sigir21

Self-supervised Graph Learning for Recommendation

Jiancan Wu
Xiangnan He

Xiang Wang
Fuli Feng

Liang Chen

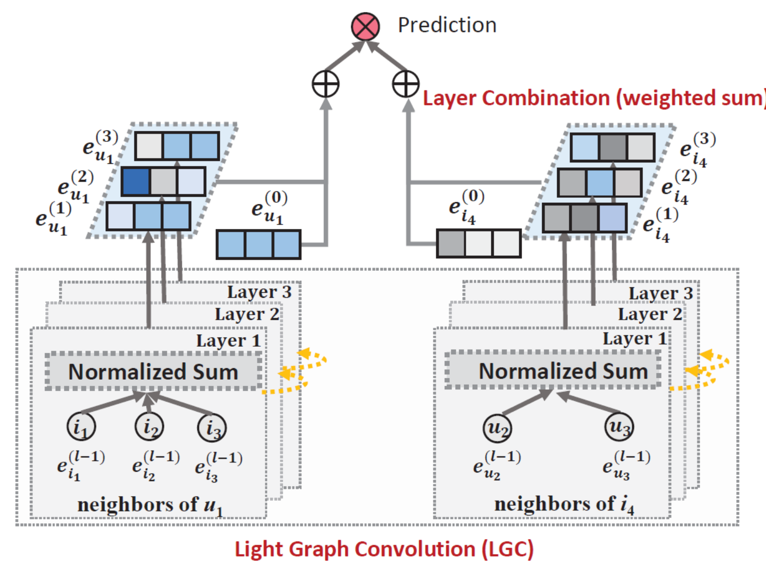
Jianxun Lian
Xing Xie

Outline

- Background
- Model: SGL
- Experiments
- Conclusion & Future Work

Recap GCN for CF

Abstract Paradigm



- Neighbor Aggregation

  (1) Representation aggregation layers

$$\mathbf{Z}^{(l)} = H\left(\mathbf{Z}^{(l-1)}, \mathcal{G}\right)$$

$$\mathbf{a}_u^{(l)} = f_{\text{aggregate}}\left(\{\mathbf{z}_i^{(l-1)} | i \in \mathcal{N}_u\}\right),$$

$$\mathbf{z}_u^{(l)} = f_{\text{combine}}(\mathbf{z}_u^{(l-1)}, \mathbf{a}_u^{(l)}),$$

  (2) Readout layer

$$\mathbf{z}_u = f_{\text{readout}}\left(\{\mathbf{z}_u^{(l)} | l = [0, \cdots, L]\}\right)$$

- Supervised Learning Loss

$$\mathcal{L}_{main} = \sum_{(u,i,j)\in O} -\log \sigma(\hat{y}_{ui} - \hat{y}_{uj}) \qquad \hat{y}_{ui} = \mathbf{z}_u^{\top}\mathbf{z}_i$$

He et al. LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. SIGIR 2020

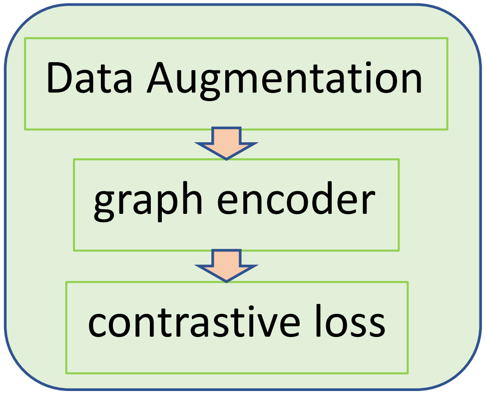
Limitations

Limitations in existing GCNs:

