Learning Contrastive Multi-View Graphs for Recommendation (Student Abstract)

Zhangtao Cheng[1], **Ting Zhong**[1*], **Kunpeng Zhang**[2], **Joojo Walker**[1], **Fan Zhou**[1]

[1] University of Electronic Science and Technology of China
[2] University of Maryland, College park
zhangtao980107@outlook.com, zhongting@uestc.edu.cn, kpzhang@umd.edu, joojokojododzi@gmail.com,
fan.zhou@uestc.edu.cn,

Abstract

This paper exploits self-supervised learning (SSL) to learn more accurate and robust representations from the user-item interaction graph. Particularly, we propose a novel SSL model that effectively leverages contrastive multi-view learning and pseudo-siamese network to construct a pre-training and post-training framework. Moreover, we present three graph augmentation techniques during the pre-training stage and explore the effects of combining different augmentations, which allow us to learn general and robust representations for the GNN-based recommendation. Simple experimental evaluations on real-world datasets show that the proposed solution significantly improves the recommendation accuracy, especially for sparse data, and is also noise resistant.

Introduction

More recently, graph neural networks (GNNs)(Wu et al. 2020) are widely adpoted in the recomender system and result in the state-of-the-art recommendation performance (He et al. 2020). However, GNNs and their variants used in recommendation tasks are confronted with several limitations (Wu et al. 2021): (1) *Data sparsity*: The observed interactions are extremely sparse w.r.t. the entire space, making it difficult for GNN models to learn high-quality representations; (2) *Noisy interactions*: Observed interactions like views and clicks are usually noisy, e.g., a user clicks a item by mistake. Unfortunately, GNNs are based on neighborhood aggregation schemes that may exaggerate the impact of noisy interactions in representation learning.

We alleviate these limitations by leveraging self-supervised signals extracted from the user-item graph to improve representation learning. Specifically, we propose a novel framework called Contrastive Multi-view Pre-training and Pseudo-Siamese Post-training framework (CMP-PSP) for graph-based recommendation. Firstly, CMP-PSP uses contrastive learning techniques (Chen et al. 2020) to pre-train general user and item embeddings representing each user/item's unique attribute. We propose to sub-sample and augment the subgraphs, and consider three data augmentation strategies, including node perturbation (it randomly throws away a certain part of vertices along with their connections), edge noising (it adds and drops a certain ratio of edges), and embedding masking (it masks a certain part of subgraph's ID embeddings), to transform the user-item subgraph into different views. With the debiased contrastive loss, we optimize the model parameters to maximize the similarity between representations of the same node from different augmented views. In the post-training stage, we adopt a pseudo-siamese network to augment representations for downstream recommendation tasks.

Methodology

Contrastive Multi-view Pre-Training (CMP): It consists of the following three components: (1) an augmentation mechanism that transforms a sampled subgraph into two correlated views of the same subgraph; (2) two dedicated subgraph GCNs as the graph encoders to obtain node representation $\boldsymbol{h} = GCN(\tilde{\mathcal{G}}, \tilde{\mathcal{E}})$. $\tilde{\mathcal{E}}$ denotes embeddings of the corresponding nodes in the subgraph $\tilde{\mathcal{G}}$. They are followed by a shared MLP. We use MLP with one hidden layer to obtain $\boldsymbol{z} = W^{(2)}\sigma(W^{(1)}\boldsymbol{h})$; and (3) the debiased contrastive loss that determines whether the two representations are derived from the same data points. Specifically, we randomly sample a subgraph from the original graph $\mathcal{G}$. Then we apply devised augmentation operations to this subgraph to obtain two different augmented views, containing $2N$ users and $2M$ items. Following (Chen et al. 2020), for each user u, we treat the same user u in the different views as the positive pairs $(\boldsymbol{z}_u^i, \boldsymbol{z}_u^j)$, and treat other $2(N-1)$ users as the negative pairs. Then the loss function for a positive pair $(\boldsymbol{z}_u^i, \boldsymbol{z}_u^j)$ is defined as:

$$\mathcal{L}_{user} = -\log \frac{\exp(\mathrm{s}(\boldsymbol{z}_u^i, \boldsymbol{z}_u^j)/\tau)}{\sum_{k=1}^{2N} \mathbf{1}_{[k \neq i,j]} \exp(\mathrm{s}(\boldsymbol{z}_u^i, \boldsymbol{z}_u^k)/\tau)}, \quad (1)$$

where $\mathrm{s}(\cdot)$ measures the similarity between two representations and is set as $\mathrm{s}(\boldsymbol{u}, \boldsymbol{v}) = \boldsymbol{cos}(\boldsymbol{u}, \boldsymbol{v}) = \boldsymbol{u}^\top \boldsymbol{v}/\|\boldsymbol{u}\|\|\boldsymbol{v}\|$, $\mathbf{1}_{[k \neq i]} \in \{0, 1\}$ is an indicator function, and τ is the temperature parameter. Analogously, we obtain the contrastive loss of the item side $\mathcal{L}_{item}$.

