Proceedings of the 2nd International Conference on Computing Innovation and Applied Physics
DOI: 10.54254/2753-8818/19/20230568

Contrastive representation learning in recommendation systems--The investigation of the performance of the self-supervised learning in large-scale recommendation systems

Yuxin Li[1,*], Jingyi Wang[2,6], Xinyang Wu[3,7], Rui Zhou[4,8], Baichuan Xu[5,9]

[1] College of Liberal Arts and Sciences, University of Illinois at Urbana-Champaign, Urbana, IL, 61801, United States
[2] College of Computer Science and Electronic Engineering, Hunan University, Changsha, 410082, China
[3] Department of Information Science and Engineering, East China University of Science and Technology, Shanghai, 200237, China
[4] The Grainger College of Engineering, University of Illinois at Urbana-Champaign, Urbana, IL, 61801, United States
[5] College of Information and Communication Engineering, Communication University of China, Beijing, 100024, China

[*]liyuxin_icc_7@163.com, [6]jingyiw@hnu.edu.cn, [7]1335045869@qq.com, [8]ruizhou3@illinois.edu, [9]baichuan.xu6@gmail.com

Abstract. Self-supervised learning (SSL) has been proposed in machine learning projects for its convenience of reducing self-labeled datasets in recent years. However, the implementation of SSL in large-scale recommendation systems has lagged behind the evolution because of their scarce and tailed characteristics. In 2021, an article proposed the use of SSL in recommendation systems to pursue an improvement in the performance of recommender models. This article is built on this previous investigation and aims to further explore the role of SSL in recommendation systems and to investigate an improvement of the model's efficiency. To answer research questions, this paper tests three models with different numbers of towers to discover the best performance of the use of SSL in recommender models. Consequently, it is found that implementing SSL on the item side only (two-tower DNNs) produced the best result. Then, when constructing the two-tower DNNs model, this article examines different numbers of negative pairs to change the InfoNCE loss to investigate a tradeoff between the number of positive and negative samples in the performance of the model. As a result, it turns out to be a weak correlation between this ratio and the performance; Hence, it is concluded that the change of the number of positive and negative samples would not necessarily affect the two-tower DNNs model. In our experiential stage, this paper uses a real-world dataset with 100k training samples to testify and compare our results.

Keywords: contrastive representation learning, super-supervised learning, recommendation systems, neural networks, two-tower DNN.

DOI: 10.54254/2753-8818/19/20230568

1. Introduction

Nowadays, online shopping systems and applications have exponentially developed. To provide more accurate predictions and recommendations based on the user's interests, researchers and scientists have considered implementing machine learning in the recommendation systems. However, due to the data sparsity and the trailed characteristics, such an implementation has encountered obstacles.

In 2020, Zhou et al. proposed a novel Self-Supervised Learning (SSL) method for recommendation systems [1]. It records four correlations in the pre-training stage: item-attribute, sequence-item, sequence-attribute and sequence-subsequence. With these four self-supervised learning objectives, the problem of data sparsity and insufficiency in prediction accuracy has been solved. After constructing the four correlations, it is suggested to maximize the mutual information between the encoded representations through the objective function. Similarly, in 2021, a paper has referred to SimCLR and applied it to categorical features to achieve the same goal as stated by Zhou et al. [1-3]. It concluded that the use of SSL was beneficial for recommender models [2]. However, Yao et al. only implemented SSL to the item side and left another important aspect of recommendation systems, user-side or query-side, supervised [2]. Therefore, stimulated by the improved performance of SSL and this finding, in this paper, we decided to investigate and compare the predictions done by models with different numbers of towers.

In addition, when evaluating the loss functions, we observed that the authors set the number of negative pairs to consistently be 1 [2]. Hence, this paper also aims to change the InfoNCE loss by adjusting the number of negative pairs to try to pursue predictions with higher accuracy.

Hence, in this work, we discover one-tower, two-tower, and three-tower DNN(s) models and examine the numbers of negative samples in InfoNCE loss.

2. Related Work

2.1. SSL and SimCLR

SSL is a method to train a model based on the features obtained from the raw data. The basic idea is that the model will auto-generate labels from the feature correlations for further use. It has been used in a variety of fields, such as natural language processing, image classification and object detection.

In our study, we manipulated mutual information maximization based on the basic SimCLR model [3]. We generated data augmentations from the chosen data and tried to maximize the mutual information between the representations of these data. These generated data are positive data. The representations derived from augmentations of other data are negative data. We try to minimize the mutual information between the representations of the negative data and the positive data.

