COMP4801 Final Year Project

Interim Report

Self-supervised Graph Representation Learning for Recommender Systems

Name: Cai Xuheng

UID: 3035639173

Supervisor: Dr. Huang Chao

Department of Computer Science, HKU

Date of Submission: 2023/1/12
## Contents

1 Introduction ........................................................................................................................................... 1  
2 Current Progress .................................................................................................................................... 2  
3 Details of Work ...................................................................................................................................... 2  
   3.1 Phase 1: General Setting (Completed) .............................................................................................. 2  
      3.1.1 Methodology ............................................................................................................................ 3  
      3.1.2 Model Performance .................................................................................................................. 4  
      3.1.3 Model Benefits ......................................................................................................................... 5  
      3.1.4 Limitations ............................................................................................................................... 7  
   3.2 Phase 2: Model Compression (To-do) .............................................................................................. 8  
      3.2.1 Descriptions of the Model Compression Setting ......................................................................... 8  
      3.2.2 Timeline for Phase 2 ............................................................................................................... 9  
4 Conclusion ............................................................................................................................................... 9  
References ................................................................................................................................................. 10
1 Introduction

Recommender systems are widely deployed in the apps and services that we use daily, such as Netflix, Amazon, and so on. They are designed to provide personalized recommendations of items to the users. As a subfield of machine learning, they have attracted a lot of attention from the technology and business industry for their ability to analyze and predict user interests and behaviors in a variety of applications such as video streaming [1], social networks [2], and E-commerce [3]. Among the various techniques developed, graph convolutional network (GCN) has become a research hotspot in recent years [4,5] due to its promising performance and extensibility for further enhancement. By modeling the user-item relationship into a graph, GCN can effectively exploit the hidden information in the data.

However, existing GCN methods suffer from several issues, the most significant of which is the over-smoothing effect. Over-smoothing refers to GCN's tendency to represent neighboring users and items in the graph with highly similar embedding vectors, which limits the model's ability to identify the target items for a user [6]. Another issue, known as the data sparsity issue, arises because many users only interacted with very few items [7]. Such a lack of information impairs the accuracy of the models' predictions. In light of these issues, this project aims at leveraging the self-supervised learning (SSL) paradigm to enhance the GCN models. In particular, the team's objective is to design novel SSL models that can achieve state-of-the-art performance.

This report will first present the current status of the project. Then it will introduce the model that the team developed in the first phase of the project in detail. Specifically, it will describe the structure, performance against competitors, advantages, and limitations of the proposed model. Finally, it will outline what the team will work on in the project's second phase.
2 Current Progress

Our project is divided into two phases, with the focus of the first phase on the general setting of recommendation and the second phase on the specific setting of model compression\(^1\).

Since this project is a continuation of the Undergraduate Research Fellowship Program that started this summer, the background research and literature survey were completed by the end of August 2022. Currently, the team has accomplished the first phase by developing a simplified yet powerful SSL model named LightGCL that can achieve outstanding performance in the general setting of recommendation. The results of the work in phase 1 have been summarized into a paper submitted to the *International Conference on Learning Representations (ICLR) 2023*.

In the next step, the team will drill into the specific setting of model compression. It is expected that the experiences and techniques developed in phase 1 will be transferred to phase 2, facilitating the team to design a new model that can achieve better performance than the competitors.

3 Details of Work

3.1 Phase 1: General Setting (Completed)

This section discusses the model developed in phase 1 with detailed analyses of its performance and benefits.

\(^1\) Details about the model compression setting will be discussed in the later section of Phase 2.
3.1.1 Methodology

The proposed model adopts the paradigm of self-supervised graph convolutional network. The idea of the graph convolutional network (GCN) is to perform graph convolution on the graph, which is essentially a way to propagate the information of users and items across the graph so that each user and item can obtain an embedding vector that contains information of its neighbors. The key idea of self-supervised learning (SSL) is to create a variant of the graph, on which the graph convolution operation is also performed. The embedding vectors learned from the original and modified graph are then compared against each other to extract the most important underlying information.

Most of the existing SSL methods generate the modified graph with random perturbation of the original graph structure, which adds noisy and irrelevant information to the graph. In order to design a new approach to generate the modified graph, the team turned to singular value decomposition (SVD), a powerful mathematical technique in linear algebra to decompose a matrix. Inspired by the capacity of SVD to extract the principal components in the matrix, the team proposed to make use of SVD to create the modified graph.

![Figure 1 The Model Structure of LightGCL](image-url)
As shown in Figure 1, the model first performs an approximated SVD on the original graph to obtain three small matrices containing the graph's essential information. Then it reconstructs the graph using the three matrices. The reconstructed graph will be slightly different from the original graph, and thus can serve as the modified graph in SSL. In this way, the modified graph is constructed with meaningful information instead of random perturbation, which can facilitate the model to learn better. After building the modified graph, the model performs graph convolution on both graphs to obtain the embedding vectors, which are then plugged into the loss functions for optimization.

