

Symmetric Metric Learning with Adaptive Margin for Recommendation

Mingming Li,^{1,2} Shuai Zhang,³ Fuqing Zhu,^{*1} Wanhui Qian,^{1,2}
Liangjun Zang,¹ Jizhong Han,¹ Songlin Hu^{*1,2}

¹Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China

²School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China

³School of Computer Science and Engineering, The University of New South Wales, Australia

{limingming, zhufuqing, qianwanhui, zangliangjun, hanjizhong, husonglin}@iie.ac.cn, cheungshuai@outlook.com

Abstract

Metric learning based methods have attracted extensive interests in recommender systems. Current methods take the user-centric way in metric space to ensure the distance between user and negative item to be larger than that between the current user and positive item by a fixed margin. While they ignore the relations among positive item and negative item. As a result, these two items might be positioned closely, leading to incorrect results. Meanwhile, different users usually have different preferences, the fixed margin used in those methods can not be adaptive to various user biases, and thus decreases the performance as well. To address these two problems, a novel Symmetric Metric Learning with adaptive margin (SML) is proposed. In addition to the current user-centric metric, it symmetrically introduces a positive item-centric metric which maintains closer distance from positive items to user, and push the negative items away from the positive items at the same time. Moreover, the dynamically adaptive margins are well trained to mitigate the impact of bias. Experimental results on three public recommendation datasets demonstrate that SML produces a competitive performance compared with several state-of-the-art methods.

Introduction

Recommender systems have been widely deployed in web applications to address the information overload issue. The goal is to help users target their interests and provide personalized recommendations. Among various recommendation methods for implicit feedback, BPR (Rendle et al. 2009) has been proved to be most successful, which learns the discriminative latent factors for users and items by factorizing user-item interaction matrix. Nevertheless, BPR suffers from a crucial drawback stemmed from the inner product that violates the triangle inequality (Ram and Gray 2012).

To this end, several metric learning based recommendation methods (Feng et al. 2015; Khoshneshin and Street 2010; Chen et al. 2012; Hsieh et al. 2017; Zhang et al. 2018) have been proposed to measure the similarity of user-item with Euclidean distance for satisfying the triangle inequality. Specifically, CML (Hsieh et al. 2017), one of the

most representative work, adopts the triple loss (Weinberger and Saul 2009; Song et al. 2017; Do et al. 2019) to ensure the distance between user and negative item (user dislikes) is larger than that between the current user and positive item (user likes) by a fixed margin of $m > 0$. That is, $d(u, v) + m \leq d(u, v^-)$ ¹. The principle assumption behind above operation is that a user might be closer to the items she likes than to those she does not. It is worth noting that other variants of CML, such as relational metric learning based methods (Tay, Tuan, and Hui 2018; Park et al. 2018; Zhou et al. 2019), are also optimized in this triple loss paradigm.

Despite the effectiveness of existing metric learning recommendation approaches, we argue that they have two inherent limitations:

Firstly, from a geometric perspective, above triple loss constraint can be regarded as a *user-centric* metric. In other words, they only consider the relationship of two edges, i.e., \overline{uv} (positive pair) and $\overline{uv^-}$ (negative pair), while ignoring the impact of the third edge of vv^- . As a result, the position of positive item may be close with the negative item, i.e., $d(u, v) > d(v, v^-)$. It violates the basic assumption of metric learning, compromising the performance of the recommender systems seriously. In the metric space, the basic assumption is that the distance measures the similarity. More concretely, short distance between a user and an item indicates that the user likes the item, and short distance between items means that they tend to co-occur. As such, $d(v, v^-)$ should not be smaller than $d(u, v)$, otherwise it will mislead the prediction of this item to other users. As shown in Figure 1(left), given a triple (u_1, v_1, v_2^-) , u_1 and u_2 have same preferences. In accordance with the constraint of CML, the learning process will be stopped when $d(u_1, v_1) + m \leq d(u_1, v_2^-)$. We can observe that the relative distance between positive item and negative item is too smaller. Thus, these two items are indistinguishable in the metric space. The learned model might make an incorrect prediction that u_2 likes the v_2 since u_2 likes v_1 and $d(u_2, v_1) \approx d(u_2, v_2)$.

Secondly, the above triple loss paradigm relies on a fix

¹ $d(u, v)$ denotes the distance of u and v .

