HCL: Hybrid Contrastive Learning for Graph-based Recommendation

Paper ID: 176

Xiyao Ma, Zheng Gao, Qian Hu, Mohamed AbdelHady

Amazon Alexa AI

Introduction

we propose Hybrid Contrastive Learning (HCL) for graph-based recommendation that integrates unsupervised and supervised contrastive learning. To summarize, the contributions of this work are three-folds:

- We identify the limitation of existing contrastive learning methods for recommendation and propose Hybrid Contrastive Learning.
- We generalize a permutational approach that performs hybrid contrastive learning across multiple views which are generated to convey incomplete and noisy information with respect to node embeddings and topology.
- Extensive experiments show the superiority of HCL regarding to model generalization and robustness over SOTA baselines on two public and one internal dataset.

Preliminary

LightGCN is a strong graph collaborative filtering (GCF) baseline for recommendation that captures the high-order connectivity from the user-item bipartite graph, and the model is trained in a supervised learning paradigm. It is applied on the user-item bipartite graph to learn user and item representations by aggregating the representation of its direct neighbors N and with the defined graph convolution operations:

$$e_u^l = \sum_{i \in \mathcal{N}_u} \frac{1}{\sqrt{|\mathcal{N}_u|}\sqrt{|\mathcal{N}_i|}} e_i^{l-1},$$
$$e_i^l = \sum_{u \in \mathcal{N}_i} \frac{1}{\sqrt{|\mathcal{N}_i|}\sqrt{|\mathcal{N}_u|}} e_u^{l-1}.$$

$$e = avgpooling\left(\{e^l \mid l = [0, \cdots, L]\}\right)$$

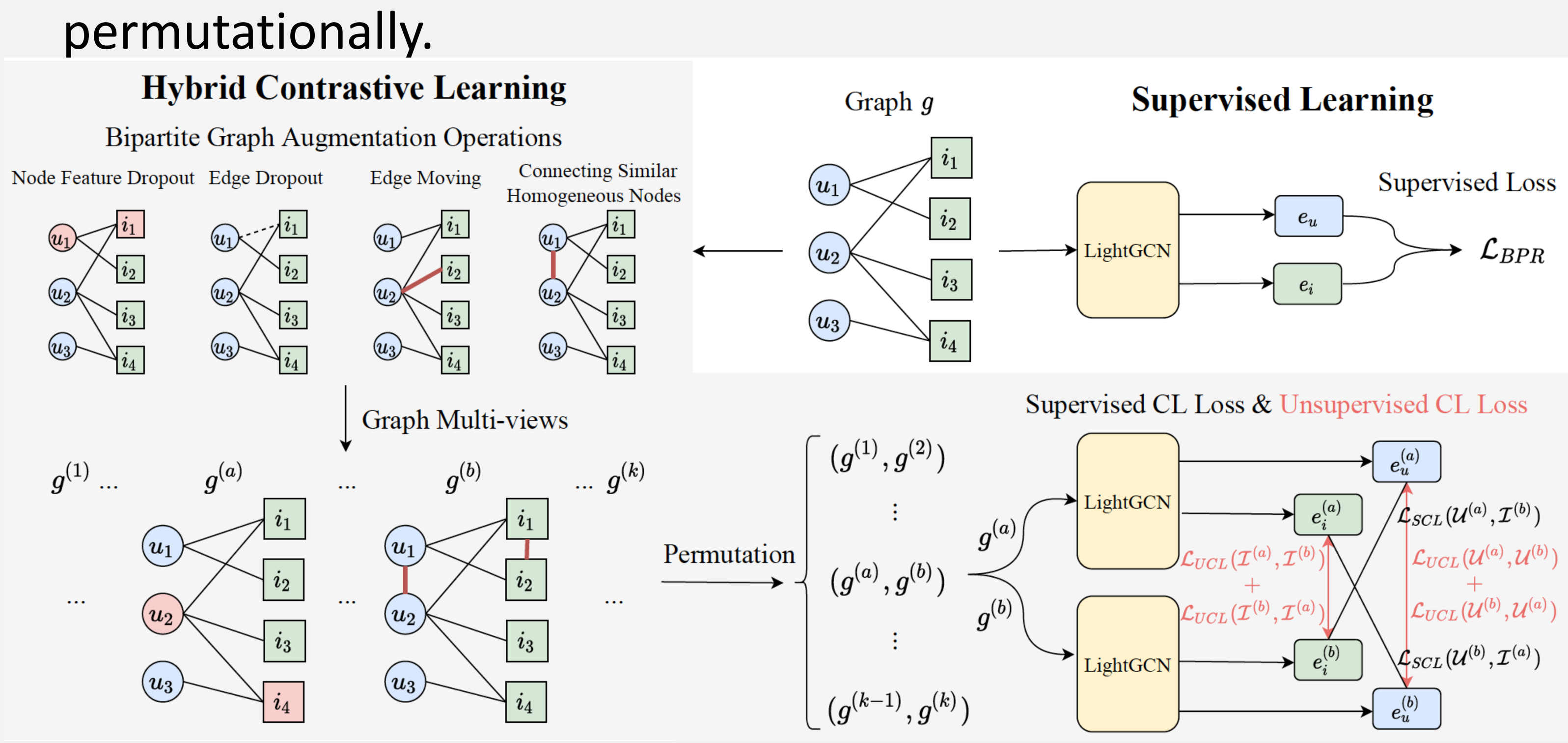
Bayesian Personalized Ranking (BPR) finally assigns higher probability to observed interactions than its unobserved interactions:

