# A Related Work

## A.1 Sequential Recommender System

Sequential recommender systems (SR) mainly leverage users' interaction data in a sequential manner to predict the next behavior. Recurrent Neural Networks (RNNs) are widely applied to capture the order relationship within behavior sequences. GRU4REC [5,18] uses multi-layer GRU to model session-parallel mini-batches. NARM [11] proposes the neural attentive recommendation machine to learn the global and local representation in the current session. HRNN [16] proposes a hierarchical RNN model to extract latent hidden user states across user's historical sessions. On the top of RNNs, STAMP [13], SASRec [8] and BERT4Rec [17] incorporate attention mechanisms to capture more accurate preference from longer sequences. [20] and [28] propose convolutional sequence embedding based recommendation model. [19] builds a model that can make use of different temporal ranges and dynamics depending on the request context. Moreover, [25] and [26] introduce graph neural networks in session-based recommendation. And long-term stable preferences from users' historical behaviors are also considered in [12,15,29,27]. Inspired by those models, many industrial EBR systems learn user representations from sequential user behaviors, such as YouTubeDNN [2] and RRN [24] from Google, TDM [31], SDM [14], ComiRec [1] and MIND [10] from Alibaba. The user representations are then used for fast nearest neighbors search with item representations from a very large corpus. However, all of these SR models and EBR systems do not consider the unclicked historical behaviors as model input in the whole user-item exposure data. In this paper, we explicitly model them for better understanding of users' preference.

## A.2 Metric Learning based Recommendation

The goal of the metric learning algorithm is to find a more effective feature representation subspace of input data. It learns an alternative distance measurement which is used to optimize the performance of the model. When it's applied to recommendation domains, [7] propose Collaborative Metric Learning (CML), which learns a joint metric space to encode not only users' preferences but also the user-user and item-item similarity. Similar metric-based models can be found in [3,4], which use Euclidean distances for modeling transitional patterns. [22] utilize Mahalanobis distance-based metric learning algorithm for automatic playlist continuation of music. To overcome the limitation of geometrical congestion and instability in CML, [21] propose Latent Relational Metric Learning (LRML) to learn a latent relation vector for each given user-item pair. To jointly model heterogeneous user behaviors, different novel metric learning methods are proposed in [9] and [30]. These models criticize that existing score learning methods cannot correctly reflect user-user and item-item similarities in their latent spaces. The triplet metric has the good modeling ability of multivariate relations [6]. Therefore, in this paper, we use metric learning via triplet loss structure to encode clicked and unclicked behavior sequences simultaneously.

**Table 1.** Triplet structure experiments on Taobao dataset.

| models | HR@50 | MRR@50 | R@50 | $F_1$@50 |
|---|---|---|---|---|
| XDM (unclk&lab) | 36.69% | 8.79% | 1.83% | 1.43% |
| XDM (lab&clk) | 36.91% | 9.29% | 1.85% | 1.45% |
| XDM (unclk&clk) | 37.34% | 9.40% | 1.86% | 1.46% |
| XDM (w/o sym) | 37.37% | 9.29% | 1.87% | 1.47% |
| XDM | **37.97%** | **9.41%** | **1.92%** | **1.50%** |

# B Experiment Details

## B.1 Data Collection

During data collection, one crucial issue is to prepare unclicked data, which is a kind of implicit feedback in our experimental platforms. They contain a large number of noises or uninformative signals, while they are the vast majority of all user behaviors. So we propose to use the following pre-defined filtering rules for selecting unclicked data with less noises

1. Select latest impressed items of a user that were not clicked within the past three days as unclicked candidates.
2. Only keep those unclicked items that are exposed to a user more than $k$ times (set $k = 1$ in this work) in the above unclicked candidates.

Items impressed to a user many times yet not clicked indicate that his/her preference at this stage is not strong. We choose those items naturally to form the unclicked sequence for each user. The maximum length of unclicked sequences was set to 100 for the sake of effective sequential modeling.