*Corresponding author
Copyright © 2020, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

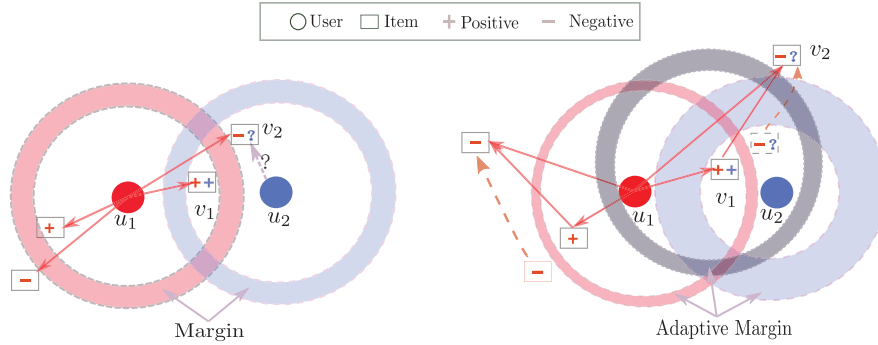


Figure 1: An illustration of metric learning. The left is CML, and the right is our proposed method SML. Item v_1 has been liked by both user u_1 and user u_2 .

margin m for all users, which may not perform well in real-world applications, especially when data distributions are complex. In recommendation scenarios, the number of interacted items varies widely for each user, and different users may also have different selection criteria called user bias. Therefore, the fixed margin setting limits the expressiveness of the model.

To circumvent above two issues, we propose a novel method called symmetric metric learning with adaptive margin (SML) for recommender systems. The illustration of SML is shown in Figure 1(right). The main idea is to measure the relationships of three edges (i.e., \overline{uv} , $\overline{uv^-}$, and $\overline{vv^-}$) from a symmetrical perspective, meanwhile taking into account the effects of user/item bias. The *user-centric* metric has considered the relationship of \overline{uv} and $\overline{uv^-}$, then we further present an *item-centric* metric that is symmetric with user-centric, to consider the relationship of \overline{vu} and $\overline{vv^-}$. The goal of the proposed symmetric metric learning is to guarantee the shortest edge is \overline{uv} so that we have $d(u, v) < d(v, v^-)$. Moreover, an adaptive margin strategy is designed to dynamically adjust the margins of users and items, in order to enhance the expressiveness of the model. Importantly, our paradigm could provide a novel angle for other communities, such as knowledge graph (Bordes et al. 2013; Wang et al. 2014). To summarize, this work makes the key contributions as follows:

- We present a novel symmetric metric learning method, which measures the trilateral relationship from the perspectives of both user-centric and item-centric effects.
- An adaptive margin mechanism is designed to dynamically adjust the margin to remit the problem of user/item bias.
- Extensive experiments are conducted on public datasets, demonstrating that the proposed method achieves competitive performances compared with state-of-the-arts.

Related Work

Metric learning (Weinberger and Saul 2009; Kulis and others 2013; Zadeh, Hosseini, and Sra 2016; Song et al. 2017) seeks an appropriate distance function for input points, such

as discrete distance, Euclidean distance, and Graph distance. They have been widely used in various fields, such as computer version (Liu et al. 2017; Wang et al. 2017; Zhang et al. 2019), recommender systems (Feng et al. 2015; Hsieh et al. 2017; Zhang et al. 2018), and knowledge graph (Bordes et al. 2013; Ji et al. 2015; Wang et al. 2014). Among them, triplet loss paradigm (Weinberger and Saul 2009) is one of the most competitive ways for ranking tasks, which makes up triplets to separate the positive pair from the negative pair by a distance margin.

In recommender systems, previous researchers focus on measuring the distance of user and item by Euclidean distance, which satisfies the conditions of non-negative, symmetry, and triangle inequality. For example, CML (Hsieh et al. 2017) illustrates the limitation of inner product similarity (Rendle et al. 2009) that only considers user-item relations. Moreover, it presents a push mechanism by using triplet loss. FML(Zhang et al. 2018) first converts the preference into distance, and then replaces the inner product with Euclidean distance. In essence, these methods aim at measuring the position of user and item in a metric vector space, and making the user’s clicked items close to this user. Inspired by the success of relational metric learning in knowledge graph (Wang et al. 2014), latent relational metric learning methods are also proposed for recommender systems. For example, TransCF (Park et al. 2018) constructs the user-item specific translation vectors by employing the neighborhood information of users and items, and then translates each user toward items according to the user’s relationships with these items. LRML (Tay, Tuan, and Hui 2018) is proposed to induce the latent relations by the memory-based attentive network. Compared with pure metric learning (CML), the latent relational metric learning methods improve the distance metric of user-item. Nevertheless, they still adopt the triple loss paradigm to measure the relative distance for a given triple (u, v, v^-) , facing the similar limitations mentioned in the previous section.