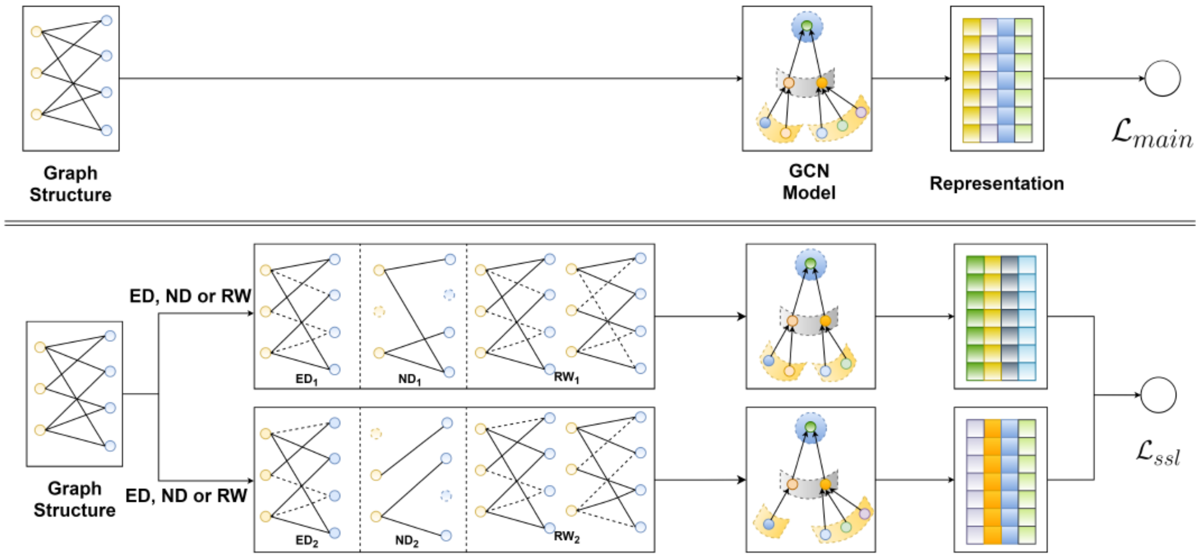
- **Sparse Supervision Signal**
  - The supervision signal comes from the observed interactions ➔ extremely sparse
- **Skewed Data Distribution**
  - Power-law distribution
  - High-degree items exert larger impact on the representation learning
- **Noises in Interactions**
  - Implicit feedback makes the learning more vulnerable to interaction noises

Self-supervised Learning

- Obtain “labels” from the data itself
- Predict part of the data from other parts



SGL framework



Supervised Task

+

SSL Task

Augmentation on graph structure:

- The features of users and items are discrete
- Users and items in the graph are inherently connected and dependent on each other

Wu et al. Self-supervised Graph Learning for Recommendation. SIGIR 2021

Data Augmentations

$$Z_1^{(l)} = H\left(Z_1^{(l-1)}, s_1(\mathcal{G})\right), \quad Z_2^{(l)} = H\left(Z_2^{(l-1)}, s_2(\mathcal{G})\right), \quad s_1, s_2 \sim \mathcal{S}$$

- Node Dropout (ND)

$$s_1(\mathcal{G}) = (M^{'} \odot \mathcal{V}, \mathcal{E}), \quad s_2(\mathcal{G}) = (M^{''} \odot \mathcal{V}, \mathcal{E}) \quad M^{'}, M^{''} \in \{0,1\}^{|\mathcal{V}|}$$

  - **Identify the influential nodes** from differently augmented views
  - Make the representation learning less sensitive to structure changes

- Edge Dropout (ED)

$$s_1(\mathcal{G}) = (\mathcal{V}, M^{'} \odot \mathcal{E}), \quad s_2(\mathcal{G}) = (\mathcal{V}, M^{''} \odot \mathcal{E}) \quad M^{'}, M^{''} \in \{0,1\}^{|\mathcal{E}|}$$

  - Capture the useful patterns of the **local structures of a node**
  - Endow the representations more robustness against the presence of single interactions, especially the noisy interactions.

- Random Walk (RW)

$$s_1(\mathcal{G}) = (\mathcal{V}, M_1^{(l)} \odot \mathcal{E}), \quad s_2(\mathcal{G}) = (\mathcal{V}, M_2^{(l)} \odot \mathcal{E}) \quad M_1^{(l)}, M_2^{(l)} \in \{0,1\}^{|\mathcal{E}|}$$

  - constructing an **individual subgraph for each node** with random walk
  - Layer-sensitive local structure

Objective Function

- Contrastive Loss --- **InfoNCE**
  - maximize the agreement of positive pairs
  - minimize that of negative pairs

$$\mathcal{L}_{ssl}^{user} = \sum_{u \in \mathcal{U}} -\log \frac{\exp(s(\mathbf{z}'_u, \mathbf{z}''_u)/\tau)}{\sum_{v \in \mathcal{U}} \exp(s(\mathbf{z}'_u, \mathbf{z}''_v)/\tau)}$$

$$\mathcal{L}_{ssl} = \mathcal{L}_{ssl}^{user} + \mathcal{L}_{ssl}^{item}$$

- Supervised Loss --- BPR

$$\mathcal{L}_{main} = \sum_{(u,i,j) \in \mathcal{O}} -\log \sigma\left(\hat{y}_{ui} - \hat{y}_{uj}\right)$$

- Multi-task Training

$$\mathcal{L} = \mathcal{L}_{main} + \lambda_1 \mathcal{L}_{ssl} + \lambda_2 \left\|\Theta\right\|_2^2$$

