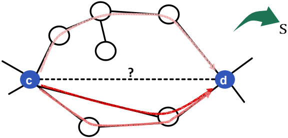
- Use hand made heuristic methods to determine links between nodes

Heuristic : A formula that defines the likelihood of a link between target nodes



$\text{score}_{CN}(\text{a,b})$

▲ Common neighbors



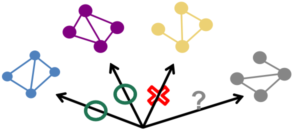
$\text{score}_{Katz}(\text{c,d})$

▲ Katz index

Problem with previous models.

- **Using hand made heuristic for link prediction is biased on data**

  - The assumptions of hand made heuristics can't be generlized to all data
  (e.g., two proteins in PPI networks that share many common neighbors are**less likely to link**)



hand made heuristics
(CN, Katz index, PageRank, SimRank etc.)

The heuristic needs to be flexible for all graph data

SEAL

***S**ubgraphs, **E**mbeddings, **A**ttibutes for **L**ink prediction*

Problem with previous models.

- **SEAL wasn't the first approach for supervised(general) heuristic**

  - WLNM(KDD, 2017) also learns from local subgraphs

  - But WLNM has **several drawbacks**

    1. trains on fix sized input GNN, thus losing structural features

    2. doesn't utilize any latent/explicit node information

    3. theoretical justifications are missing

Theoretical justification

- **Theoretical background for general heuristic**

  - Does general heuristic for link prediction exist? **Yes!**

  - Researchers proved that most high-order heuristics can be unified to **γ-decaying heuristic**

$$\mathcal{H}(x,y) = \eta \sum_{l=1}^{\infty} \gamma^l f(x,y,l)$$

$\sum_{l=1}^{\infty} \beta^l \lvert \text{walks}^{\langle l \rangle}(x,y) \rvert$	$[\pi_x]_y + [\pi_y]_x$	$\gamma \frac{\sum_{a \in \Gamma(x)} \sum_{b \in \Gamma(y)} \text{score}(a,b)}{\lvert\Gamma(x)\rvert \cdot \lvert\Gamma(y)\rvert}$	...
<Katz index>	**<PageRank>**	**<SimRank>**	**...**

Theoretical justification

- **Theoretical background for general heuristic**

$$\mathcal{H}(x,y) = \eta \sum_{l=1}^{\infty} \gamma^l f(x,y,l)$$

- γ : decaying factor between 0 and 1
- η : positive constant or function of γ that is upper bounded by a constant
- f : nonnegative function of x,y,l under the given network
- $G^h_{x,y}$: h − hop enclosing subgraph between target node x, y

As long as two properties are satisfied

- *(property 1)* $f(x,y,l) \leq \lambda^l$ *where* $\lambda < \frac{1}{\gamma}$; *and*
- *(property 2)* $f(x,y,l)$ *is calculable from* $G^h_{x,y}$ *for* $l = 1, 2, \cdots, g(h)$, *where* $g(h) = ah + b$ *with* $a, b \in \mathbb{N}$ *and* $a > 0$,

then $\mathcal{H}(x,y)$ *can be approximated from* $G^h_{x,y}$ *and the approximation error decreases at least exponentially with* h.

Theoretical justification

- **What we achieved from the general heuristic analysis**

1. γ -decaying heuristic can be generlized to all graph networks

2. γ -decaying heuristic can be approximated via enclosing subgraph

Thus our objective boils down to

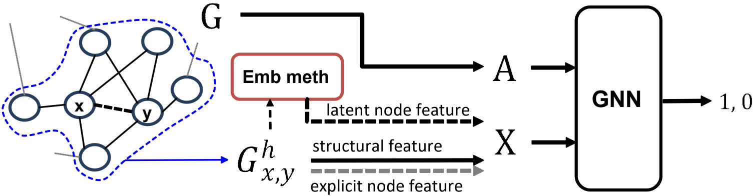
"Construct a model that could accurately calculate $f(x,y,l)$ "

Architecture

- **SEAL framework**

*node2vec and DGCNN is selected for Emb meth, GNN respectively

* node2vec pre-trained on the network with negative injection(explained later)