3.1.2 Model Performance

The team conducted extensive experiments on five real-world datasets using ten state-of-the-art competitor models as the baselines. The evaluation metrics were Recall@N and NDCG@N, which focused on the number of successfully predicted items and the order of the recommended items, respectively. The results are summarized in Table 1.

As seen from the table, the proposed model LightGCL consistently outperformed the baselines by approximately 10%, indicating the superiority of the proposed model in the recommendation task.

Table 1 The Overall Model Performance Against Baselines

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Metric</th>
<th>NCF</th>
<th>GCFF</th>
<th>LightGCN</th>
<th>DGCF</th>
<th>HyRec</th>
<th>MRGCN</th>
<th>SGFL</th>
<th>HCCF</th>
<th>SHT</th>
<th>SimGCL</th>
<th>LightGCL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yelp</td>
<td>r@20</td>
<td>0.2552</td>
<td>0.0462</td>
<td>0.0483</td>
<td>0.0466</td>
<td>0.0472</td>
<td>0.0464</td>
<td>0.0503</td>
<td>0.0526</td>
<td>0.0626</td>
<td>0.0651</td>
<td>0.0718</td>
</tr>
<tr>
<td></td>
<td>r@20</td>
<td>0.0202</td>
<td>0.0039</td>
<td>0.0089</td>
<td>0.0095</td>
<td>0.0095</td>
<td>0.0095</td>
<td>0.0095</td>
<td>0.0100</td>
<td>0.0100</td>
<td>0.0132</td>
<td>0.1357</td>
</tr>
<tr>
<td></td>
<td>r@20</td>
<td>0.0487</td>
<td>0.0070</td>
<td>0.0803</td>
<td>0.0774</td>
<td>0.0791</td>
<td>0.0626</td>
<td>0.0669</td>
<td>0.1040</td>
<td>0.1091</td>
<td>0.1191</td>
<td>0.1292</td>
</tr>
<tr>
<td></td>
<td>r@20</td>
<td>0.0289</td>
<td>0.0058</td>
<td>0.0527</td>
<td>0.0511</td>
<td>0.0522</td>
<td>0.0544</td>
<td>0.0571</td>
<td>0.0681</td>
<td>0.0709</td>
<td>0.0778</td>
<td>0.0852</td>
</tr>
<tr>
<td>Gowalla</td>
<td>r@20</td>
<td>0.0171</td>
<td>0.0951</td>
<td>0.0985</td>
<td>0.0944</td>
<td>0.0901</td>
<td>0.0955</td>
<td>0.1030</td>
<td>0.1070</td>
<td>0.1232</td>
<td>0.1387</td>
<td>0.1578</td>
</tr>
<tr>
<td></td>
<td>r@20</td>
<td>0.0106</td>
<td>0.0535</td>
<td>0.0593</td>
<td>0.0522</td>
<td>0.0499</td>
<td>0.0574</td>
<td>0.0623</td>
<td>0.0644</td>
<td>0.0731</td>
<td>0.0818</td>
<td>0.0935</td>
</tr>
<tr>
<td></td>
<td>r@20</td>
<td>0.1434</td>
<td>0.1392</td>
<td>0.1431</td>
<td>0.1401</td>
<td>0.1356</td>
<td>0.1393</td>
<td>0.1500</td>
<td>0.1535</td>
<td>0.1804</td>
<td>0.1956</td>
<td>0.2245</td>
</tr>
<tr>
<td>Gowalla</td>
<td>r@20</td>
<td>0.0118</td>
<td>0.0684</td>
<td>0.0710</td>
<td>0.0671</td>
<td>0.0669</td>
<td>0.0689</td>
<td>0.0726</td>
<td>0.0767</td>
<td>0.0881</td>
<td>0.0975</td>
<td>0.1108</td>
</tr>
<tr>
<td></td>
<td>r@20</td>
<td>0.0197</td>
<td>0.1742</td>
<td>0.1789</td>
<td>0.1763</td>
<td>0.1801</td>
<td>0.1497</td>
<td>0.1833</td>
<td>0.2219</td>
<td>0.2173</td>
<td>0.2265</td>
<td>0.2613</td>
</tr>
<tr>
<td></td>
<td>r@20</td>
<td>0.1297</td>
<td>0.2109</td>
<td>0.2128</td>
<td>0.2101</td>
<td>0.2178</td>
<td>0.1814</td>
<td>0.2205</td>
<td>0.2629</td>
<td>0.2573</td>
<td>0.2613</td>
<td>0.3106</td>
</tr>
<tr>
<td>Gowalla</td>
<td>r@20</td>
<td>0.1427</td>
<td>0.2331</td>
<td>0.2322</td>
<td>0.2340</td>
<td>0.2340</td>
<td>0.1962</td>
<td>0.2268</td>
<td>0.2261</td>
<td>0.2211</td>
<td>0.3345</td>
<td>0.2799</td>
</tr>
<tr>
<td></td>
<td>r@20</td>
<td>0.0142</td>
<td>0.0317</td>
<td>0.0319</td>
<td>0.0211</td>
<td>0.0302</td>
<td>0.0296</td>
<td>0.0327</td>
<td>0.0322</td>
<td>0.0441</td>
<td>0.0474</td>
<td>0.0588</td>
</tr>
<tr>
<td></td>
<td>r@20</td>
<td>0.0085</td>
<td>0.0243</td>
<td>0.0236</td>
<td>0.0154</td>
<td>0.0225</td>
<td>0.0219</td>
<td>0.0249</td>
<td>0.0247</td>
<td>0.0328</td>
<td>0.0360</td>
<td>0.0436</td>
</tr>
<tr>
<td></td>
<td>r@20</td>
<td>0.0253</td>
<td>0.0483</td>
<td>0.0499</td>
<td>0.0351</td>
<td>0.0432</td>
<td>0.0489</td>
<td>0.0551</td>
<td>0.0555</td>
<td>0.0719</td>
<td>0.0750</td>
<td>0.0933</td>
</tr>
<tr>
<td>Amazon</td>
<td>r@20</td>
<td>0.0133</td>
<td>0.0285</td>
<td>0.0290</td>
<td>0.0201</td>
<td>0.0246</td>
<td>0.0284</td>
<td>0.0312</td>
<td>0.0314</td>
<td>0.0420</td>
<td>0.0451</td>
<td>0.0551</td>
</tr>
<tr>
<td>Tmall</td>
<td>r@20</td>
<td>0.0082</td>
<td>0.0209</td>
<td>0.0225</td>
<td>0.0235</td>
<td>0.0233</td>
<td>0.0203</td>
<td>0.0268</td>
<td>0.0314</td>
<td>0.0387</td>
<td>0.0473</td>
<td>0.0528</td>
</tr>
<tr>
<td></td>
<td>r@20</td>
<td>0.0099</td>
<td>0.0141</td>
<td>0.0154</td>
<td>0.0163</td>
<td>0.0300</td>
<td>0.0139</td>
<td>0.0183</td>
<td>0.0231</td>
<td>0.0262</td>
<td>0.0328</td>
<td>0.0361</td>
</tr>
<tr>
<td></td>
<td>r@20</td>
<td>0.0071</td>
<td>0.0196</td>
<td>0.0208</td>
<td>0.0218</td>
<td>0.0199</td>
<td>0.0188</td>
<td>0.0246</td>
<td>0.0284</td>
<td>0.0352</td>
<td>0.0429</td>
<td>0.0473</td>
</tr>
</tbody>
</table>
3.1.3 Model Benefits