$$\mathcal{L}_{\mathrm{BPR}} = \sum_{(u,i) \in \mathcal{Y}, (u,j) \notin \mathcal{Y}} -\log \sigma\left(\hat{y}_{ui} - \hat{y}_{uj}\right),$$
$$\hat{y}_{ui} = e_u^\top e_i.$$

Method: HCL

In general, the proposed HCL has three steps:

- We propose novel bipartite graph augmentation strategies by taking node embeddings and topology into consideration to generate different incomplete and noisy views for the input user-item graph.
- The proposed hybrid contrastive learning performs unsupervised and supervised contrastive learning on homogeneous nodes and observed user-item interactions, respectively.
- We conduct the hybrid contrastive learning among multiple views permutationally.



Bipartite Graph Augmentation

we propose four bipartite graph augmentation strategies to generate different graph views that contain incomplete and noisy information about node embedding and node topology to boost downstream contrastive learning, including node embedding dropout, edge dropout, edge moving and connecting similar homogeneous nodes.

Hybrid Contrastive Learning

Unsupervised Contrastive Learning: We pull together the different views of the same node and push apart those of different nodes.

$$\mathcal{L}_{UCL}(\mathcal{U}^{(a)}, \mathcal{U}^{(b)}) = \sum_{u \in \mathcal{U}} -\log \frac{\exp\left(f\left(e_u^{(a)}, e_u^{(b)}\right)/\tau\right)}{\sum_{v \in \mathcal{U}} \exp\left(f\left(e_u^{(a)}, e_v^{(b)}\right)/\tau\right)}$$

Supervised Contrastive Learning: We propose to encourage the consistency of the embeddings of the users and the interacted items by computing supervised contrastive learning (SCL) loss given the observed user-item interactions. We maximize agreement between user representation and item representation generated from different views.

$$\mathcal{L}_{SCL}(\mathcal{U}^{(b)}, \mathcal{I}^{(a)}) = \sum_{(u,i) \in \mathcal{Y}} -\log \frac{\exp\left(f\left(e_u^{(b)}, e_i^{(a)}\right)/\tau\right)}{\sum_{q \in \mathcal{Q}} \exp\left(f\left(e_u^{(b)}, e_q^{(a)}\right)/\tau\right)}$$

Multi-view Permutation

The total multi-view HCL loss is the summation of the HCL loss terms computed on every pair of graph views permutationally, and each HCL loss term is the summation of unsupervised contrastive learning losses on user and item nodes and supervised contrastive learning losses.

$$\mathcal{L}_{HCL}^{multi-view} = \sum_{a,b} \mathcal{L}_{HCL}(g^{(a)}, g^{(b)}),$$

$$\begin{aligned} \mathcal{L}_{HCL}(g^{(a)}, g^{(b)}) = & \mathcal{L}_{UCL}(\mathcal{U}^{(a)}, \mathcal{U}^{(b)}) + \mathcal{L}_{UCL}(\mathcal{U}^{(b)}, \mathcal{U}^{(a)}) \\ & + \mathcal{L}_{UCL}(\mathcal{I}^{(a)}, \mathcal{I}^{(b)}) + \mathcal{L}_{UCL}(\mathcal{I}^{(b)}, \mathcal{I}^{(a)}) \\ & + \mathcal{L}_{SCL}(\mathcal{U}^{(a)}, \mathcal{I}^{(b)}) + \mathcal{L}_{SCL}(\mathcal{U}^{(b)}, \mathcal{I}^{(a)}). \end{aligned}$$

