ST5188 Project Research Notes

Yeo Ming Jie, Jonathan

1 Comments from Meeting on 22 Feb

Propose to represent users (or items) in terms of item embeddings. Dimensionality reduction. Does it lead to improved recommendation? or significant reduction in runtime at the drawback of slight reduction in accuracy? Can we find relevant literature on this? Able to formally prove if the proposed solution alleviates oversmoothing? Leverage on MixGCF to improve the training performance. (to balance the limitation of our proposed method)

2 Literature Research

2.1 INGCF: An Improved Recommendation Algorithm Based on NGCF [1]

Sun et al. (2022) propose an enhanced version of the Neural Graph Collaborative Filtering (NGCF) algorithm, known as INGCF. The key contributions of are as follows:

- In NGCF, node features are propagated through L layers to obtain $L+1$ (including initial) embeddings or feature representations for user-item nodes. These embeddings are concatenated to obtain final user-item embeddings, before inner product is taken to generate the prediction score.
- Sun et al. introduce a feature extraction enhancement layer, comprising of a 4-layer IndRNN layer and a self-attention mechanism layer after the concatenation to form a feature extraction enhancement layer. The status update of IndRNN is given by the following:

$$h_t = \mathrm{ReLU}(We_t + u \odot h_{t-1} + b) \tag{1}$$

where h_t and q_t are the hidden state and input vector at time step t, W is the input weight, u the recursive weight and b an offset value. Hadamard product $\odot$ (element-wise product) is taken between u and h_{t-1}. The output of the IndRNN layer is fed into a self-attention mechanism layer given by:

$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}(\frac{QK^T}{\sqrt{d_k}})V, \tag{2}$$

where Q, K and V denote the Query, Key and Value respectively, d_k representing the dimension of each head of the self-attention mechanism.

Experimental results demonstrate that INGCF achieves better performance in terms of both accuracy and efficiency than the original NGCF and other baseline algorithms. Note: A subsequent extension of this work to LightGCN has recently been pre-published. See [2] for more information.

Code for IndRNN: `https://github.com/batzner/indrnn`

Implementation and walkthrough of Self-Attention: Graph Attention Networks in Python

2.2 Effect of Cluster-based Sampling on the Over-smoothing Issue in Graph Neural Network [3]

Hoang and Ta (2022) propose the usage of cluster-based sampling for mitigating the over-smoothing issue in graph neural networks (GNNs) caused by repeated graph convolutional operations. Similar nodes are grouped into clusters, and representative nodes are sampled from each cluster for training. The sampling is performed in a way that preserves the local connectivity structure of the input graph, which helps the GNN to capture more discriminative features while avoiding over-smoothing. The results show that the proposed method outperforms other sampling methods and improves the performance of GNNs by reducing over-smoothing while preserving the expressiveness of the model.

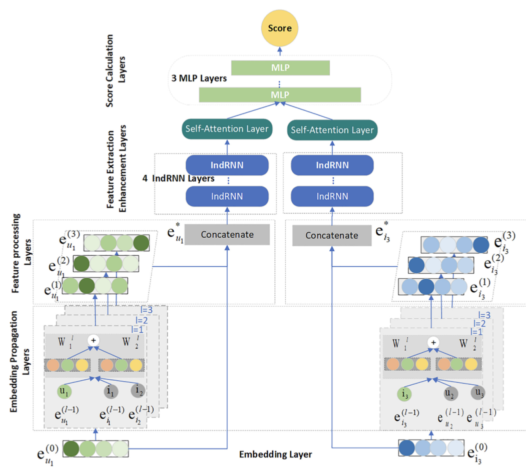

Figure 1: INGCF proposed overall architecture

Algorithm 1 Training GNN with cluster-base sampling

Require: Graph $(\mathcal{G}, \mathcal{V}, \mathcal{E})$ with node labels $X \in \mathbb{R}^{N \times d}$, labels $Y \in \mathbb{R}^{N \times b}$, number of epochs T.
Ensure: Vector representations H
Partition graph nodes into c clusters $\mathcal{V}_1, \mathcal{V}_2, \cdots, \mathcal{V}_c$
for t = 0 to T **do**
  Randomly choose q clusters, $t_1, , t_q$ from $\mathcal{V}$ without replacement
  Form the subgraph $\hat{\mathcal{G}}$ with nodes $\hat{V} = [\mathcal{V}_{t_1}, \mathcal{V}_{t_2}, , \mathcal{V}_{t_q}]$ and links $A_{\hat{\mathcal{V}}, \hat{\mathcal{V}}}$
  Compute $g \leftarrow \nabla \mathcal{L}_{A_{\hat{\mathcal{V}}, \hat{\mathcal{V}}}}$ (loss on the subgraph $A_{\hat{\mathcal{V}}, \hat{\mathcal{V}}}$ using GNN model
  Conduct Adam update using gradient estimator g
end for

Figure 2: Cluster-based Sampling Algorithm

Similar Code: `https://github.com/google-research/google-research/tree/master/cluster_gcn`

2.3 Revisiting Graph Based Collaborative Filtering: A Linear Residual Graph Convolutional Network Approach [4]

Chen et al. (2020) propose a novel network structure called Linear Residual Graph Convolutional Collaborative Filtering (LR-GCCF) specifically designed to address the over-smoothing problem that can arise in graph convolution aggregation operations.

