TransCDR: User Group Enhanced Cross-Domain Recommendation via Transformers

Cheng Zhang
Faculty of Artificial Intelligence in Education
Central China Normal University
Wuhan
zc2021@mails.ccnu.edu.cn

Abstract

Cross-domain recommendation (CDR) has become a research hot spot in recent years. CDR learns the information in the source domain and transfer it into the target domain. Recently, autoencoder in deep learning has been utilized in CDR. However, existing method cannot reveal the semantic relationships of latent representations. In this paper, we propose a novel user group enhanced model for CDR based on Transformer (TransCDR) that provides a solution to this challenge. Specifically, we propose a novel user group enhanced methodology and a novel encoder-decoder framework that learns the semantic information via Transformer in the encoded latent space, which open a new research direction for CDR. Experimental results show that our model is competitive with state-of-art methods and can learn the semantic relationships of user rating patterns. Our code is available[1].

1. Introduction

Recommendation systems have long been a hotspot area in machine learning. In the last decade, there were numerous researches on recommendation systems. Recently, as deep learning become prevalent in all research domains. Neural networks started in take place in recommendation systems [1-3] and achieved promising and competitive performance compared with traditional numerical methods [4], such as matrix factorization (MF). In order to solve the cold start and sparsity problems, cross-domain recommendation (CDR) has attracted much attention in the past few years. As an emerging research area, existing CDR methods are mainly based on traditional methodologies or primitive fully connected neural networks (FCN), which means the superior architectures, such as graph neural network (GCN), are not explored in literature. Another reason is that the data structure of CDR has large differences compared with the main stream tasks that can be easily modified with deep learning methodologies. This challenge includes: 1) severe data sparsity versus the intrinsic data greedy nature of deep neural networks. 2) The objective of CDR is hard to restate in deep learning styles. Most methods for CDR still rely on the ideology of MF, which is not capable of parallel training.

Therefore, in this work, we explore a novel methodology for CDR using the cutting-edge Transformer architecture. Specifically, we propose a novel framework for CDR, named as user group CDR. In contrast to conventional CDR objectives that predicts single user's ratings, we propose to calculate a group of users' information and parallelly compute and predict. In this framework, the computational efficiency can be exponentially reduced. Besides, we present a novel and neat encoder-decoder architecture for CDR based on Transformer. Other than existing approaches that are either hard to train or have sophisticated modules, our method is simple, direct, and efficient. We follow the Occam's razor principle that has been verified in most state-of-the-art methods [5-8] in natural language processing (NLP). The input of our model is a group of users with their rating patterns. Then, we use encoder to encode these patterns into dense representations in low dimensional spaces, this tackles the sparsity problem in certain degree. Next, several Transformer blocks are applied to exploit the rating pattern relationships among the group of users. In this way, the user preference in the source domain can be readily learned by the attention mechanism. At last, the decoder reconstructs the original rating sequences to get the final predictions of the whole group. Our methodology achieves promising and competitive results without bells and whistles. Our contributions are summarized as follows:

1) We propose a novel user group enhanced methodology for CDR, which paves a brand-new road for deep learning methodology in CDR.

2) We propose a novel encoder-decoder framework for CDR based on Transformer, in which the relationships of different users can be learned.

[1]Code is available at: https://github.com/Younai2021/TransCDR

3) We conduct experiments on two datasets to evaluate our model and investigate the effectiveness of Transformer.

4) We provide possible future work directions on Transformer based CDR.

2. Related Work

2.1. Cross-domain recommendation systems

Generally, cross-domain recommendation can be categorized into matrix factorization (MF)-based methods and autoencoder (AE)-based methods. The main stream is MF-based methods, which aims to learn a user latent vector and an item latent vector and then combine them to get the final prediction by matrix multiplication or generalized matrix multiplication using neural networks. In essence, these methods must depend on an explicitly defined user-item interaction function to get the prediction. On the bright side, this reserves the interpretability of the model. However, the indispensable user-item interaction extremely restricts the expression ability of the model. The other category is AE-based methods which learns the user rating pattern by mapping the ratings in to a latent semantic space by an encoder and reconstruct the ratings to the original space by a decoder. These methods can better learn the users' overall rating preferences compared to MF-based methods. However, how to learn the semantic information in the latent space remains intact.