We trained the model in the multi-task learning fashion with the final loss

$$\mathcal{L}_{final} = \mathcal{L}_{BPR} + \lambda_1 \mathcal{L}_{HCL}^{multi-view} + \lambda_2 \|\Theta\|_2^2$$

Experiments

Datasets

We adopt two widely-used public datasets, Yelp2018 and Amazon-book, and one internal Alexa Recipe dataset to evaluate model performances across the experiments.

Baselines

We mainly adopt three categories of models as baselines for performance comparison: Non-GCF models (MF, NCF), GCF models (NGCF, LightGCN), and GCF model with contrastive learning (SGL).

Experiments

Model Performance Comparison on Public Datasets

Datasets		Yelp2018				Amazon-book			
Category	Models	Precision@20	Recall@20	HitRate@20	NDCG@20	Precision@20	Recall@20	HitRate@20	NDCG@20
Non-GCF	MF-BPR [7]	0.0223	0.0491	0.3224	0.0394	0.0119	0.0285	0.1801	0.0221
	NCF [14]	0.0203	0.0441	0.3	0.0357	0.0097	0.023	0.1532	0.0174
GCF	NGCF [3]	0.0229	0.0511	0.3323	0.0417	0.012	0.0294	0.182	0.0221
	LightGCN [4]	0.0259	0.0575	0.361	0.047	0.0143	0.0356	0.2134	0.027
GCF+CL	SGL [5]	0.0262	0.0581	0.366	0.0476	0.0147	0.0367	0.2176	0.0282
	HCL (k=2) [5]	0.0281	0.063	0.3846	0.0516	0.0162	0.0387	0.2247	0.0297
	HCL (k=3)	**0.0287**	**0.0643**	**0.3935**	**0.0525**	**0.0166**	**0.0395**	**0.2304**	**0.0307**
	Improvement (%)	9.5	10.7	7.5	10.3	12.9	7.6	5.9	8.9

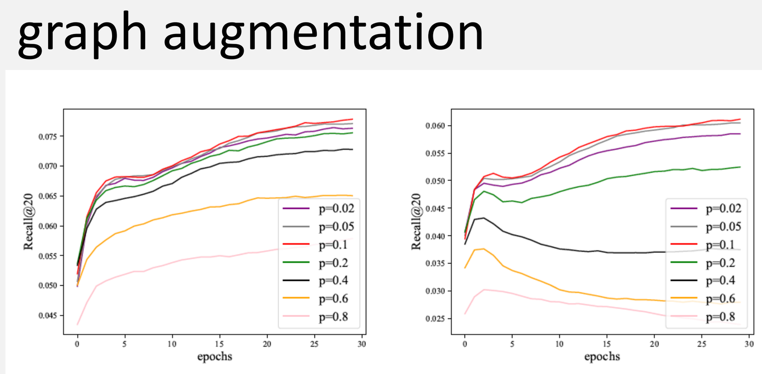
Model Relative Performance Comparison on Recipe Dataset with SGL.

	Recipe			
Models	Precision@20	Recall@20	HitRate@20	NDCG@20
MF	+2.2%	+4.9%	+4.8%	+2.7%
NCF	+45.6%	+10.5%	+4.0%	+6.7%
NGCF	-2.2%	-46.0%	-50.2%	-55%
LightGCN	+23.9%	+9.4%	+3.9%	+5.5%
SGL	+0.0%	+0.00%	+0.00%	+0.0%
HCL (k=2)	+50%	+15.4%	+6.4%	10.2%
HCL (k=3)	**+54.3%**	**+16.3%**	**+9.1%**	**15.6%**