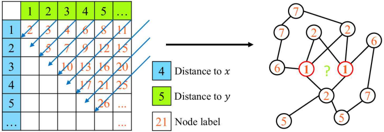
Architecture

- **SEAL framework(subgraph extraction and structural feature)**

  - Extract a h-hop subgraph surrounding the target nodes(h selected from {1, 2})

  - Label the structural role of nodes in the subgraph via **Double-Radius Node Labeling(DRNL)**

$$f_l(i) = 1 + min(d_x, d_y) + (d/2)[(d/2) + (d\%2) - 1]$$
$$d_x := d(i,x),\ d_y := d(i,y),\ d := d_x + d_y,$$



d(i,j) = shortest distance between i,j

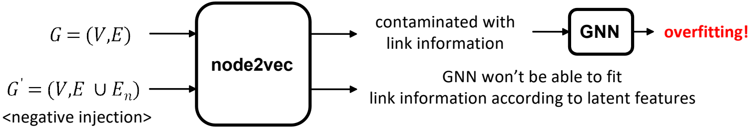
Architecture

- **SEAL framework(latent features)**

  - Latent features capture global properties and long range effects of nodes

  - Use negative injection in training node2vec to avoid overfitting in the GNN

$E_p \subseteq E, E_n \cap E = \emptyset$



Architecture

- **SEAL framework(explicit features)**

  - Not always available, but included if available

- **Example of explicit features**

  - citation networks : word distribution of document nodes

  - social networks : user's profile information

Architecture

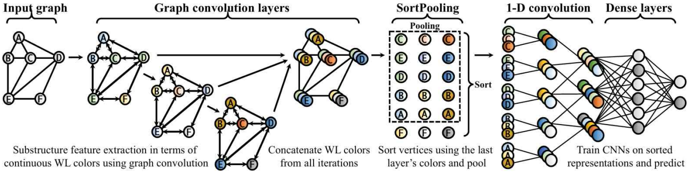
- **DGCNN for SEAL**

  - Adjacency matrix(A), node feature matrix(X) is given as input

  - SortPooling allows two things

    1. nodes with similar structural roles are likely to go to the same input layer

    2. allows fixed size input



Experiments

- **Evaluation metrics**

  - AUC, average precision(AP)

- **Datasets**

  - USAir, NS, PB, Yeast, C.ele, Power, Router, and E.coli

- **Compared methods**

  - heuristics(CN, PA, AA, RA, Katz, PR, SR, ENS)

  - latent feature methods(MF, SBM, node2vec, SPC, VGAE)

  - heuristic learning methods(WLK, WLNM)

Experiments

- **Comparison to heuristic methods**

  - Restrict SEAL to use only graph structure features

Table 1: Comparison with heuristic methods (AUC).

Data	CN	Jaccard	PA	AA	RA	Katz	PR	SR	ENS	WLK	WLNM	**SEAL**
USAir	93.80±1.22	89.79±1.61	88.84±1.45	95.06±1.03	95.77±0.92	92.88±1.42	94.67±1.08	78.89±2.31	88.96±1.44	**96.63**±0.73	95.95±1.10	**96.62**±0.72
NS	94.42±0.95	94.43±0.93	68.65±2.03	94.45±0.93	94.45±0.93	94.85±1.10	94.89±1.08	94.79±1.08	97.64±0.25	98.57±0.51	98.61±0.49	**98.85**±0.47
PB	92.04±0.35	87.41±0.39	90.14±0.45	92.36±0.34	92.46±0.37	92.92±0.35	93.54±0.41	77.08±0.80	90.15±0.45	93.83±0.59	93.49±0.47	**94.72**±0.46
Yeast	89.37±0.61	89.32±0.60	82.20±1.02	89.43±0.62	89.45±0.62	92.24±0.61	92.76±0.55	91.49±0.57	82.36±1.02	95.86±0.54	95.62±0.52	**97.91**±0.52
C.ele	85.13±1.61	80.19±1.64	74.79±2.04	86.95±1.40	87.49±1.41	86.34±1.89	**90.32**±1.49	77.07±2.00	74.94±2.04	89.72±1.67	86.18±1.72	**90.30**±1.35
Power	58.80±0.88	58.79±0.88	44.33±1.02	58.79±0.88	58.79±0.88	65.39±1.59	66.00±1.59	76.15±1.06	79.52±1.78	82.41±3.43	84.76±0.98	**87.61**±1.57
Router	56.43±0.52	56.40±0.52	47.58±1.47	56.43±0.51	56.43±0.51	38.62±1.35	38.76±1.39	37.40±1.27	47.58±1.48	87.42±2.08	94.41±0.88	**96.38**±1.45
E.coli	93.71±0.39	81.31±0.61	91.82±0.58	95.36±0.34	95.95±0.35	93.50±0.44	95.57±0.44	62.49±1.43	91.89±0.58	96.94±0.29	97.21±0.27	**97.64**±0.22