Proposed Method

In this section, we first describe the problem definition. Then, a brief review of CML is described to show what

problem it brings. After that, we discuss how to alleviate the above problem via the heuristic viewpoint. At last, we present a specific method named symmetric metric learning with adaptive margin (SML).

Problem Definition

In this paper, the top-K recommendation problem can be formulated as follows. There is a set of users \mathcal{U} and a set of items \mathcal{V} , and all user-item interactions are denoted as $\mathcal{I} = \{(u, v) | u \in \mathcal{U}, v \in \mathcal{V}\}$. $\mathcal{N}_u \subseteq \mathcal{V}$ denotes the set of items that user u has previously interacted with. Given above interaction information, we aim to predict whether user u has potential interest in items $v \in \mathcal{V} \setminus \mathcal{N}_u$ with which he has interaction before.

Limitation of CML

We first review the framework of CML (Hsieh et al. 2017) and then analyze its weakness. The key intuition of CML is that users and items could be placed in a low dimensional space and their explicit closeness could be measured by using Euclidean distance to satisfy the inequality property. For CML, the distance function for each user-item interaction can be defined as:

$$d(u, v) = \|\alpha_u - \beta_v\|_2^2, \quad (1)$$

where $\|\cdot\|_2$ denotes the L_2 -norm; α_u and β_v are the representation of user u and item v in the metric vector space, respectively.

Given a triple (u, v, v^-) , $(u, v) \in \mathcal{I}$ indicates the positive sample, while $(u, v^-) \notin \mathcal{I}$ denotes the negative sample. CML adopts the triple loss (Weinberger and Saul 2009; Song et al. 2017) that guarantees the distance between user u and negative item v^- is larger than that between the current user u and positive item v by a given margin of $m > 0$:

$$d(u, v) + m \leq d(u, v^-). \quad (2)$$

Thus, the final objective function could be formulated as follows:

$$\mathcal{L} = \sum_{(u,v) \in \mathcal{I}} \sum_{(u,v^-) \notin \mathcal{I}} w_{u,v} [d(u, v) - d(u, v^-) + m]_+, \quad (3)$$

where $[x]_+ = \max(x, 0)$ denotes the standard hinge loss, $w_{u,v}$ is the loss weight. For fair comparison, we set $w_{u,v}$ be equal to one for all samples in this paper (Zhang et al. 2018; Tay, Tuan, and Hui 2018). To optimize the Equation 3, we can calculate its gradient respectively. if $d(u, v) + m \leq d(u, v^-)$, their gradients are equal to zero; otherwise, as follows: $\frac{\partial \mathcal{L}}{\partial \alpha_u} = 2(\beta_v - \beta_{v^-})$, $\frac{\partial \mathcal{L}}{\partial \beta_v} = 2(\beta_v - \alpha_u)$, $\frac{\partial \mathcal{L}}{\partial \beta_{v^-}} = 2(\alpha_u - \beta_{v^-})$. As shown in Figure 2, according to the framework of stochastic gradient, e.g., $\hat{u} \leftarrow u - \eta \frac{\partial \mathcal{L}}{\partial \alpha_u}$, η denotes the step size, CML could achieve the goal that $d(\hat{u}, \hat{v}) + m \leq d(\hat{u}, \hat{v}^-)$.

However, we find an unexpected problem that the position of positive item may be close with the negative item, leading to $d(\hat{v}, \hat{v}^-) < d(\hat{v}, \hat{u})$. Especially, it becomes more severe when three points u , v , and v^- are all in a straight

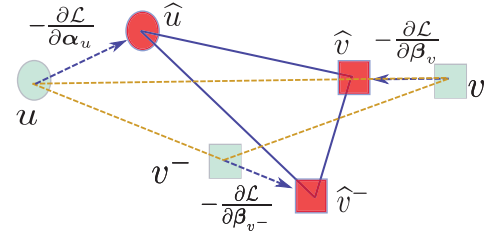


Figure 2: An illustration of the gradient update process of CML.

line, $d(v, v^-) \approx m$. This phenomenon is counterintuitive in recommendation scenario. For example, user u likes item v_1 and dislikes item v_2 , it means that v_1 and v_2 are dissimilar or different. So, in the Euclidean space, v_1 and v_2 should be far apart from each other. Essentially, the triple loss paradigm CML adopted can be seen as the user-centric metric, which only considers the relationship of two edges, i.e., uv and uv^- , ignoring the impact of the third edge of vv^- .

Symmetric Metric Learning with Adaptive Margin

To remit the above problem, we need to push the positive items away from the negative items for the target user. A heuristic yet inflexible strategy is to increase the margin m , or append the additional condition $\max d(v, v^-) = \max \|\beta_v - \beta_{v^-}\|_2^2$.