Ablation Study

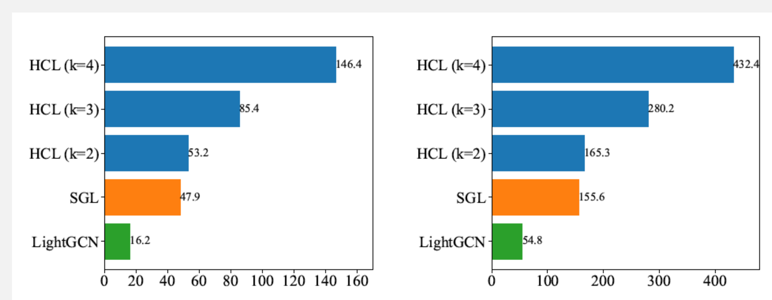
Datasets	Yelp2018		Amazon-book	
Methods	Recall@20	NDCG@20	Recall@20	NDCG@20
HCL	0.0643	0.0525	0.0395	0.0307
- DA	0.063 (-2.0%)	0.0510 (-2.8%)	0.0385 (-2.5%)	0.0298 (-3.0%)
- SCL	**0.0623 (-3.1%)**	**0.0506 (-3.6%)**	**0.0381 (-3.5%)**	**0.0292 (-4.9%)**
- MV	0.063 (-2.0%)	0.0516 (-1.7%)	0.0387 (-2.0%)	0.0297 (-3.3%)
- PL	0.0627 (-2.5%)	0.0508 (-3.2%)	0.0385 (-2.5%)	0.0296 (-3.6%)

Effect of probability for bipartite graph augmentation



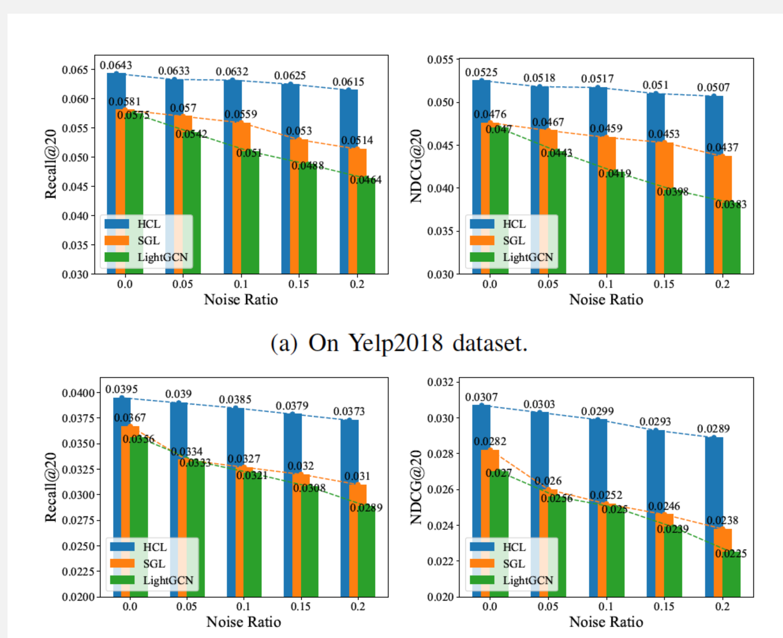
(a) Yelp2018 (b) Amazon-book

Time (seconds) of training models for one epoch.



(a) Yelp2018 (b) Amazon-book

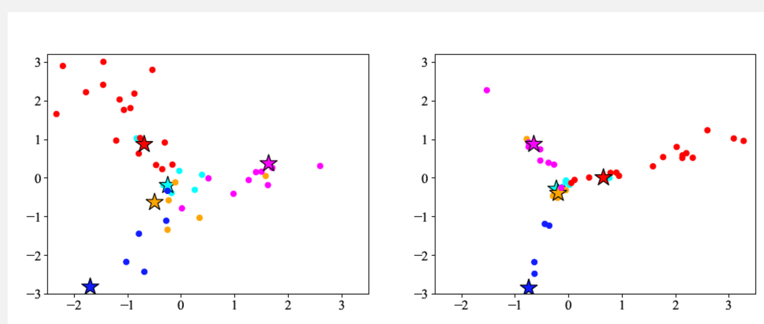
Model performance with noise ratios. The bar represents the results in terms of Recall@20 and NDCG@20, respectively



(a) On Yelp2018 dataset.

(b) On Amazon-book dataset.

Visualization of users and items embeddings learnt by SGL and the proposed HCL.



(a) SGL (b) HCL