2.2. Transformer

Currently, Transformer [9] has been applied in numerous domains, such as natural language processing [6], visual recognition, object detection. However, Transformer remains intact in cross-domain recommendation systems. Impressed by the transformer's powerful ability of learning the relationships among word sequences, we decide to find a way to apply Transformer architecture in CDR to learn the semantic information in the latent rating representations. Inspired by recent Transformer applications on many different domains, we find a way to use Transformer in CDR, that is collecting a group of user ratings and considering them as a sequence, then input them into the Transformer blocks to get the interacted user rating representations, and then reconstruct the latent representation to the original form to get the prediction. This methodology is common in Transformer based methods but still untouched in the CDR domain. Therefore, we explore the Transformer architecture in CDR for the first time.

3. TransCDR

3.1. Problem Definition

In this paper, we assume that the input data takes the form of explicit feedbacks such as user ratings of items. We also assume that the source and target domains have the same set of users, denoted by $\mathcal{U} = \{1,2,\dots,U\}$. The item sets from the source and target domains are $\mathcal{I}^S = \{1,2,\dots I^S\}$ and $\mathcal{I}^T = \{1,2,\dots I^T\}$, where S and T are the item amount of source and target domains respectively. The corresponding rating matrices are given by $\mathcal{Y}^S = \{y_{ui}^S | u \in \mathcal{U}, i \in \mathcal{I}^S\}$ and $\mathcal{Y}^T = \{y_{ui}^T | u \in \mathcal{U}, i \in \mathcal{I}^T\}$. We denote the set of observed item ratings given by u as $\mathcal{I}_u^S$ ($\mathcal{I}_u^T$), and the unobserved ones as $\overline{\mathcal{I}_u^S}$ ($\overline{\mathcal{I}_u^T}$) for source (target) domain. In each domain, the goal of recommendation can be specified as selecting a subset of items from $\overline{\mathcal{I}_u^S}$ ($\overline{\mathcal{I}_u^T}$) for user u according to certain criteria that maximizes the user's satisfaction. In other words, the recommendation model aims to give predictions on the unknown ratings of each user, i.e., $\hat{y}_{ui}^S(u \in \mathcal{U}, i \in \overline{\mathcal{I}_u^S})$ or $\hat{y}_{ui}^T(u \in \mathcal{U}, i \in \overline{\mathcal{I}_u^T})$. The values of $\hat{y}_{ui}^S$ ($\hat{y}_{ui}^T$) are numerical ratings in a certain range, e.g. [1,5]. The recommendation task we consider here is the single domain problem, where we predict $\hat{y}_{ui}^T(u \in \mathcal{U}, i \in \overline{\mathcal{I}_u^T})$, via leveraging the information from the source rating matrix $\mathcal{Y}^S$. We use widely-adopted metrics to measure the performance of all approaches used in this paper, such as RMSE defined by $\sqrt{(1/MN)\sum_{u=1}^{M}\sum_{i=1}^{N}(\hat{y}_{ui}^T - y_{ui}^T)^2}$, where M, N denote the number of users and items in the testing set respectively.