2.2. Self-supervised Learning for Large-scale Item Recommendations

Inspired by Yao et al., we considered their two-tower DNNs model as a reference for our comparisons between the three models: supervised learning (one-tower DNN) and SSL (two-tower DNNs and three-tower DNNs) [2]. Also, since Yao et al. have presented the relative results of different data augmentation methods and claimed that the feature correlated masking with dropout outperformed other methods, we decided to inherit this method for feature embeddings of our three models [2].

*2.2.1. Two-tower DNNs Model with SSL.*The two-tower DNNs model incorporates supervised and SSL parts in its model architecture [2].

Each tower consists of three layers: input, representation, and matching layers. The data will be passed into embeddings and neural networks to be trainable.

The SSL part of the two-tower model has towers passed by augmented item features. Each tower in the SSL part shares parameters in the DNN layer.

2.2.2. Data Augmentation - Correlated Feature Masking. One of the key elements in the SSL model is to determine the method of data augmentation. Yao et al. proposed a novel augmentation method,

DOI: 10.54254/2753-8818/19/20230568

Correlated Feature Masking (CFM) with dropout, and prove its better performance among other methods, such as Random Feature Masking or pure Masking [2].

The CFM implements two stages. It first randomly selects one masking feature from inputs and then makes a chain of correlated features based on their correlation with the selected one. After applying these two processes, the model will mask features in the whole chain.

As proposed by Yao et al., the combination of CFM and dropout is the method that produces the best predictions [2].

3. Methodology

Our methodology used Matrix Factorization (MF) as the base backbone and referred to the basic framework of SimCLR, which is to maximize the performance of similar items and minimize the correlation of different items [3]. To achieve this goal, we induced the InfoNCE loss function.

We divided our investigation into two stages. First, we discover the performance of supervised and Self-supervised MF in recommendation systems in section 3.1. Then, followed by the choice of the model framework with the best performance, we investigated the influence of the change in the number of negative samples on the model's performance, and our discussion will be presented in section 3.2.

3.1. Model Architectures

As mentioned in section 2.2.1, we proposed a two-tower DNNs framework. Inspired by their model and realizing that they had only implemented SSL on the item side, we decided to investigate a three-tower DNNs model, applying SSL on both the user and the item sides, and comparing their results. This section aims to find an answer to our first research question [2].

3.1.1. Supervised MF (One-tower DNN model). First, to justify the improvements that SSL could bring to recommender models, we tested the results under the MF without implementing SSL, or namely the one-tower DNN model. The results will be considered as the control group of our experiment. Figure 1 represents the architecture of the one-tower model.

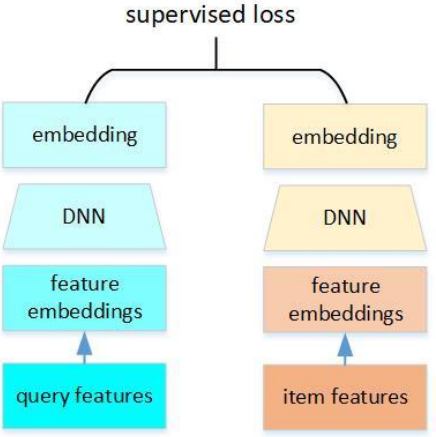

Figure 1. The one-tower DNN model.

For this model, we calculated our InfoNCE loss function using formula (1).

$$L_{main} = -\frac{1}{N}\sum_{i\in[N]} \log \frac{\exp(s(q_i, x_i)/\tau)}{\sum_{j\in[N]} \exp\left(\frac{s(q_i, x_j)}{\tau}\right)} \quad (1)$$

In the formula, qi and xi are the corresponding queries and items, whereas xj are items that are irrelevant to the investigated query, $qi.t$ is the SoftMax temperature. Up to this stage, N, the number of negative samples, is set to 1.

DOI: 10.54254/2753-8818/19/20230568

Our aim here is to maximize the function concerning qi and $xi, s(qi, xi)$, and to minimize the function of qi and $xj, s(qi, xi)$.

3.1.2. Self-supervised MF (Two-tower DNNs Model). Additionally, we referenced the two-tower DNNs model stated by Yao et al. [2]. The model consists of the supervised part and the self-supervised part. To achieve the SSL model, we followed the best data augmentation method mentioned in section 2.2.2, that is applying the CFM with dropout to the item embeddings, as shown in the right part of Figure 2.