To design a more effective solution, the following two characteristics shall be taken into consideration: Firstly, it is worth noting that the label (positive/negative) of each item is subject to change for different users. This is significantly different from the traditional classification task (Zadeh, Hosseini, and Sra 2016; Song et al. 2017) where the label is objective and fixed. For example, an item may be positive for user u_1 , while negative for user u_2 . In this case, when its margin m is widened uniformly, positive items will be tightly gathered around the user, which is equivalent to the pull model. As a result, the solution space becomes smaller, and the items and users are indistinguishable. Secondly, the item that should be recommended to a user might fall on the boundary with a high probability due to the wide margin, so it would be predicted as a negative sample, reducing the recall rate of the recommender systems. Besides, if we adopt the additional condition $\max d(v, v^-)$, the training cost is too high. While the goal is only to get a relatively large distance, not the maximum.

As such, we propose a novel method, symmetric metric learning with adaptive margin for recommender systems, to model the complex user-item interactions.

Symmetric Metric Learning As shown in Figure 3, we assume that the condition of user-centric metric has been satisfied, i.e., $d(u, v) + m \leq d(u, v^-)$. From a geometric perspective, we can find that the positive item v is inside of a ball where u is the center and r_u is the radius. While negative item v^- is outside of a larger ball where u is the center and $r_u + m$ is the radius. In the same vector space, users and items share the same metric. Consequently, we can analyze

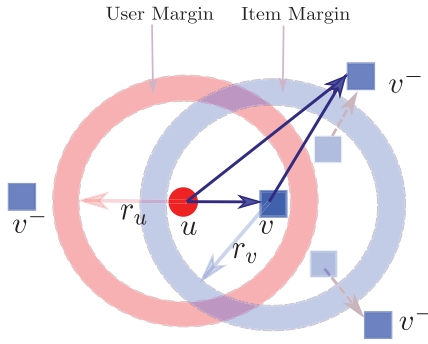


Figure 3: An illustration of symmetric metric learning in 2-dimensional space, $r_u = r_v = d(u, v)$.

their relationship from the item viewpoint: v and v^- are not similar, implying that their distance should be greater than the $d(v, u)$. Therefore, similar to the user-centric model, we also adopts the margin loss to keep positive item v closer to user u than negative item v^- by the margin of n :

$$d(v, u) + n \leq d(v, v^-) \iff d(u, v) + n \leq d(v, v^-). \quad (4)$$

Here $d(v, u) = d(u, v)$ due to the symmetry of Euclidean distance. Analogously, the negative item v^- should be outside of the ball with $r_v + n$ radius and v as the center. It is obvious that the Equation 4 can be seen as the item-centric metric, which is symmetric with the user-centric metric. According to the context, we can make $r_u = r_v = d(u, v)$. Combined with the condition of Equation 2 and Equation 4, we know that negative items for the target user should be outside of the union of the two balls. In summary, our goal is to guarantee \overline{uv} to be the shortest edge of the triangle \triangle_{u, v, v^-} .

Adaptive Margin Based on the fact that different users have different criteria, the margin m should be personalized for each user. Similarly, for the symmetric constraint, the margin n should be also changed with items. In this paper, we use the adaptive margin m_u and n_v to replace the origin margin m and n respectively. Intuitively, we prefer to use larger adaptive margin to reduce the variations. Thus, the adaptive margins could be formulated as:

$$\mathcal{L}_{AM} = - \left(\frac{1}{|\mathcal{U}|} \sum_u m_u + \frac{1}{|\mathcal{I}|} \sum_v n_v \right). \quad (5)$$

Loss function In summary, the final loss function of SML is as follows:

$$\begin{aligned} & \sum_{(u, v) \in \mathcal{I}} \sum_{(u, v^-) \notin \mathcal{I}} \left([d(u, v) - d(u, v^-) + m_u]_+ \right. \\ & \left. + \lambda [d(u, v) - d(v, v^-) + n_v]_+ \right) + \gamma \mathcal{L}_{AM} \quad (6) \\ & \text{s.t.}, \quad m_u \in (0, l], \quad n_v \in (0, l]; \end{aligned}$$

Table 1: Statistics of three datasets.