Besides the model's overall performance, the team also analyzed in detail the proposed model's benefits in alleviating the common issues of the GCN models.

Firstly, to investigate the effectiveness of LightGCL in learning a moderately dispersed embedding distribution, the team randomly sampled 2,000 nodes from two datasets, Yelp and Gowalla, and mapped their embeddings to the 2-D space with t-SNE [8]. The visualizations of these embeddings are presented in Figure 2. As shown in the figure, the embedding distributions of non-SSL methods (LightGCN [4] and MHCN [9]) formed indistinguishable clusters in the embedding space, indicating the presence of the over-smoothing issue. In contrast, the existing SSL-based methods (SGL [10] and SimGCL [11]) learned over-uniform distributions with no clear community structure to capture the collaborative relations between users. Compared with them, the proposed model learned an embedding distribution with clear community structures, while the embeddings inside each community were reasonably dispersed to reflect the user-specific preferences.

![Figure 2 The Visualization of the Embeddings Learned by Different Models in 2D Space](image)

Secondly, the team evaluated the robustness of the proposed model in mitigating the data sparsity issue. The team grouped the sparse users by their interaction degrees and calculated the Recall@20 metric of each group on the Yelp and Gowalla datasets. The
results are plotted in Figure 3. As can be seen from the figure, the performance of HCCF and SimGCL varied across datasets, but LightGCL consistently outperformed them in all cases. In particular, the proposed model performed notably well on the extremely sparse user group (< 15 interactions), as the Recall@20 of these users was not much lower (and was even higher on Gowalla) than that of the whole dataset.