Hard Negative Mining

- Gradient of the self-supervised

$$\frac{\partial \mathcal{L}_{ssl}^{user}(u)}{\partial z_u'} = \frac{1}{\tau \left\| z_u' \right\|} \left\{ c(u) + \sum_{v \in \mathcal{U} \setminus \{u\}} c(v) \right\}$$

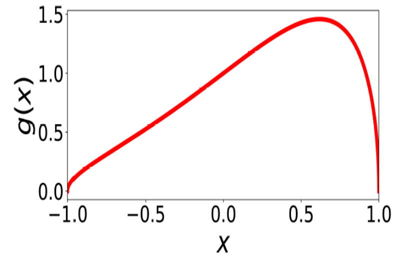
$$P_{uv} = \frac{\exp(s_u'^T s_v''/\tau)}{\sum_{v \in \mathcal{U}} \exp(s_u'^T s_v''/\tau)};\ s_u' = \frac{z_u'}{\|z_u'\|} \text{ and } s_u'' = \frac{z_u''}{\|z_u''\|}$$

$$c(u) = \left(s_u'' - (s_u'^T s_u'') s_u' \right) (P_{uu} - 1),$$

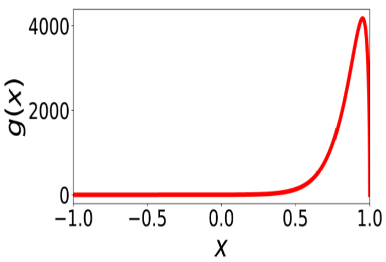
$$\boxed{c(v) = \left(s_v'' - (s_u'^T s_v'') s_u' \right) P_{uv},}$$ Contribution from negative sample

L_2 norm of $c(v)$ $\quad \|c(v)\|_2 \propto \sqrt{1 - (s_u'^T s_v'')^2} \exp(s_u'^T s_v''/\tau)$

$g(x) = \sqrt{1 - x^2} \exp\left(\frac{x}{\tau}\right)$ $\quad$ x is the cosine similarity between s_u' and s_v''



(a) $g(x)$, $\tau = 1$



(b) $g(x)$, $\tau = 0.1$

$$\begin{cases} -1 \le x < 0 & \text{Easy negative} \\ 0 < x \le 1 & \text{Hard negative} \end{cases}$$

→offer much larger gradients to guide the optimization

Wu et al. Self-supervised Graph Learning for Recommendation. SIGIR 2021

Experiment Settings

- **Datasets:**
  - Yelp2018, Amazon-Book, Alibaba-iFashion
- **Evaluation Metrics:**
  - recall@20, ndcg@20
- **Dataset partition:** randomly select 80% data for training set, and 20% data for testing set.

Table 2: Statistics of the datasets.

Dataset	#Users	#Items	#Interactions	Density
Yelp2018	31,668	38,048	1,561,406	0.00130
Amazon-Book	52,643	91,599	2,984,108	0.00062
Alibaba-iFashion	300,000	81,614	1,607,813	0.00007

Wu et al. Self-supervised Graph Learning for Recommendation. SIGIR 2021

Experiment Results

Dataset	Yelp2018		Amazon-Book		Alibaba-iFashion	
Method	Recall	NDCG	Recall	NDCG	Recall	NDCG
NGCF	0.0579	0.0477	0.0344	0.0263	0.1043	0.0486
LightGCN	0.0639	0.0525	0.0411	0.0315	0.1078	0.0507
Mult-VAE	0.0584	0.0450	0.0407	0.0315	0.1041	0.0497
DNN+SSL	0.0483	0.0382	0.0438	0.0337	0.0712	0.0325
SGL-ED	**0.0675**	**0.0555**	**0.0478**	**0.0379**	**0.1126**	**0.0538**
%Improv.	5.63%	5.71%	9.13%	12.46%	4.45%	6.11%
p-value	5.92e-8	1.89e-8	5.07e-10	3.63e-10	3.34e-8	4.68e-10

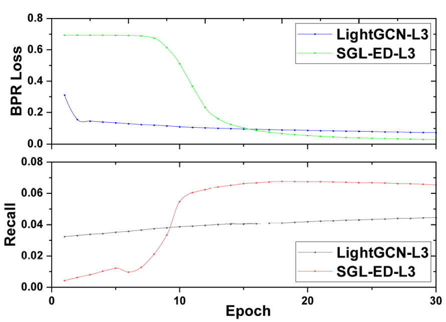
✓ SGL achieves significant improvements over the state-of-the-art baselines → **outstanding performance**