3.2. User Group Enhancement for CDR

Traditional CDR models are trained and tested on single user, which is not a propriate formulation for deep learning methodology. Therefore, in this paper, we firstly propose to train and predict ratings on a user group scale. As shown in Fig.1, the input of the model is a user group, denoted by $\mathcal{G} = \{u_1, u_2, \dots u_k\}$, where k is the group size. Then, the ratings of each user on two domains are selected by the user index and concatenated as the input the model. Accordingly, the output of the model is the user group ratings on two domains, denoted by $\hat{\mathcal{R}} = \{\hat{\mathcal{R}}^S, \hat{\mathcal{R}}^T\} = \{\hat{\mathcal{I}}_1^S, \hat{\mathcal{I}}_2^S, \dots, \hat{\mathcal{I}}_k^S, \hat{\mathcal{I}}_1^T, \dots \hat{\mathcal{I}}_k^T\}$, where $\hat{\mathcal{I}}_i^S$ is the i-th user's ratings in the source domain and $\hat{\mathcal{I}}_i^T$ is the ratings in the target domain. By redefine the task in a group scale, the models can be trained more efficiently and the relationships of different users can be learned by the model. In the training stage $\hat{\mathcal{R}}$ is optimized while in the inference stage, only $\hat{\mathcal{R}}^T$ is used for prediction in the target domain.

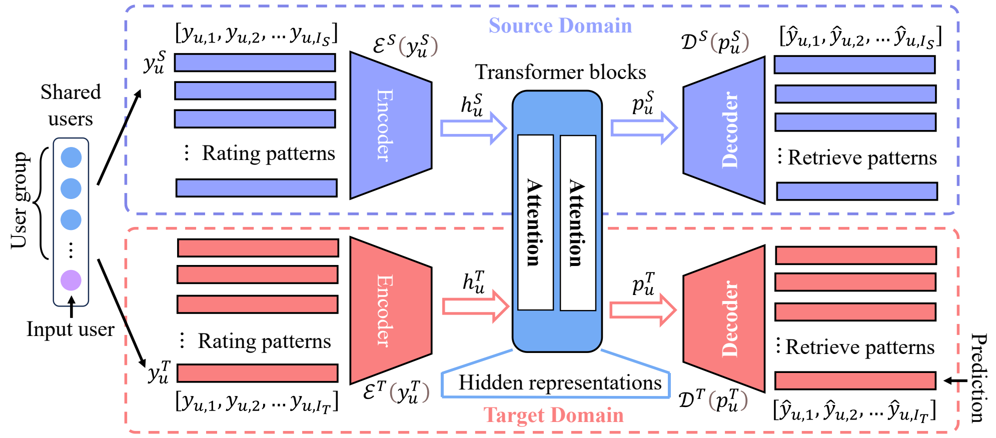

Figure 1: Overview of TransCDR. The input is a user group. First, the rating patterns of each user on both source domain and target domain is feed into two domain specific encoders. Then, the hidden representations of the rating patterns are processed by the Transformer blocks. Finally, the processed hidden representations are retrieved into original space by two domain specific decoders. The output of decoder is the predicted ratings of each user in the user group.

3.3. Transformer Framework for CDR

Based on user group formulation, we propose a novel encoder-decoder framework for CDR based on Transformer. Autorec [10] explored autoencoder framework for collaborate filtering (CF). We believe that autoencoder (AE) framework is a promising alternative against widely adopted matrix factorization (MF) framework. Therefore, in this paper, we further explore the AE framework with up-to-date deep learning architectures. We only focus on the User autoencoder in this work and defer Item autoencoder in the future work.

Encoder. First, the source and target domain encoders learn a set of embeddings to represent each user's preferences. The encoders take the partially observed rating vectors for each user, i.e., y_u^S and y_u^T as input and maps each vector in to low-dimensional dense latent space where the property and relationships of patterns can be better revealed than in the original sparse space. The encoders can be formulated as:

$$h_u^S = \mathcal{E}^S(y_u^S) = \sigma(W_1 y_u^S + b_1) \quad (1)$$

$$h_u^T = \mathcal{E}^T(y_u^T) = \sigma(W_2 y_u^T + b_2) \quad (2)$$

where $\mathcal{E}^S(\bullet)$ and $\mathcal{E}^T(\bullet)$ denote source domain encoder and target domain encoder. $\sigma(\bullet)$ is the activation function. h_u^S and h_u^T are the embeddings of rating patterns. The embeddings can be either trained using Autorec or directly trained end-to-end in the TransCDR.