# C EXPERIMENT ANALYSIS

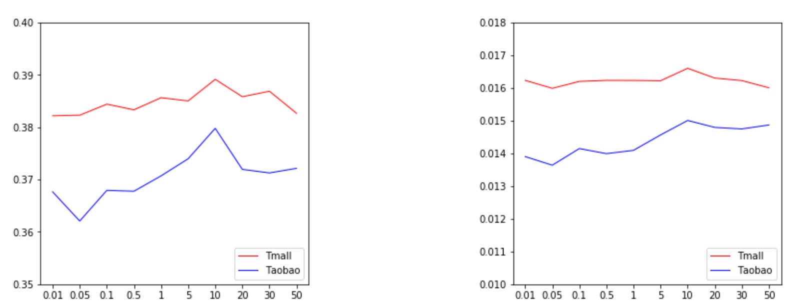
## C.1 Role of Metric Learning

Comparing the result of *XDM (w/o metric)* with *XDM (w/o sym)* and *XDM*, we can observe that the introduction of metric learning module improves the performance of *XDM* significantly. In this section, we will analyze it in detail. Metric learning algorithm is proposed to model unclicked sequences more appropriately with considering its relationship with clicked sequences. The predefined margin $m^*$ comes into play in a way of threshold limit, and the hyper-parameter $\lambda$ controls the importance ratio between different loss terms. Note that the goal of experiments below is to show how model performance is affected by these experimental variables. It does not mean we optimize our model on test dataset, though results are reported on test.

**The effect of triplet structure** In order to explore the effectiveness of the triplet structure, we only kept one of the three feature relationship in the triplet loss by replacing triplet with pair loss: minimizing the distance between label and clicked representations (lab&clk), maximizing label and unclicked representations (unclk&lab), and maximizing unclicked and clicked representations (unclk&clk). Methods using asymmetric triplet loss (*XDM (w/o sym)*) and using symmetric triplet loss (*XDM*) are included as comparative experiments.

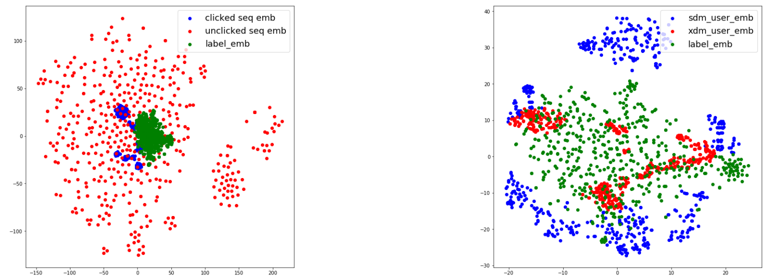
From Table 1, we can conclude that pair structure optimization with unclk&clk is better than the others (lab&clk and unclk&lab). It shows the introduction of unclicked sequences is effective and should be handled carefully, otherwise causing negative effect to the model. Comparing *XDM (w/o sym)* with the three pair experiments, it's clear that the triplet structure can model unclicked data well and produce a positive effect. This also conforms the necessity of importing triplet metric. In addition, comparing *XDM (w/o sym)* with *XDM* we can infer the important role of symmetric constraints. To sum up, the triplet loss is effective for unclicked sequence modeling, and the added symmetric constraint makes the model learn a better user representation. The main reason why the triplet loss works well is that a dynamic balance is achieved by using metric learning that controls the distance among three representations.

**The effect of hyper-parameter** The loss trade-off coefficient $\lambda$ in *XDM* is an important hyper-parameter, which acts between the cross-entropy loss and the triplet loss to adapt different scale of two losses to a suitable range. Thus $\lambda$ can directly determine the importance of the triplet loss in the model learning. A comparison experiment in Figure 1 was performed on changes caused by $\lambda$, where parameter experiments followed the same rule as parameter $m^*$. The parameter values are taken from a subdivided set $\{0.01, 0.05, 0.1, 0.5, 1, 5, 10, 20, 30, 50\}$. We find that the effect gets obvious when the parameter $\lambda$ is greater than 1, and reach the optimal at point 10.




(a) Effect of $\lambda$ on HR@50 (b) Effect of $\lambda$ on $F_1$@50

**Fig. 1.** The effect of equilibrium coefficient $\lambda$ which acts between cross-entropy loss and triplet loss.




(a) Features visualization in triplet loss (b) User representation visualization

**Fig. 2.** (a) Distribution of feature representation involved in triplet loss calculation; (b) User representation visualization generated by *XDM* and *SDM*.