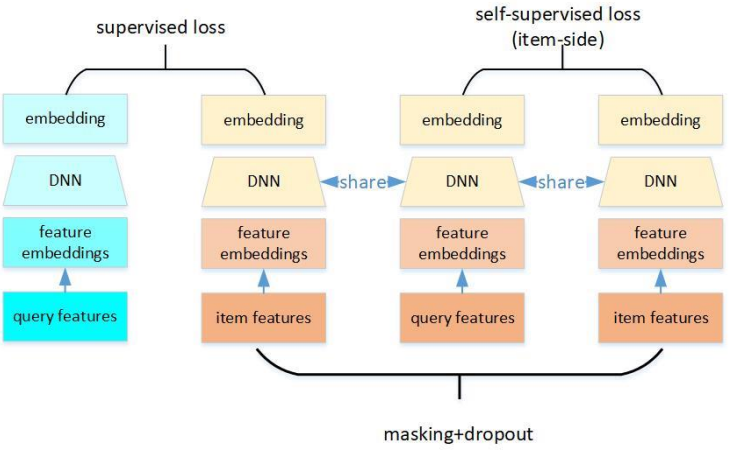

Figure 2. The two-tower DNNs model.

For this model, our loss function consists of two parts, the main (supervised) loss and the self-supervised loss. The main loss is defined the same as the formula (1), whereas the self-supervised loss is defined using formula (2).

$$L_{self(\{x_i\};\, H,G)} \coloneqq -\frac{1}{N}\sum_{i\in[N]} \log \frac{\exp(s(z_i, z_i')/\tau)}{\sum_{j\in[N]} \exp\left(\frac{s(z_i, z_j')}{\tau}\right)} \tag{2}$$

In this formula, H and G are neural networks, zi' is the augmented data of zi, both of whom are defined as positive pairs. zj' and zi on the denominator are negative pairs. t is the SoftMax temperature, and N, the number of negative pairs, is set to be 1.

The main goal for the loss function (2) is to represent the categorical features, that is to maximize the function concerning zi and zi' and to minimize the correlation between zj' and zi.

After calculating the supervised and self-supervised loss using formula (1) and (2), we apply formula (3) to get the InfoNCE of the whole system.

$$L = Lmain(\{(qi, xi)\}) + a\, Lself(\{xi\}) \tag{3}$$

a in formula (3) is the coefficient to adjust the contribution of the SSL part to the two-tower DNNs model. For convenience, we decided to inherit the tested a applied by Yao et al. in our experiment [2].

3.1.3. Self-supervised MF (Three-tower DNNs Model). Finally, since SSL has been proven to be efficient, we were curious about the performance of implementing SSL to the query side (user-side) as well, and hence, we proposed the three-tower DNNs model [2]. In our model, we applied SSL in both the query and the item sides as illustrated in Figure 3. Therefore, our loss function consists of three parts: the main loss and two self-supervised losses.

DOI: 10.54254/2753-8818/19/20230568

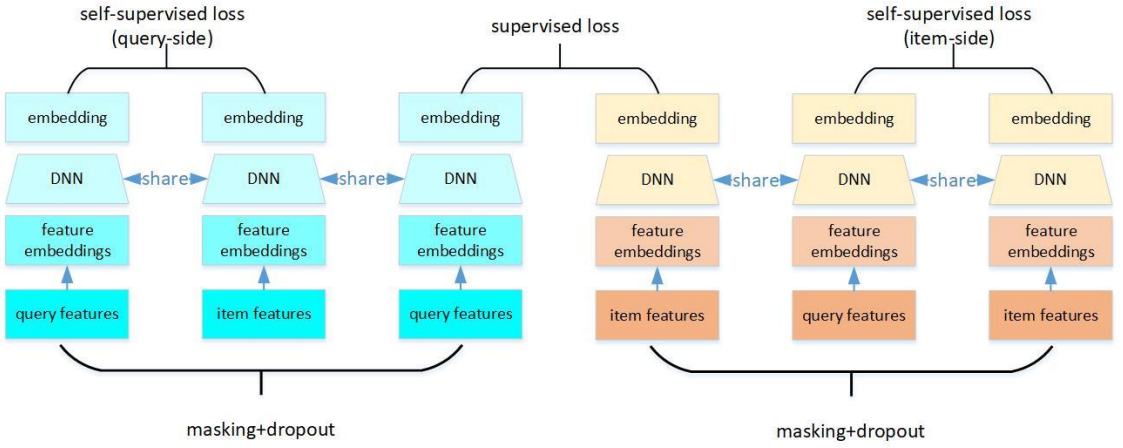

Figure 3. The three-tower DNNs model.