The team also illustrated the proposed model’s ability to mitigate popularity bias compared to HCCF and SimGCL. Popularity bias refers to the unwanted situation where the model blindly recommends popular items to the users simply because of their frequent presence in the interaction graph. Similar to the method described above to evaluate the resistance against data sparsity, the long-tail items were grouped by their degree of interactions, and the decomposed Recall@20 was calculated for each group. The results are charted in Figure 4. As shown by the figure, LightGCL performed better than the competitors in most cases, indicating the ability of the proposed model to resist the popularity bias issue.
3.1.4 Limitations

Despite the superior performance and the benefits of the proposed method, the team found that it performed poorly on datasets containing a large proportion of noisy interactions. The team injected different ratios of random noises into the Yelp dataset, and tested the performance of LightGCN, SimGCL, and the proposed LightGCL on the modified datasets. The ratios of the models’ performance on these noisy datasets to that on the original dataset are summarized in Table 2. We can see from the table that the performance of LightGCL degraded faster than the other two competitors as the noise ratio increased. The reason for this drawback may be that the model exploits the information in the graph to a large extent, and thus becomes highly dependent on the reliability of the information in the graph. Therefore, if the graph contains too much noise, the model will be misled.
### Table 2 The Models' Performance with different noise ratios in the dataset, presented as a percentage of the performance on the original dataset.

<table>
<thead>
<tr>
<th>Noise Ratio</th>
<th>5%</th>
<th>10%</th>
<th>15%</th>
<th>20%</th>
<th>25%</th>
</tr>
</thead>
<tbody>
<tr>
<td>SimGCL</td>
<td>0.9777</td>
<td>0.9623</td>
<td>0.9442</td>
<td>0.9220</td>
<td>0.9011</td>
</tr>
<tr>
<td>LightGCN</td>
<td>0.9632</td>
<td>0.9346</td>
<td>0.8775</td>
<td>0.8591</td>
<td>0.8326</td>
</tr>
<tr>
<td>LightGCL</td>
<td>0.8377</td>
<td>0.7439</td>
<td>0.6869</td>
<td>0.6476</td>
<td>0.6121</td>
</tr>
</tbody>
</table>

### 3.2 Phase 2: Model Compression (To-do)

In the second phase, the team plans to dig into the specific setting of model compression, leveraging the experience gained in Phase 1.

#### 3.2.1 Descriptions of the Model Compression Setting

In the real-world applications, recommender systems are often implemented in embedded systems, mobile devices, and small-scale hardware where only limited computational resources are available [13]. However, GNN-based recommender systems induce high computational costs, reducing their practicability in these scenarios. In light of this, some researchers have been exploring strategies to compress the trained model so as to reduce the inference time. Some of them adopt the parameter pruning paradigm, where as many parameters are set to zero as possible after training [12]. Another commonly used approach is the knowledge distillation strategy, where a teacher model and a student model are trained together to preserve the learned knowledge in the simpler student model [13].

While many papers have explored the compression of the general GNN models (usually for the task of node classification), there are few on the compression of the GNN recommenders. What's worse, the general GNN compression methods are usually not
applicable to GNN recommenders due to the nature of the BPR loss used in recommendation tasks. Therefore, in the second phase of our project, we focus on developing a workable strategy for the compression of GNN-based recommender systems.

### 3.2.2 Timeline for Phase 2

<table>
<thead>
<tr>
<th>Nov. – Dec. 2022</th>
<th>Literature Review</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jan. – Mar. 2023</td>
<td>Experiments</td>
</tr>
<tr>
<td>Apr. 2023</td>
<td>Summary</td>
</tr>
</tbody>
</table>

As summarized in Table 3, the team expects to complete the relevant literature review by December 2022 and start the main experiments of the second phase in January 2022. Extensive experiments will be conducted from January to March 2023, and the whole project is scheduled to be completed by April 2023.

### 4 Conclusion

This report summarized the current progress of the research project on leveraging self-supervised learning to overcome common issues in recommender systems. Specifically, the team has proposed a novel model named LightGCL that adopts a simplified SSL framework using singular value decomposition. Extensive experiments on five real-world datasets proved the superiority of the proposed model against ten state-of-the-art competitors. More analyses were given in the report to illustrate the model's strong ability to alleviate common issues in recommendation, including over-smoothing, data sparsity, and popularity bias.
Despite the outstanding performance of the proposed model, the report also pointed out its limitations in dealing with noisy datasets. With its sophisticated exploitation of the data, the model heavily relies on the credibility of the input data. Thus, the model performance degrades as more noise is added to the data. The question of how to design models that maintain resistance against noises while exploiting the data in-depth is left to future research.

In the next step, the team will focus on transferring the previously developed knowledge to the model compression setting. The literature review is expected to finish by December, and the whole project will be completed before April 2023.

References