Experiment Results

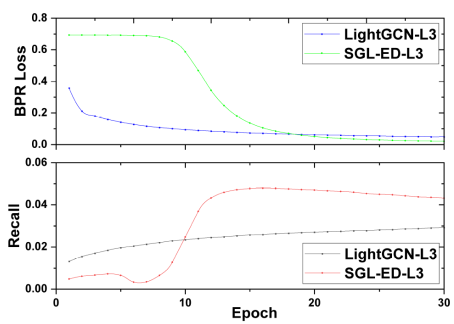
- Performance comparison among different SGL implementations and LightGCN at different layers :

Datase		Yelp2018		Amazon-Book		Alibaba-iFashion	
#Layer	Method	Recall	NDCG	Recall	NDCG	Recall	NDCG
1 Layer	LightGCN	0.0631	0.0515	0.0384	0.0298	0.0990	0.0454
	SGL-ND	**0.0643(+1.9%)**	**0.0529(+2.7%)**	0.0432(+12.5%)	0.0334(+12.1%)	**0.1133(+14.4%)**	**0.0539(+18.7%)**
	SGL-ED	0.0637(+1.0%)	0.0526(+2.1%)	**0.0451(+17.4%)**	**0.0353(+18.5%)**	0.1125(+13.6%)	0.0536(+18.1%)
	SGL-RW	0.0637(+1.0%)	0.0526(+2.1%)	**0.0451(+17.4%)**	**0.0353(+18.5%)**	0.1125(+13.6%)	0.0536(+18.1%)
2 Layers	LightGCN	0.0622	0.0504	0.0411	0.0315	0.1066	0.0505
	SGL-ND	0.0658(+5.8%)	0.0538(+6.7%)	0.0427(+3.9%)	0.0335(+6.3%)	**0.1106(+3.8%)**	**0.0526(+4.2%)**
	SGL-ED	**0.0668(+7.4%)**	**0.0549(+8.9%)**	**0.0468(+13.9%)**	**0.0371(+17.8%)**	0.1091(+2.3%)	0.0520(+3.0%)
	SGL-RW	0.0644(+3.5%)	0.0530(+5.2%)	0.0453(+10.2%)	0.0358(+13.7%)	0.1091(+2.3%)	0.0521(+3.2%)
3 Layers	LightGCN	0.0639	0.0525	0.0410	0.0318	0.1078	0.0507
	SGL-ND	0.0644(+0.8%)	0.0528(+0.6%)	0.0440(+7.3%)	0.0346(+8.8%)	0.1126(4.5%)	0.0536(+5.7%)
	SGL-ED	**0.0675(+5.6%)**	**0.0555(+5.7%)**	**0.0478(+16.6%)**	**0.0379(+19.2%)**	0.1126(+4.5%)	0.0538(+6.1%)
	SGL-RW	0.0667(+4.4%)	0.0547(+4.2%)	0.0457(+11.5%)	0.0356(+12.0%)	**0.1139(+5.7%)**	**0.0539(+6.3%)**

- Training curves of SGL-ED and LightGCN :