Transformer Blocks. Transformer blocks is the central part of TransCDR. We use the vanilla Transformer architecture [9] in this work and leave the better modifications, such as deformable Transformer [11], ConViT [12], as a future work. The input of Transformer is the rating embedding sequence of a group of users in two domains, which is represented as:

$$H = [h_1^S, h_2^S, \dots h_k^S, h_1^T, h_2^T, \dots, h_k^T] \quad (3)$$

Transformer is constructed by stacking M blocks. Each block comprises a multi-head self-attention (MSA) module and a multi-layer perception (MLP) module, with a layer norm (LN) operation and skip connection added between the two modules. Self-attention (SA) is defined as:

$$SA(H^l) = softmax\left(\frac{H^l W_Q (H^l W_K)^T}{\sqrt{s}}\right)(H^l W_V), \quad (4)$$

where $W_Q, W_K,$ and $W_V \in \mathbb{R}^{d\times d}$ represent the query matrix, the key matrix, and the value matrix. X^l is the output of the l-th Transformer layer. s is part of the scaling factor $\frac{1}{\sqrt{s}}$. In SA, s equals the dimension d of the tokens. MSA is an extension of SA with h self-attention operations, which are called heads. In MSA, s is typically set to $\frac{d}{h}$. Therefore, MSA can be formulated as:

$$MSA(H^l) = [SA_1(H^l); SA_2(H^l); \cdots; SA_h(H^l)]W_P, \quad (5)$$

After defining MSA, the operations of a Transformer block can be expressed as:

$$\tilde{H}^{l-1} = MSA\big(LN(H^{l-1})\big) + H^{l-1}, \quad (6)$$

$$H^l = MLP\left(LN\big(\tilde{H}^{l-1}\big)\right) + \tilde{H}^{l-1}. \quad (7)$$

The MLP module is constructed by two linear projections, with a activation function in between.
After the last Transformer layer, we can get the processed rating patterns of each user, denoted as:

$$P = [p_1^S, p_2^S, \dots p_k^S, p_1^T, p_2^T, \dots, p_k^T]. \quad (8)$$

Decoder. In order to get the final rating predictions of each user. We need to reconstruct the hidden representations into the original rating space. This is

Table 1: Statistics of the datasets.

No.	Year	Dataset		User#	Item#		Ratings#			Sparsity	
		Source	Target	Shared	Source	Target	Size	Source	Target	Source	Target
1	2014	Automotive	Toys and Games	1637	11905	16070	3.63M	1373768	2252771	99.91%	99.91%
2	2018	Movies and TV	Office Products	-	-	-	14.35M	8765568	5581313	-	-
2 (pruned)	2018	Movies and TV	Office Products	3026	10465	8303	5M	2500000	2500000	99.84%	99.89%

achieved by two domain specific decoders. Similar to the encoders, they can be formulated as:

$$\hat{y}_u^S = \mathcal{D}^S(p_u^S) = \sigma(W_3 p_u^S + b_3) \tag{9}$$

$$\hat{y}_u^T = \mathcal{D}^T(p_u^T) = \sigma(W_4 p_u^T + b_4) \tag{10}$$

where $\mathcal{D}^S(\bullet)$ and $\mathcal{D}^T(\bullet)$ denote source domain encoder and target domain encoder. $\hat{y}_u^S$ and $\hat{y}_u^T$ are the rating predictions on source and target domains of the u-th user. In the inference stage, only the $\hat{y}_u^T$ is used for evaluation.