Here, we also consider the relation between a user and an item in the representation process. Generally, items that have not been interacted with are treated as the negative samples. However, there may exist false negative items; that is,

*corresponding author.

some negative items might be potential positive items that the user likes. To alleviate this problem, we leverage the debiased contrastive loss (Chuang et al. 2020) to approximate the distribution of negative examples. For each user u, the loss needs to be optimized becomes:

$$\mathcal{L}_{user}^{item} = \sum_{l=1}^{N} -\log \frac{\exp(\mathrm{s}(\boldsymbol{z}_u^l, \boldsymbol{z}_i^j)/\tau)}{\exp(\mathrm{s}(\boldsymbol{z}_u^l, \boldsymbol{z}_i^j)/\tau) + Lg'},$$

$$g\left(\boldsymbol{z}_u^l, \boldsymbol{z}_i^j, \{\boldsymbol{z}_i^k\}_{k=1}^L\right) = \frac{1}{1-\tau^+}\left(\frac{1}{L}\sum_{k=1}^{L}(\mathcal{N}) - \tau^+ L\mathcal{P}\right),$$

$$g'\left(\boldsymbol{z}_u^l, \boldsymbol{z}_i^j, \{\boldsymbol{z}_i^k\}_{k=1}^L\right) = \max\left\{g\left(\boldsymbol{z}_u^l, \boldsymbol{z}_i^j, \{\boldsymbol{z}_i^k\}_{k=1}^L\right), e^{-\frac{1}{\tau}}\right\},$$

where $L = M - 1$. Here a sampled negative item may comes from the positive items with a probability $1 - \tau^+$. $\mathcal{N} = \exp(\mathrm{s}(\boldsymbol{z}_u^l, \boldsymbol{z}_i^k)/\tau)$ and $\mathcal{P} = \exp(\mathrm{s}(\boldsymbol{z}_u^l, \boldsymbol{z}_i^j)/\tau)$. Finally, we can summarize the objective of the pre-training:$\mathcal{L}_{ssl} = \mathcal{L}_{user} + \mathcal{L}_{item} + \mathcal{L}_{user}^{item}$.

Pseudo-Siamese Network for Post-Training (PSP): Using the pseudo-siamese network architecture, similar model structure may have different parameters, enabling our model to learn meaningful representations of the different input separately and combine them at a higher level. Afterwards, we concatenate the learned user and item features to form richer augmented representations for prediction. Here, the prediction is defined as the inner product of the final user and item representations: $\hat{y}_{ui} = h_u^T h_i$.

To learn model parameters, we employ the *Bayesian Personalized Ranking* (BPR) loss, which is a pairwise loss that enforces the prediction of an observed interaction to be higher than its unobserved counterparts:

$$\mathcal{L}_{BPR} = -\sum_{u=1}^{n}\sum_{i\in N_u}\sum_{j\notin N_u} \ln\sigma(\hat{y}_{ui} - \hat{y}_{uj}) + \lambda\|\Theta\|^2, \quad (2)$$

where λ controls the L_2 regularization strength, Θ denotes the set of model parameters, and N_u denotes the set of items that are interacted by user u.

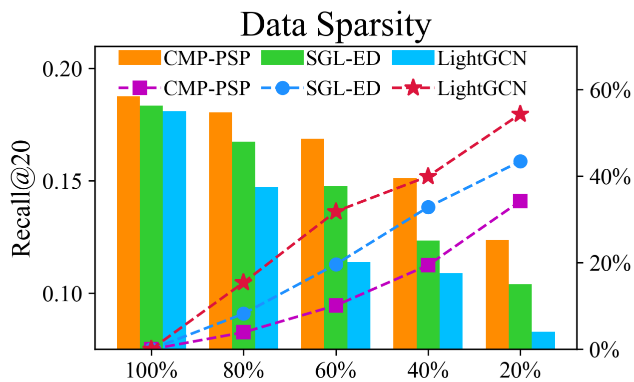
Results

Performance Comparisons: Table 1 shows the overall performance comparison results. Compared to NGCF (Wang et al. 2019) and LightGCN (He et al. 2020) (the best GCN-based CF baseline), CMP-PSP achieves 3.71% on Recall@20, 4.52% on NDCG@20 for the Gowalla dataset. The improvement is attributed to CMP-PSP's ability to alleviate the data sparsity and noisy interactions issues in user-item interaction learning with self-supervised signals.

Model Robustness: We conduct an experiment to investigate CMP-PSP's robustness to the data sparsity. The results are summarized in the Figure 1. To verify CMP-PSP's robustness to data sparsity, we use different proportions of the entire training set. We can find that the performance of LightGCN and SGL-ED deteriorates greatly when using less training data. However, the performance of CMP-PSP decreases slightly with less training data. This result indicates that CMP-PSP is able to alleviate the data sparsity problem greatly. We attribute this result to the ability of CMP-PSP to figure out useful interactions and reduce dependence on certain edges by comparing different augmented views of nodes.

Dataset	**Gowalla**	
Method	**Recall**	**NDCG**
NGCF	0.1569	0.1327
LightGCN	0.1810	0.1524
SGL-ED	0.1835	0.1539
CMP-PSP	**0.1877**	**0.1593**

Table 1: Overall Performance Comparisons.



(a) Gowalla

Figure 1: Impact of sparse data. The bars represent Recall@20 values and the polylines denote the performance degradation gain.

Acknowledgments

This work was supported by National Natural Science Foundation of China (Grant No. 62176043 and No. 62072077).

References

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020. A simple framework for contrastive learning of visual representations. In *ICML*, 1597–1607.

Chuang, C.-Y.; Robinson, J.; Lin, Y.-C.; Torralba, A.; and Jegelka, S. 2020. Debiased Contrastive Learning. In *NeurIPS*.

He, X.; Deng, K.; Wang, X.; Li, Y.; Zhang, Y.; and Wang, M. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *SIGIR*, 639–648.

Wang, X.; He, X.; Wang, M.; Feng, F.; and Chua, T.-S. 2019. Neural graph collaborative filtering. In *SIGIR*, 165–174.

Wu, J.; Wang, X.; Feng, F.; He, X.; Chen, L.; Lian, J.; and Xie, X. 2021. Self-supervised Graph Learning for Recommendation. In *SIGIR*.

Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; and Philip, S. Y. 2020. A comprehensive survey on graph neural networks. *TNNLS*.