Datasets	#users	#items	#interactions	sparsity
Instant Video	4,902	1,683	36,486	0.9956
IMDB	1,310	1,635	84,919	0.9604
Yelp	1,631	1,633	78,966	0.9704

where l is bound to prevent the margin from being too large. Furthermore, we apply regularization by normalizing all user embeddings and item embeddings to be constrained within the Euclidean ball, mitigating the issue of 'curse of dimensionality' (Zhang et al. 2018; Tay, Tuan, and Hui 2018; Park et al. 2018). Finally, we apply a constraint of $\|\alpha_u^*\|_2 \leq l$ and $\|\beta_v^*\|_2 \leq l$ for regularization at the end of each mini-batch.

Experiments

In this section, we first describe the experimental settings, including datasets, evaluation protocols, baselines, and implementation details. Subsequently, we conduct extensive experiments to respond the following research questions:

RQ1 How is the effectiveness of SML? Can it provide a competitive performance compared with baselines and the tweaked state-of-the-art metric learning methods?

RQ2 How do the hyper-parameters λ and γ affect the performance and how to choose optimal values?

RQ3 What benefits do the symmetric perspective of items and adaptive margins bring to the top-K recommendation?

RQ4 What are the optimal values of user margins and item margins?

Experimental Settings

Datasets We perform extensive experiments on three publicly accessible datasets: Amazon Instant Video², Yelp Dataset Challenge (Yelp)³, and IMDB⁴, of which the statistics are summarized in Table 1. These datasets have been widely adopted in previous literatures (Chin et al. 2018; Tay, Luu, and Hui 2018). For Amazon Instant Video, the datasets are pre-processed in a 5-core fashion. For Yelp, we use a 10-core setting, providing a comparison on a denser dataset as (Tang et al. 2015; Chen et al. 2016).

Evaluation protocols To evaluate the recommendation performance, we randomly divide the training set and testing set following the ratio 9: 1 for each dataset. Moreover, 10% records in the training set are selected as the validation set randomly for hyper-parameters selection. *Note*: for each user/item, at least one interaction is included in the training set. Due to the enormous time cost of ranking all items for each user, we sample 499 items which have no interaction with the target user, following previous work (Park et al. 2018; Tay, Tuan, and Hui 2018; He et al. 2018). To evaluate the ranking accuracy and quality, we adopt three widely-used metrics (Zhang et al. 2018;

²<http://jmcauley.ucsd.edu/data/amazon/>

³<https://www.yelp.com/dataset/challenge>

⁴<http://ir.hit.edu.cn/~dytang/paper/ac12015/dataset.7z>

Table 2: Distance measurement and loss function comparison with baselines. f^* indicates an aggregate function, such as a neural network. u^{nbr} and v^{nbr} denote the neighbors of user and item, respectively. $y_{u,v}$ is the binary label, and σ denotes the activation function.

Methods	Distance	Loss
BPR(Rendle et al. 2009)	$d(u, v) = \alpha_u \beta_v^T$	$-\sum_{(u,v) \in \mathcal{I}} \sum_{(u,v^-) \notin \mathcal{I}} \frac{1}{1 + \exp(-d(u,v) + d(u,v^-))}$
MLP/NCF(He et al. 2017)	$d(u, v) = \sigma(f^*(\alpha_u, \beta_v))$	$-\sum_{(u,v)} y_{u,v} \log d(u, v) + (1 - y_{u,v}) \log(1 - d(u, v))$
FML(Zhang et al. 2018)	$d(u, v) = \ \alpha_u - \beta_v\ _2^2$	$\sum_{(u,v)} w_{u,v} \ y_{u,v} - d(u, v)\ ^2$
TransCF(Park et al. 2018)	$d(u, v) = \ \alpha_u + \mathbf{r}_{u^{nbr}, v^{nbr}} - \beta_v\ _2^2$	$\sum_{(u,v) \in \mathcal{I}} \sum_{(u,v^-) \notin \mathcal{I}} [d(u, v) - d(u, v^-) + m]_+ + \lambda [d(u, v) - d(v, v^-) + n_v]_+ + \gamma \mathcal{L}_{AM}$
LRML(Tay, Tuan, and Hui 2018)	$d(u, v) = \ \alpha_u + \mathbf{r}_{u,v} - \beta_v\ _2^2$	
CML(Hsieh et al. 2017)	$d(u, v) = \ \alpha_u - \beta_v\ _2^2$	
SML	$d(u, v) = \ \alpha_u - \beta_v\ _2^2$	

Wang et al. 2019): Precision (P@K), Hit Ratio (HR@K), and Normalized Discounted Cumulative Gain (NDCG@K).

Baselines We evaluate SML comparing with the following baselines. Table 2 summarizes the similarities and differences of the distance measurements and loss functions used in the all methods.