Experiments

- **Comparison to latent feature methods**

  - Structural feature + latent feature incorporated to the node information matrix

  - Interestingly, joint learning is not always good(e.g., Power)

Table 2: Comparison with latent feature methods (AUC).

Data	MF	SBM	N2V	LINE	SPC	VGAE	**SEAL**
USAir	94.08±0.80	94.85±1.14	91.44±1.78	81.47±10.71	74.22±3.11	89.28±1.99	**97.09**±0.70
NS	74.55±4.34	92.30±2.26	91.52±1.28	80.63±1.90	89.94±2.39	94.04±1.64	**97.71**±0.93
PB	94.30±0.53	93.90±0.42	85.79±0.78	76.95±2.76	83.96±0.86	90.70±0.53	**95.01**±0.34
Yeast	90.28±0.69	91.41±0.60	93.67±0.46	87.45±3.33	93.25±0.40	93.88±0.21	**97.20**±0.64
C.ele	85.90±1.74	86.48±2.60	84.11±1.27	69.21±3.14	51.90±2.57	81.80±2.18	**89.54**±2.04
Power	50.63±1.10	66.57±2.05	76.22±0.92	55.63±1.47	**91.78**±0.61	71.20±1.65	84.18±1.82
Router	78.03±1.63	85.65±1.93	65.46±0.86	67.15±2.10	68.79±2.42	61.51±1.22	**95.68**±1.22
E.coli	93.76±0.56	93.82±0.41	90.82±1.49	82.38±2.19	94.92±0.32	90.81±0.63	**97.22**±0.28

Experiments

- **Incorporating explicit features**

  - VGAE and SEAL are the only methods that incorporate explicit features

  - Structural feature + latent feature + explicit feature

Table 12: Comparison with network embedding methods (AUC and standard deviation, OOM: out of memory).

	N2V	LINE	SPC	VGAE	WLNM	SEAL
arXiv	96.18±0.40	84.64±0.03	87.00±0.14	OOM	99.19±0.03	**99.40**±0.14
Facebook	99.05±0.07	89.63±0.06	98.59±0.11	98.21±0.22	99.24±0.03	**99.40**±0.08
BlogCatalog	85.97±1.56	90.92±2.05	96.74±0.31	OOM	96.55±0.08	**98.10**±0.60
Wikipedia	76.59±2.06	74.44±0.66	99.54±0.04	89.74±0.18	99.05±0.03	**99.63**±0.05
PPI	70.31±0.79	72.82±1.53	92.27±0.22	85.86±0.43	88.79±0.38	**93.52**±0.37

Conclusion

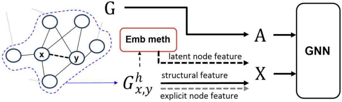
- **Supervised learning link prediction model SEAL**

  - Use **S**ubgraphs, **E**mbeddings, **A**ttributes for **L**ink prediction

  - Provided **theoretical justification**

  - Proposed DRNL to **label structural roles** of nodes

  - Utilized DGCNN to **preserve the structural information**



- **Inspiration to other researchs**

  - Opened new directions for knowledge graph completion and recommender systems etc.

Thank You!

Contact: Sooho Moon (Email: moonwalk725@cau.ac.kr)