The overall architecture of Transformer-based autoencoder framework can be concisely summarized as:

$$\hat{y}^T = \mathcal{D}^T\left(\mathcal{T}\left(\mathcal{E}^S(y) \oplus \mathcal{E}^T(y)\right)\right) \tag{11}$$

where $\mathcal{E}(\bullet)$ is the encoder. $\mathcal{T}(\bullet)$ denotes Transformer blocks. $\oplus$ is the matrix concatenate operation. $\mathcal{D}(\bullet)$ is the decoder.

4. Experiments

4.1. Experimental Settings

Datasets. In this section, we evaluate our model on two subsets from Amazon review dataset 2014 [13, 14] and Amazon review dataset 2018 [15] with shared users in different categories. We define different item categories as domains, where we select users with at least 5 ratings (5-core). We select categories that are relatively irrelevant. The details of two datasets are shown in Table 1. Statistics show that both the source and target domain are extremely sparse with most ratings are unobserved. Since the original dataset 2 is too large and it takes over 2 hours for preprocessing, we must compromise to use the pruned version with smaller data. The original data files are preprocessed using the code from DARec [16] to get the shared users in source and target domain.

Implementation details. The batch size is set to 7 and the group size is set to 200. We train our model for 200 epochs. We use Adam optimizer with a weight decay of 1e-5. The learning rate is set to 0.001. The embedding size is set to 200. In pretrained embedding scenarios, we train the encoder from AutoRec. The encoder is trained with 40 epochs when the loss is stable. For Transformer hyperparameters, we set the depth to 3, head number to 5, MLP ratio to 4, and the hidden dimension to 200. We train our model on one NVIDIA RTX2060 GPU on a laptop devise. In the inference stage, the user group size is set to 40 by default.

Table 2: Comparison with baseline CDR models on RMSE.

Methods	Dataset	
	Automotive & Toys and Games	Movies and TV & Office Products (pruned)
AutoRec	3.4668	2.1704
U-DARec	3.4533	2.1486
TransCDR (ours)	**3.4393**	**2.1319**

4.2. Comparison with Baselines

We compare our approach with several relevant state-of-the-art CDR models.

- **AutoRec [10]:** AutoRec uses autoencoder to perform the unknown rating from the observed ones.
- **CoNet [17]:** CoNet is a transfer learning method that enables dual knowledge transfer across domains by introducing cross connections from one base network to another and vice versa.
- **DARec [16]:** DARec is a deep domain adaptation model that is capable of extracting and transferring patterns from rating matrices only and without relying on any auxillary information.

For AutoRec and DARec, we use an opensource implementation[2]. We use the default configurations for time efficiency. For CoNet, we didn't conduct experiments due to limited time. Table 2 shows the RMSE results of our method and compared methods. We can observe that our method outperforms AutoRec and DARec on two datasets, which shows the capacity of our method.

Optimization performance. In order to verify that our model is adequately trained and not overfitted on the dataset, we record loss curve during training. Results in Fig. 2 show that our model is appropriately trained on both datasets.

[2] https://github.com/Yu-Fangxu/DARec

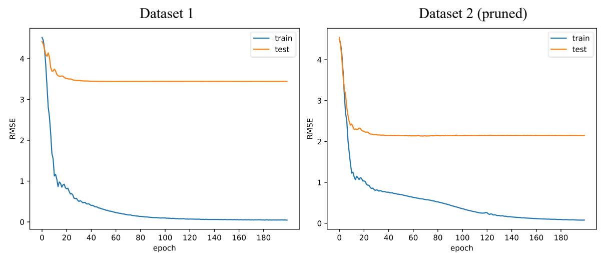

Figure 2: Loss curve during training.

4.3. Ablation Studies

In this section, we conduct abundant ablation studies on our model parameters and analyze their contribution to the model. Experiments are conducted on the first dataset by default.

Effect of Transformer layers. We set different number of Transformer layers to investigate the effect of the depth of our model. We conduct experiments on two datasets. First, we remove all Transformer blocks and directly use the embeddings as the input of the decoder. As shown in Table 3, compared to the standard model, removing the Transformer blocks significantly harms the performance. Therefore, the Transformer layers is necessary and effective as the core of TransCDR. Moreover, we further investigate the number Transformer layers. Results in Fig.3 show the tendency that deeper network leads to better performance.