- **BPR** (Rendle et al. 2009) is a classical pair-wise learning-to-rank method, whose optimization criterion aims to maximize the differences between negative and positive samples.
- **MLP** (He et al. 2017) is a module of neural collaborative filtering method, which only uses the multi-layered perceptron (MLP) to learn the feature representations.
- **NCF** (He et al. 2017) is a competitive neural collaborative filtering method, which combines multi-layered perceptron (MLP) with generalized matrix factorization (GML).
- **FML** (Zhang et al. 2018) is a metric learning method, which first converts preferences into distances and then replaces the inner product with Euclidean distance.
- **TransCF** (Park et al. 2018) is a collaborative translational metric learning method, which constructs user-item specific translation vectors by employing the neighborhood information. Each user are translated toward items using above translation vectors.
- **LRML** (Tay, Tuan, and Hui 2018) is a latent relational metric learning method inspired by TransE (Bordes et al. 2013), which employs an augmented memory module to induce a latent relation for each user-item interaction.
- **CML** (Hsieh et al. 2017) is a collaborative metric learning method, which assumes that users and items could be placed in a low dimensional space, and the explicit closeness between users and items could be measured by Euclidean distance.

Note that CML is our stable baseline. LRML and TransCF are the latest methods based on the CML paradigm.

Implementation Details We implement our model in Tensorflow. The implementation of the comparison methods are from the public codes that the authors provided in their papers or open source project (BPR/FML/CML⁵, MLP/NCF⁶, TransCF⁷, and LRML⁸). We optimize the proposed SML with the Adam optimizer and tune the learning rate in $\{0.10, 0.05, 0.01\}$ for different datasets. The embedding size is fixed to 100. The batch size is 512. In terms of variable, all weight variables are randomly initialized with uniform distributions of $[-0.01, 0.01]$, and all latent vectors (such as α_u and β_v) are randomly initialized with normal distributions (mean: 0.1, variance: 0.03). Without specification, we show the results of all datasets with $l=1.0$. The code is released at Github⁹.

Performance Comparison (RQ1)

Table 3 shows the overall performances of our proposed method as well as the baseline methods. We can make the following observations.

First, we can observe that CML outperforms BPR, MLP, and NCF. This is consistent with the previous work (Tay, Tuan, and Hui 2018; Zhang et al. 2018), indicating that distance metric is useful and overcomes the inherent limitation of the inner product.

Second, our proposed method significantly outperforms all the baseline methods on three datasets in terms of different metrics. Particularly, SML performs competitively to CML, which is the most relevant method with us. These results explain that the item/user-centric and adaptive margin are helpful and improve the expressive ability of the model.

Third, compared with the relational metric learning based TransCF and LRML, which are also based on the user-centric metric, we can see that SML still achieves a significant improvement. This conclusion shows that the improve-

⁵<https://github.com/cheungdaven/DeepRec>

⁶https://github.com/hexiangnan/neural_collaborative_filtering

⁷<https://github.com/pcy1302/TransCF>

⁸https://github.com/vanztytay/WWW2018_LRML

⁹<https://github.com/MingmingLie/SML>

Table 3: Performance comparisons on item ranking task in terms of P@K, HR@K, and NDCG@K on three datasets. The best performance is denoted in boldface.

K=5	Instant Video			IMDB			Yelp		
	P@5	HR@5	NDCG@5	P@5	HR@5	NDCG@5	P@5	HR@5	NDCG@5
BPR	0.1999	0.6918	0.7196	0.6202	0.4331	0.8115	0.5669	0.5316	0.8005
MLP	0.1887	0.6367	0.6924	0.4371	0.3132	0.5867	0.4935	0.4638	0.6708
NCF	0.2045	0.6903	0.7322	0.6208	0.4449	0.8214	0.5778	0.5431	0.8004
FML	0.1994	0.6728	0.6713	0.6032	0.4323	0.8029	0.5847	0.5495	0.8094
TransCF	0.2113	0.7059	0.7497	0.6387	0.4577	0.8390	0.6038	0.5675	0.8215
LRML	0.2047	0.6908	0.7364	0.6424	0.4604	0.8440	0.5990	0.5630	0.8196
CML	0.2048	0.6911	0.7358	0.6409	0.4593	0.8408	0.6029	0.5667	0.8237
SML	0.2166	0.7309	0.7698	0.6735	0.4827	0.8755	0.6277	0.5900	0.8518