(a) Yelp2018-L3

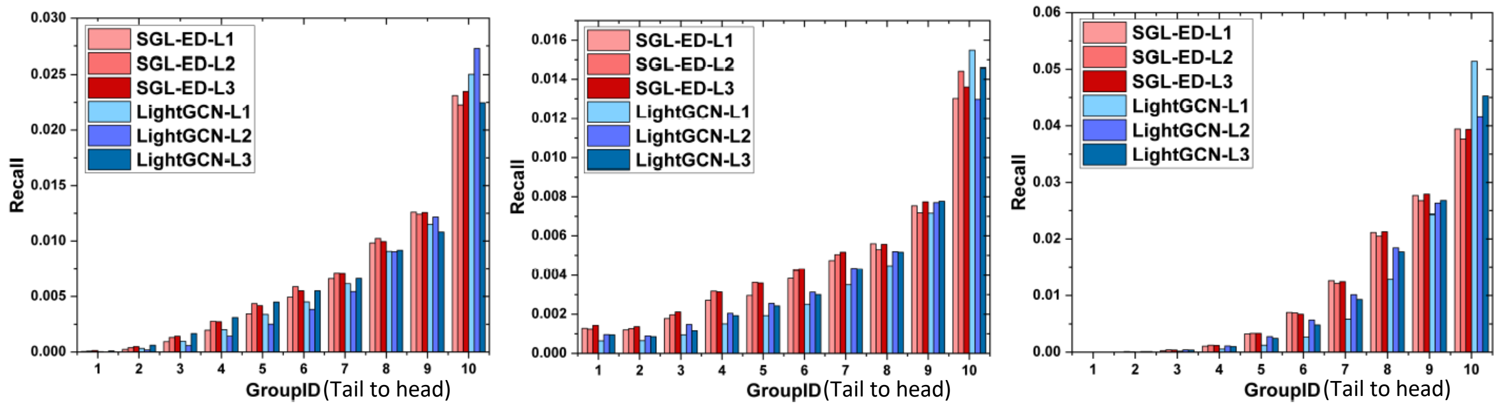


(b) Amazon-Book-L3

✓ InfoNCE vs. BPR
✓ Hard negative mining

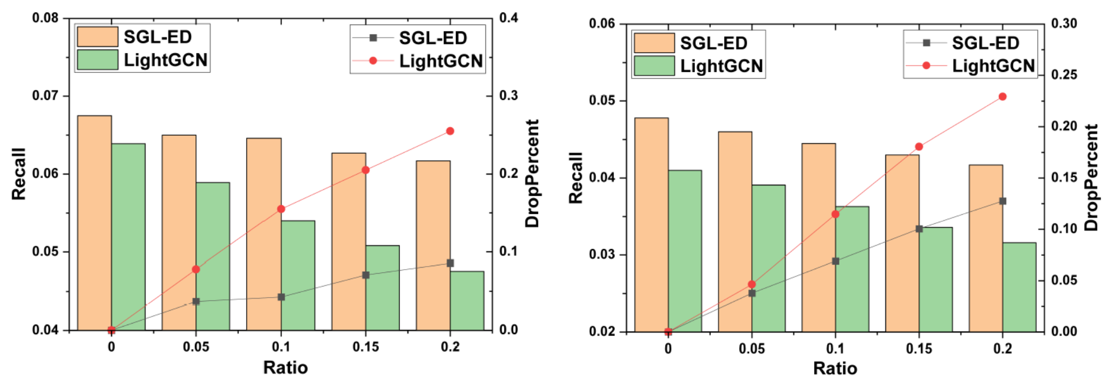
Benefits of SGL

- Long-tail Recommendation



(a) Yelp2018 (b) Amazon-Book (c) Alibaba-iFashion

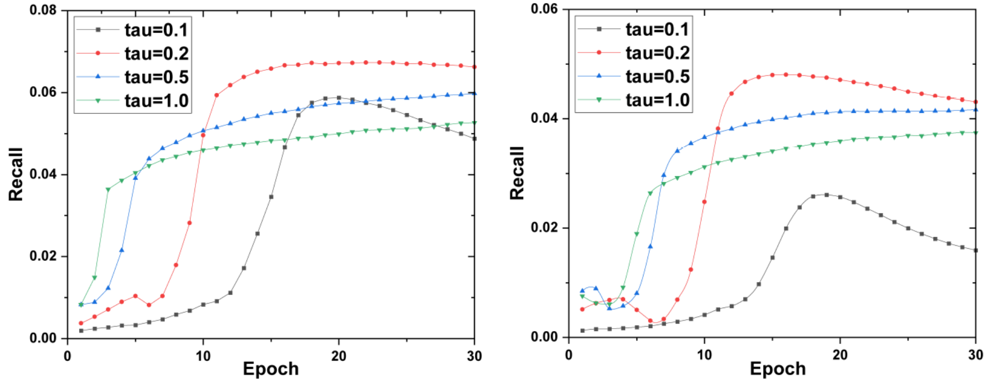
- Robustness to Noisy Interactions



(a) Yelp2018 (b) Amazon-Book

Study of SGL

- Effect of Temperature



(a) Yelp2018 (b) Amazon-Book

- Effect of Negatives and Pretrain

Dataset	Yelp2018		Amazon-Book	
Method	Recall	NDCG	Recall	NDCG
SGL-ED-batch	0.0670	0.0549	0.0472	0.0374
SGL-ED-merge	0.0671	0.0547	0.0464	0.0368
SGL-pre	0.0653	0.0533	0.0429	0.0333
SGL-ED	**0.0675**	**0.0555**	**0.0478**	**0.0379**

Conclusion & Future Work

✓Conclusion

- A model-agnostic framework SGL to supplement the supervised recommendation task with self-supervised learning on user-item graph
- Devise three types of data augmentation from different aspects to construct the auxiliary contrastive task
- Prove in theory that SGL inherently encourages learning from hard negatives

✓Future Work

- Explore new perspectives, such as counterfactual learning to identify influential data points
- Pre-training and fine-tuning in recommendation?
- Fulfill the potential of SSL to address the long-tail issue

Thanks & QA?

- The code is available at https://github.com/wujcan/SGL