### C.2 Visualization

We are interested in whether the feature space is appropriate and whether the feature expression can correctly reflect different preferences. Hence we did t-SNE [23] visualization for different types of feature representations which were reduced to a two-dimensional space, and explained the adaptability of feature space for *XDM*. We visualized various feature representations, *i.e.,* $\boldsymbol{h}_{u,t}$, $\boldsymbol{n}_{u,t}$, and $\boldsymbol{c}_{u,t}$ in the calculation of the triplet loss, as shown in Figure 2(a). The final user representations $\hat{\boldsymbol{z}}_{u,t}$ and $\boldsymbol{h}_{u,t}$ obtained from *XDM* and *SDM* respectively are presented in Figure 2(b). Figure 2(a) illustrates that the representations of clicked sequences and labels are closely clustered, and unclicked ones keep a certain distance around them. This result fits our optimization goal and makes the feature fusion in *XDM* more meaningful. Figure 2(b) shows that compared with *SDM*, user representation obtained from *XDM* is more closely clustered with labels. *XDM* can learn a better sequential recommender by introducing unclicked sequences.

### C.3 Case Study

In order to show the real performance of our model, as well as to explain what characteristics the unclicked data reflects, real cases from the Taobao dataset were analyzed in this section. We sampled a representative user and displayed her behaviors and top prediction results. Since it is hard to say whether filtering out all information contained in unclicked sequences will actually lead to actual online metric gains, we only compare top prediction results in this case. In Figure 3, the user's clicked item sequence concentrates on women's wear and children's wear, while the unclicked sequence concentrates mostly on shoes. From this point of view, the user does not have a strong preference for shoes currently, although she clicked shoes items in her clicked sequence. So we hope that the model can learn this negative interest and does not recommend shoes at least in top positions. Comparing recommendation results of *XDM* and *SDM*, we find that *XDM* filters out shoes items well, while *SDM* still give priority to those items. It can be seen that noises in clicked behaviors could be eliminated by incorporating unlicked item sequences. Finally, comparing with the user's ground truth behaviors, we find it basically coincides with the result of *XDM* at category level, which indicates the importance of modeling unclicked items and the validity of our model.


**Fig. 3.** Items on the top are a user's historical behaviors, including clicked and unclicked sequences. Items on the bottom are top prediction results of XDM and SDM, and real behaviors (labels). Items in SDM results with red dashed box are similar to unclicked items.

## References

1. Cen, Y., Zhang, J., Zou, X., Zhou, C., Yang, H., Tang, J.: Controllable multi-interest framework for recommendation. In: SIGKDD. pp. 2942–2951 (2020)
2. Covington, P., Adams, J., Sargin, E.: Deep neural networks for youtube recommendations. In: RecSys. pp. 191–198 (2016)
3. Feng, S., Li, X., Zeng, Y., Cong, G., Chee, Y.M., Yuan, Q.: Personalized ranking metric embedding for next new poi recommendation. In: IJCAI (2015)
4. He, R., Kang, W.C., McAuley, J.: Translation-based recommendation. In: RecSys. pp. 161–169 (2017)
5. Hidasi, B., Karatzoglou, A., Baltrunas, L., Tikk, D.: Session-based recommendations with recurrent neural networks. arXiv preprint arXiv:1511.06939 (2015)
6. Hoffer, E., Ailon, N.: Deep metric learning using triplet network. In: International Workshop on Similarity-Based Pattern Recognition. pp. 84–92. Springer (2015)
7. Hsieh, C.K., Yang, L., Cui, Y., Lin, T.Y., Belongie, S., Estrin, D.: Collaborative metric learning. In: WWW. pp. 193–201 (2017)
8. Kang, W.C., McAuley, J.: Self-attentive sequential recommendation. In: ICDM. pp. 197–206 (2018)