K=10	Instant Video			IMDB			Yelp		
	P@10	HR@10	NDCG@10	P@10	HR@10	NDCG@10	P@10	HR@10	NDCG@10
BPR	0.1011	0.7198	0.7396	0.3529	0.5561	0.8111	0.3322	0.6433	0.7793
MLP	0.0950	0.6426	0.6931	0.2850	0.4087	0.5758	0.3010	0.5678	0.6828
NCF	0.1065	0.7188	0.7405	0.3858	0.5529	0.8214	0.3328	0.6255	0.7752
FML	0.1071	0.7229	0.6874	0.3758	0.5386	0.8209	0.3366	0.6327	0.7872
TransCF	0.1097	0.7406	0.7570	0.4096	0.5871	0.8199	0.3563	0.6697	0.8083
LRML	0.1057	0.7132	0.7420	0.4119	0.5903	0.8262	0.3531	0.6638	0.8001
CML	0.1055	0.7118	0.7408	0.4100	0.5876	0.8205	0.3539	0.6653	0.8133
SML	0.1118	0.7546	0.7759	0.4312	0.6180	0.8553	0.3701	0.6958	0.8324

ment of the symmetric metric is remarkable and could provide important potentials for recommendation.

Hyper-parameter Studies (RQ2)

SML introduces two additional hyper-parameters λ and γ to control the learning of symmetric view on item (item-centric) and the strength of adaptive margin respectively. Here we show how the two hyper-parameters impact the performance and also shed lights on how to set them. We only show the results on Amazon Instant Video and IMDB due to the limitation of space.

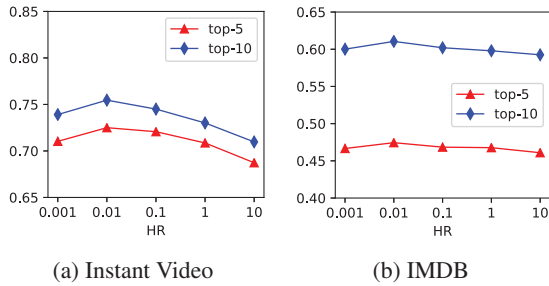


Figure 4: Performance of SML with respect to different values of λ on two datasets.

First, we fix γ to 10 for Instant Video and 0.1 for IMDB, and vary λ . As shown in from Figure 4, the optimal value is around 0.01 for two datasets. Besides, we also observe that increasing λ leads to the performance gradual declines on both datasets. Thus, too larger value of λ will destroy

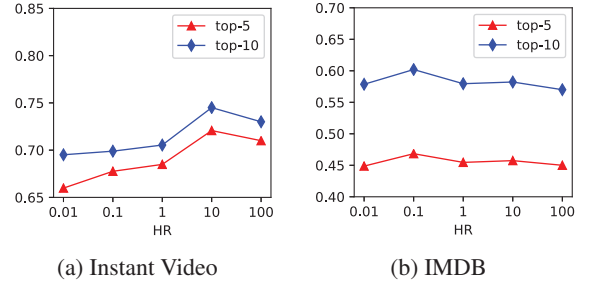


Figure 5: Performance of SML with respect to different values of γ on two datasets.

the learning process of user’s metric. As such, we suggest setting λ to 0.01.

Second, we fix λ to the default value of 0.01 and vary γ . The results in term of HR@K are shown in Figure 5, we find that the optimal values are around 10 and 0.1 for Instant Video and IMDB respectively. When γ is too smaller (e.g., less than 0.01) on Instant Video, SML behaves minor improvements, which shows the adaptive margin module has not been given an important attention. Moreover, when γ is too large (e.g., larger than 100), the performance also drops dramatically. These results are also consistent with our previous discussion that margin cannot be increase infinitely.

Ablation Studies (RQ3)

We evaluate the contribution of each of SML’s components, the symmetric metric learning on the perspective of item (denoted as s), the user specific margin (denoted as u), and

Table 4: Performances on three datasets by different components.

K=5	Instant Video			IMDB			Yelp		
	P@5	HR@5	NDCG@5	P@5	HR@5	NDCG@5	P@5	HR@5	NDCG@5
base	0.2048	0.6911	0.7258	0.6409	0.4593	0.8408	0.6029	0.5667	0.8237
base+u	0.2130	0.7256	0.7558	0.6521	0.4617	0.8542	0.6107	0.5709	0.8354
base+s	0.2141	0.7326	0.7618	0.6279	0.4500	0.8226	0.6073	0.5680	0.8284
base+s+p	0.2150	0.7256	0.7605	0.6355	0.4554	0.8301	0.6036	0.5673	0.8278
base+s+u	0.2169	0.7347	0.7627	0.6430	0.4608	0.8400	0.6117	0.5749	0.8321
base+s+u+p(all)	0.2166	0.7309	0.7698	0.6735	0.4827	0.8755	0.6277	0.5900	0.8518