Table 3: Ablation study on Transformer blocks.

Transformer Blocks	Dataset	
	Automotive & Toys and Games	Movies and TV & Office Products (pruned)
×	3.4710	2.1854
✓ (3 by default)	3.4393	2.1319

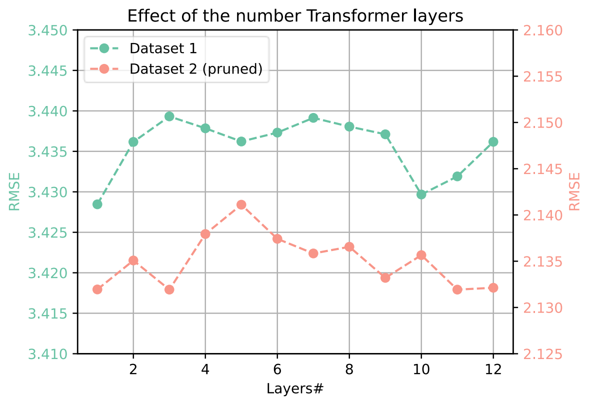

Figure 3: Effect of the number Transformer layers.

Effect of group size. Group size (i.e., the number of users inputted into the Transformer in a single time) is the maximum token number (MTN) in to the Transformer input. MTN effects the computation complexity of self-attention (SA) in Transformer, with is the major computational cost of the model. The computational complexity is quadratic to MTN [18]. This relationship is formulated as:

$$\Omega(SA) = 4kD^2 + 2k^2D. \qquad (12)$$

where k denotes maximum token number and D is the embedding dimension. Normally, in NLP or computer vision, the maximum token number is set to about 200. Therefore, in our experiments, we test k from 50 to 400 and record the training time in each input. Results in Fig.4 show that there is no significant discrepancy in different group size, but the training is monotonously increasing as the group size increases.

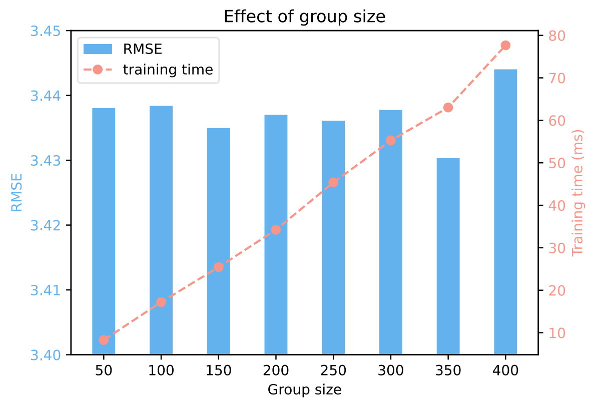

Figure 4: RMSE on different group size and corresponding training time consumption.

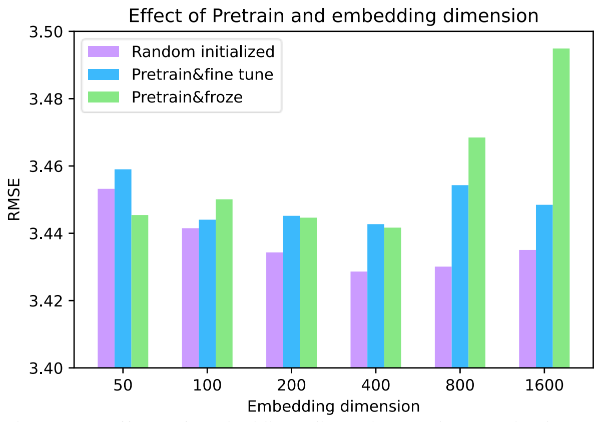

Figure 5: Effect of embedding dimension and pre-trained user embeddings.