We inherited the CFM and dropout in the SSL of both sides and calculated the two self-supervised loss functions using formula (2). After these procedures, we obtained the InfoNCE loss function of the three-tower DNNs model as presented in formula (4).

$$L = Lmain(\{(qi, xi)\}) + b\, Lself_{item(\{xi\})} + c\, Lself_{query(\{xi\})} \quad (4)$$

b and c are the coefficients to adjust the relative importance of the two SSL losses in the whole system.

3.2. Data Augmentation

As presented in sections 3.1.1 and 3.1.2 and formulas (1) and (2), when calculating the loss functions, we have set N to be 1. Consequently, to answer our second research question, we decided to investigate the corresponding results by adjusting the N ranging from 1 to 20.

We kept using CFM with dropout as our data augmentation method and gradually increased the number of negative pairs in the recommender model.

This investigation was explored after our discussion of the first research question, which means that we only adjusted the scope of N on the model with the best performance. This is because one of the major goals of our paper is to find the most efficient model for large-scale recommendations, so the two less effective models were abandoned at the first stage.

4. Experiment

In this section, we used a real-world dataset with 100k samples to address our two research questions:

RQ1: Which number of towers produces the best predictions as a recommender model?

RQ2: How and to what extent does the change in number of negative pairs in SSL affect the performance of the recommender model?

4.1. Datasets

In our experiment, we used a ML 100k dataset as our original data. The data file consists of item ids, user ids, and ratings.

To better fit into the model, we have binarized the ratings by setting the ones larger or equal to 4 to be 1 and the other to be 0. Also, we have randomly generated the train and test dataset with a ratio of 8:2. Table 1 illustrates the characteristics of our dataset.

DOI: 10.54254/2753-8818/19/20230568

Table 1. The result of our original dataset.

Dataset	User#	Item#	Interaction#	Sparsity
ML 100k	943	1682	100,000	93.70%

According to Table 1, our original dataset has a high sparsity rate as indicated as the common characteristic of data in recommendation systems.

4.2. Experiment Protocol

In our experiment, for the back-bone two-tower DNNs model, we adopted the learning rate, SoftMax temperature, and batch size from the work of Yu et al. since they have already reported a protocol that produces the best performance [4]. Specifically, we tested our ML 100k dataset using the two-tower DNNs model with 0.01 as its learning rate, 0.6 as its SoftMax temperature, and 1024 as its batch size. For the other two models, one-tower and three-tower DNNs models, after experimenting, we used 0.001 as their learning rate since this gives the best results and the rest of the parameters remain the same as those in the two-tower DNNs model. All of our three models are trained with the Adam optimizer with embedding size 64.

As mentioned in the Methodology section, we considered and adjusted the SSL loss multiplier feature to determine the relative importance of the SSL loss on the overall loss of the whole system. We have tested the multiplier within a range including [0.1, 0.3, 1.0, 3.0]. Also, guided by Yu et al., dropout rate plays an important role in SSL performance, and we conducted experiments using the rates from [0.1, 0.2, 0.3, 0.4, 0.5] [4]. We presented the best result derived from the combinations of different multiplier features and dropout rates.

4.3. Comparisons between the Three Models

To answer the RQ1, we compared three models with the two-tower DNNs model as backbone. They are: one-tower DNN (supervised MF, denoted as baseline), two-tower DNNs (SSL on the item side only, denoted as SSL for item side), and three-tower DNNs (SSL on both the query and the item side, denoted as SSL for both user and item side).

To compare the models, we selected four aspects, Hit Ratio, Precision, Recall, and Normalized Discounted Cumulative Gain, on their Top-20 predictions (denoted as HR@20, Precision@20, Recall@20, and NDCG@20 respectively). The explanations and formulas of these aspects are shown below [5]:

Hit Ratio measures the number of times that a test item occurs in the recommended list and it is calculated by formula (5).

$$\text{HR@20} = \frac{\text{Number of Hits @20}}{Ground - truth\ Item\ set} \tag{5}$$

Precision measures the number of correct predictions among all positive sets (user-expected results) and its formula is presented as formula (6).

$$\text{Precision} = \frac{Number\ of\ True\ and\ Positive}{Number\ of\ True\ and\ Positive\ + Number\ of\ False\ and\ Positive} \tag{6}$$

Recall is used to describe the number of items that are successfully recalled and is calculated by formula (7).

$$\text{Recall} = \frac{Number\ of\ True\ and\ Positive}{Number\ of\ True\ and\ Positive + Number\ of\ False\ and\ Negative} \tag{7}$$

Normalized Discounted Cumulative Gain examines the model's ability to get top ranks accurately. It is considered to be the most important indication among all the four measurements and is calculated by formula (8).