9. Lee, D., Park, C., Ju, H., Hwang, J., Yu, H.: Action space learning for heterogeneous user behavior prediction. In: IJCAI. pp. 2830–2836 (2019)
10. Li, C., Liu, Z., Wu, M., Xu, Y., Zhao, H., Huang, P., Kang, G., Chen, Q., Li, W., Lee, D.L.: Multi-interest network with dynamic routing for recommendation at tmall. In: CIKM. pp. 2615–2623 (2019)
11. Li, J., Ren, P., Chen, Z., Ren, Z., Lian, T., Ma, J.: Neural attentive session-based recommendation. In: CIKM. pp. 1419–1428 (2017)
12. Li, Z., Zhao, H., Liu, Q., Huang, Z., Mei, T., Chen, E.: Learning from history and present: Next-item recommendation via discriminatively exploiting user behaviors. In: KDD. pp. 1734–1743 (2018)
13. Liu, Q., Zeng, Y., Mokhosi, R., Zhang, H.: Stamp: short-term attention/memory priority model for session-based recommendation. In: KDD. pp. 1831–1839 (2018)
14. Lv, F., Jin, T., Yu, C., Sun, F., Lin, Q., Yang, K., Ng, W.: Sdm: Sequential deep matching model for online large-scale recommender system. In: CIKM. pp. 2635–2643 (2019)
15. Ma, C., Kang, P., Liu, X.: Hierarchical gating networks for sequential recommendation. In: KDD. pp. 825–833 (2019)
16. Quadrana, M., Karatzoglou, A., Hidasi, B., Cremonesi, P.: Personalizing session-based recommendations with hierarchical recurrent neural networks. In: RecSys (2017)
17. Sun, F., Liu, J., Wu, J., Pei, C., Lin, X., Ou, W., Jiang, P.: Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In: CIKM. pp. 1441–1450 (2019)
18. Tan, Y.K., Xu, X., Liu, Y.: Improved recurrent neural networks for session-based recommendations. In: Proceedings of the 1st Workshop on Deep Learning for Recommender Systems. pp. 17–22 (2016)
19. Tang, J., Belletti, F., Jain, S., Chen, M., Beutel, A., Xu, C., H Chi, E.: Towards neural mixture recommender for long range dependent user sequences. In: WWW. pp. 1782–1793. ACM (2019)
20. Tang, J., Wang, K.: Personalized top-n sequential recommendation via convolutional sequence embedding. In: WSDM. pp. 565–573 (2018)
21. Tay, Y., Anh Tuan, L., Hui, S.C.: Latent relational metric learning via memory-based attention for collaborative ranking. In: WWW. pp. 729–739 (2018)
22. Tran, T., Sweeney, R., Lee, K.: Adversarial mahalanobis distance-based attentive song recommender for automatic playlist continuation. In: SIGIR. pp. 245–254 (2019)
23. Van Der Maaten, L.: Accelerating t-sne using tree-based algorithms. JMLR **15**(1), 3221–3245 (2014)
24. Wu, C.Y., Ahmed, A., Beutel, A., Smola, A.J., Jing, H.: Recurrent recommender networks. In: WSDM. pp. 495–503 (2017)
25. Wu, S., Tang, Y., Zhu, Y., Wang, L., Xie, X., Tan, T.: Session-based recommendation with graph neural networks. In: AAAI. vol. 33, pp. 346–353 (2019)
26. Xu, C., Zhao, P., Liu, Y., Sheng, V.S., Xu, J., Zhuang, F., Fang, J., Zhou, X.: Graph contextualized self-attention network for session-based recommendation. In: IJCAI. pp. 3940–3946 (2019)
27. Ying, H., Zhuang, F., Zhang, F., Liu, Y., Xu, G., Xie, X., Xiong, H., Wu, J.: Sequential recommender system based on hierarchical attention networks. In: IJCAI (2018)
28. You, J., Wang, Y., Pal, A., Eksombatchai, P., Rosenburg, C., Leskovec, J.: Hierarchical temporal convolutional networks for dynamic recommender systems. In: WWW. pp. 2236–2246 (2019)

29. Yu, Z., Lian, J., Mahmoody, A., Liu, G., Xie, X.: Adaptive user modeling with long and short-term preferences for personalized recommendation. In: IJCAI. pp. 4213–4219 (2019)
30. Zhou, X., Liu, D., Lian, J., Xie, X.: Collaborative metric learning with memory network for multi-relational recommender systems. arXiv preprint arXiv:1906.09882 (2019)
31. Zhu, H., Li, X., Zhang, P., Li, G., He, J., Li, H., Gai, K.: Learning tree-based deep model for recommender systems. In: SIGKDD. pp. 1079–1088 (2018)