Ablation studies on user embeddings. We investigate the embedding settings of TransCDR. We conduct experiments on two aspects: 1) Effect of pretraining: the embeddings can be either trained using Autorec or directly trained end-to-end in the TransCDR. 2) Effect of the dimension of embeddings. Results in Fig.5 show that pretraining the encoder from AutoRec brings no improvement to the model. The best result is yielded

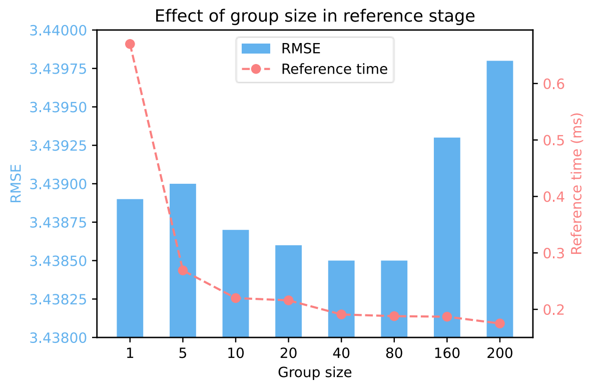

Figure 6: RMSE on different user group size in inference and inference time.

when the embedding size is 400 without pretraining. The model is better to be trained in an end-to-end manner to reserve the consistency of embeddings and hidden representations.

Effect of user group size in inference. In the inference stage, the input of our model could be a group of users or a single user with reference users. When the user ratings are predicted in groups, the inference time on per user reduces. Figure 6 shows the results of different group size and corresponding inference time per user. In this experiment, the maximum token number (MTN) is set to 200, which means the maximum group size is limited to 200. The best results are gained when the inference group size is set to 40~80. The main reason is that the relationships of users can be better learned in this setting.

4.4. Visualization

We visualize the similarities of the user rating patterns to reveal the similarity and relationships of the users. Specifically, we use t-SNE to map several different high-dimensional rating representations of users in the testing dataset 1. Results are shown in Fig.7. The top left one is the visualization of original ratings of each user. We can observe that in this representation, each user is an isolated point, which means there are no similarities among users. The top right one is from the AutoRec encoded representation. Results show that although the users in this representation have some similarities, but it is hard to find cluster patterns, which indicate the similar user rating patterns. In contrast, in our encoded embeddings in the bottom left corner, this cluster patterns are more obvious. Moreover, the processed hidden rating patterns of our model in the bottom right corner show the strongest cluster properties. This means that our method can better encode the user rating patterns in similarities and process the relationships of different users.

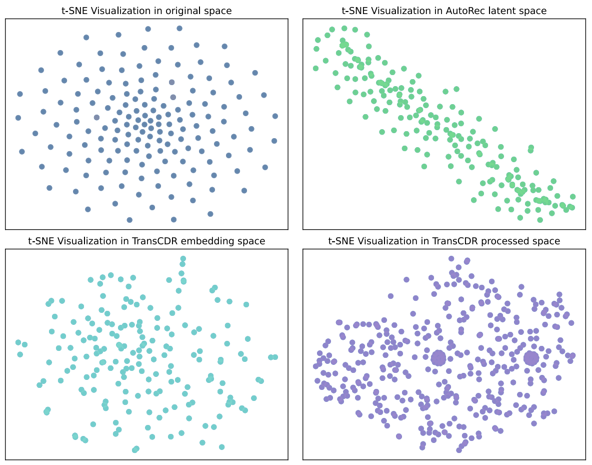

Figure 7: t-SNE visualization of user rating patterns in different representations.

5. Conclusion

In this paper. We introduced a novel group enhanced methodology for CDR that calculate a group of users' ratings in parallel. Besides, we explored Transformer framework for cross-domain recommendation for the first time. Concretely, the proposed TransCDR encodes the user ratings into the latent space embeddings. Then, the semantic information of rating representations is learned by Transformer. Finally, the decoder reconstructs the ratings back to the original rating space for prediction. Abundant experiments and visualizations verify our methodology for CDR. We hope our initial work of group enhanced methodology and Transformer framework will inspire and facilitate future research for CDR.