DOI: 10.54254/2753-8818/19/20230568

$$\text{NDCG@20} = \text{Z}_\text{k} \sum_{i=1}^{20} \frac{2^{r_i} - 1}{\log_2(i+1)} \tag{8}$$

Table 2 illustrates the performance measured by the four aspects of the Top-20 predictions with the three models respectively.

Table 2. The performance of the Top-20 predictions made by the three models.

Model Name	HR@20	Precision@20	Recall@20	NDCG@20
Baseline	0.0656	0.0689	0.1468	0.1109
SSL for item side	0.1333	0.1399	0.1985	0.2144
SSL for both user and item side	0.1093	0.1148	0.1739	0.1732

From Table 2, it is shown that the two SSL models outperform the baseline model (non-SSL), which further indicates the benefits of implementing SSL in recommendation systems. However, compared to our backbone, the two-tower DNNs model, the SSL model for both the user and item side have poorer performance. Hence, the two-tower DNNs model has been justified to be the best model among the three, and subsequently, we used this model architecture to investigate the impact of different numbers of negative samples on the model's performance.

4.4. Results from Different Numbers of Negative Pairs

To answer RQ2, we changed the number of negative sets used in InfoNCE loss in the two-tower model. The negative pairs are randomly selected from existing samples. And for each respective number, we ran the model several times to achieve an average performance. We tested the numbers, N, from 1 to 20, and Table 3 illustrates the average performance measured by the four indications in section 4.3 of N as 1, 5, 10, 15.

Table 3. The results gained from different numbers of negative pairs.

Number of negative pairs	HR@20	Precision@20	Recall@20	NDCG@20
1	0.1049	0.1109	0.1590	0.1576
5	0.0921	0.0974	0.1378	0.1350
10	0.0961	0.1015	0.1402	0.1409
15	0.0996	0.1053	0.1456	0.1449

As shown in Table 3, as the number of negative pairs increases, the performance has a small fluctuation and does not indicate a clear increasing or decreasing trend. Hence, enlarging the number of negative pairs might not be an effective way to improve the performance of the two-tower DNNs model.

5. Conclusion

This paper investigates the performance of a supervised model and two SSL models in recommendation systems with MF as the backbone. Specifically, it examines the impact of different numbers of towers on a model's performance. It also evaluates the changes in the numbers of negative pairs as a means of improving the predictions of a SSL model.

After conducting a ML 100k dataset, it is concluded that implementing SSL is an effective way to improve recommender models. It also draws a result that applying SSL to the item side only outperforms the one that applies to both the query and item side. This might imply that item-side embedding is more important for SSL in recommendation systems. In addition, it is found that enlarging the numbers of negative pairs in loss functions will not necessarily improve the model's

DOI: 10.54254/2753-8818/19/20230568

predictions, which suggests that the tradeoff between positive and negative pairs might not be a dominant factor when calculating InfoNCE loss.

For further research, it is recommended to test the impact of SSL in other basic backbone such as Neural Collaborative Filtering.

Acknowledgement

Yuxin, Li and Jingyi, Wang contributed equally to this work and should be considered co-first authors.

References

[1] Zhou, K., Wang, H., Zhao, W. X., Zhu, Y., Wang, S., Zhang, F., Wang, Z., & Wen, J.-R. (2020). S^3-Rec: Self-Supervised Learning for Sequential Recommendation with Mutual Information Maximization. https://doi.org/10.48550/arXiv:2008.07873

[2] Yao, T., Yi, X., Cheng, D. Z., Xu, F., Chen, T., Menon, A., Hong, L., Chi, E. H., Tjoa, S., Kang, J. (Jay), & Ettinger, E. (2021). Self-supervised Learning for Large-scale Item Recommendations. https://doi.org/10.48550/arXiv.2007.12865

[3] Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020). A Simple Framework for Contrastive Learning of Visual Representations. https://doi.org/10.48550/arXiv:2002.05709v3

[4] Yu, J., Yin, H., Xia, X., Chen, T., Li, J., & Huang, Z. (2022). Self-Supervised Learning for Recommender Systems: A Survey. https://doi.org/10.48550/arXiv.2203.15876

[5] He, X., Chen, T., Kan, M., Chen, X. (2015). TriRank: Review-aware Explainable Recommendation by Modeling Aspects. http://doi.org/10.1145/2806416.2806504