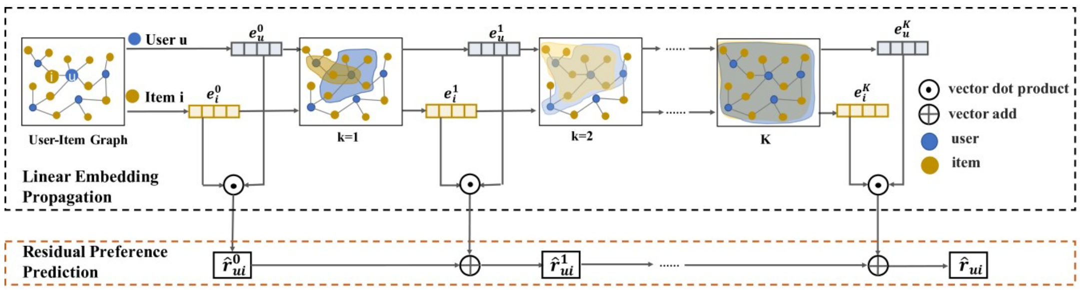

Figure 3: LR-GCCF proposed overall architecture

By incorporating a linear residual connection, the proposed framework can effectively preserve the high-frequency information of the input graph and improve the performance of GCN-based models in recommendation tasks. Since the proposed model is linear, LR-GCN is not only easier to train but can scale to large datasets to yield better efficiency and performance.

Note: This paper proves the equivalence of residual prediction and taking each user (item)'s embedding as a concatenation of all layers' embeddings, as in NGCF. Linear propagation is equivalent to that adopted by LightGCN.

Source code: `https://github.com/newlei/LR-GCCF`

2.4 ITSM-GCN: Informative Training Sample Mining for Graph Convolutional Network-based Collaborative Filtering

The main contribution of the paper is the introduction of the ITSM-GCN algorithm, which aims to improve the training efficiency and effectiveness of GCN-based collaborative filtering models. The ITSM-GCN algorithm achieves this by selecting informative training samples based on their prediction errors, relevance, and diversity. Specifically, the algorithm uses a two-step sampling process that first selects the samples with high prediction errors and then diversifies the selected samples to ensure sufficient coverage of the item space.

Note: Positive and Negative Sampling strategy built on top of LightGCN for demonstration. No code implementation available.

References

[1] Sun, W., Chang, K., Zhang, L. & Meng, K. INGCF: An Improved Recommendation Algorithm Based on NGCF. *Algorithms And Architectures For Parallel Processing*. pp. 116-129 (2022)

[2] Sun, W., Chang, K. & Zhang, B. ECCF: An Improved GNN Collaborative Filtering Algorithm for Edge Computing. (Research Square,2023), https://doi.org/10.21203/rs.3.rs-2532581/v1

[3] Hoang, T. & Ta, V. Effect of Cluster-based Sampling on the Over-smoothing Issue in Graph Neural Network. *2022 14th International Conference On Knowledge And Systems Engineering (KSE)*. pp. 1-6 (2022)

[4] Chen, L., Wu, L., Hong, R., Zhang, K. & Wang, M. Revisiting Graph based Collaborative Filtering: A Linear Residual Graph Convolutional Network Approach. (arXiv,2020), https://arxiv.org/abs/2001.10167

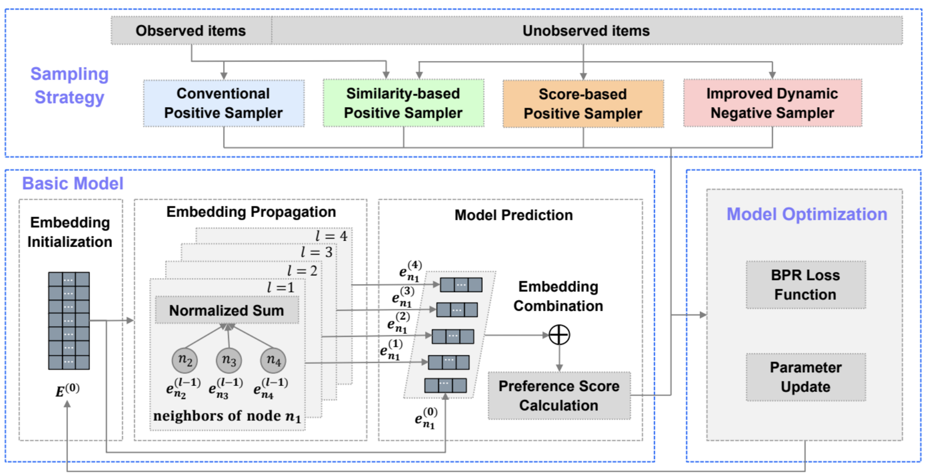

Figure 4: ITSM GCN Framework