6. Future Work

Due to the limited time, we provide three possible directions for future work to improve the naïve TransCDR, listed as follows:

1) Introducing Graph clusters of users. In this paper, the user group is randomly selected from the dataset. Therefore, in some cases, the selected group may cannot include similar rating patterns. If the user group is sampled from strongly related users, the rating patterns can be better revealed by the attention mechanism in the Transformer layers.

2) Adding a classifier to specify the source domain and target domain. In this paper, we did not add constraints on the output of the source domain and target domain, which means the domain information is not explicitly learned during the training stage. Therefore, it might help to add a classifier after the output hidden representations of the two domains and specify the domain as a classification subtask. We believe that training with the domain information will improve the performance of TransCDR.

3) Explore item autoencoder and deep autoencoder. Autoencoder framework is the main ideology of this work. We believe that a good autoencoder architecture can better represent the rating patterns thus better reveal the property of pattern. In some scenarios, item based autoencoder may have better performance. Besides, deeper autoencoder may have stronger capacity of learning the latent representation of rating patterns.

References

[1] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proceedings of the 26th international conference on world wide web*, 2017, pp. 173-182.

[2] X. Wang, X. He, M. Wang, F. Feng, and T.-S. Chua, "Neural graph collaborative filtering," in *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*, 2019, pp. 165-174.

[3] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "Lightgcn: Simplifying and powering graph convolution network for recommendation," in *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, 2020, pp. 639-648.

[4] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "Bayesian personalized ranking from implicit feedback," in *Proc. of Uncertainty in Artificial Intelligence*, 2014, pp. 452-461.

[5] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," 2018.

[6] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:.04805,* 2018.

[7] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," *OpenAI blog,* vol. 1, p. 9, 2019.

[8] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal*, et al.*, "Language models are few-shot learners," *Advances in neural information processing systems,* vol. 33, pp. 1877-1901, 2020.

[9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez*, et al.*, "Attention is all you need," *Advances in Neural Information Processing Systems,* vol. 30, 2017.

[10] S. Sedhain, A. K. Menon, S. Sanner, and L. Xie, "Autorec: Autoencoders meet collaborative filtering," in *Proceedings of the 24th international conference on World Wide Web*, 2015, pp. 111-112.

[11] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable detr: Deformable transformers for end-to-end object detection," *arXiv preprint arXiv:2010.04159,* 2020.

[12] S. d'Ascoli, H. Touvron, M. L. Leavitt, A. S. Morcos, G. Biroli, and L. Sagun, "Convit: Improving vision transformers with soft convolutional inductive biases," in *International Conference on Machine Learning*, 2021, pp. 2286-2296.

[13] R. He and J. McAuley, "Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering," in *proceedings of the 25th international conference on world wide web*, 2016, pp. 507-517.

[14] J. McAuley, C. Targett, Q. Shi, and A. Van Den Hengel, "Image-based recommendations on styles and substitutes," in *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*, 2015, pp. 43-52.

[15] J. Ni, J. Li, and J. McAuley, "Justifying recommendations using distantly-labeled reviews and fine-grained aspects," in *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, 2019, pp. 188-197.

[16] F. Yuan, L. Yao, and B. Benatallah, "DARec: Deep domain adaptation for cross-domain recommendation via transferring rating patterns," *arXiv preprint arXiv:.10760,* 2019.

[17] G. Hu, Y. Zhang, and Q. Yang, "Conet: Collaborative cross networks for cross-domain recommendation," in *Proceedings of the 27th ACM international conference on information and knowledge management*, 2018, pp. 667-676.

[18] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang*, et al.*, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10012-10022.