K=10	Instant Video			IMDB			Yelp		
	P@10	HR@10	NDCG@10	P@10	HR@10	NDCG@10	P@10	HR@10	NDCG@10
base	0.1055	0.7118	0.7308	0.4100	0.5876	0.8205	0.3539	0.6653	0.8133
base+u	0.1112	0.7506	0.7724	0.4209	0.6075	0.8342	0.3603	0.6761	0.8178
base+s	0.1126	0.7600	0.7708	0.4025	0.5768	0.7998	0.3570	0.6711	0.8153
base+s+p	0.1121	0.7568	0.7686	0.4063	0.5823	0.8061	0.3575	0.6721	0.8117
base+s+u	0.1128	0.7614	0.7713	0.4158	0.5960	0.8229	0.3598	0.6761	0.8164
base+s+u+p(all)	0.1118	0.7546	0.7759	0.4312	0.6180	0.8553	0.3701	0.6958	0.8324

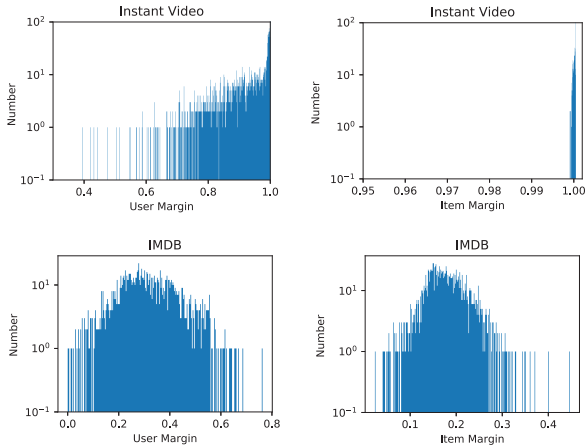


Figure 6: Visualization of the margins of users and items on Instant Video and IMDB datasets respectively.

the item specific (denoted as p), to the overall performance while keeping all hyperparameters at their optimal settings. The results on three datasets are shown in Table 4. The base model is similar to CML, denoted as 'base'; base+ x denotes base model with the components $x \in \{s, u, p\}$ used, e.g., base+s+u denotes both the item-centric and user specific margin are used. We can see that the base model performs the worst for yelp, whereas others improve the performance significantly. Compared with base model, base+u makes a great improvement on three datasets, it indicates that the adaptive user margins are beneficial for the performance of recommender systems. What's more, there are several interesting results. For Instant Video, item margins make little rule for performance, and the following visualization subsection also illustrates this problem. For IMDB, user margins and item margins are useful and bring importantly improvement in performance, which also indicates that the biases of users and items are different in this data distribution. Thus, we suggest that the most effective components should get through ablation testing.

Margin Visualization (RQ4)

To provide a more intuitive understanding of the adaptive margins of users and items, we visualize them on two datasets (Instant Video and IMDB) shown in Figure 6. It is obvious that given a fixed margin for all users is inapposite.

For Instant Video, the user margins are in the range of 0.6 to 1.0, and item margins are all integrated into the scope of 1.0. These results show that there are huge biases on different users on Instant Video, while little bias on items. Similarly, the user margins and items margins are in the range of 0.1 to 0.5 and 0.1 to 0.3 on IMDB, respectively. Impressively, the adaptive margin could eliminate the impact of the user/item bias to benefit the performance of recommender systems.

Conclusion

In this paper, we focus on the application of metric learning based methods for implicit feedback in recommender systems. First, we analyze the problem of exiting paradigm, which only uses the user-centric metric with a fixed margin, while ignores the impact of item-centric metric and user/item bias. To address these issues, we propose a novel paradigm named symmetric metric learning with adaptive margin (SML). SML measures the trilateral relationship of given triple (u, v, v^-) from the perspective of both user-centric and item-centric to improve performance. What's more, the adaptive margin strategy helps the model to select the appropriate margins for different users and items. Extensive experiments demonstrate that SML produces competitive performances compared with state-of-the-art methods. We believe that our symmetric paradigm with adaptive margin would shed a new angle for other research areas.

References

Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; and Yakhnenko, O. 2013. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*, 2787–2795.

Chen, S.; Moore, J. L.; Turnbull, D.; and Joachims, T. 2012. Playlist prediction via metric embedding. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, 714–722. ACM.

Chen, H.; Sun, M.; Tu, C.; Lin, Y.; and Liu, Z. 2016. Neural sentiment classification with user and product attention. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, 1650–1659.

Chin, J. Y.; Zhao, K.; Joty, S.; and Cong, G. 2018. ANR: aspect-based neural recommender. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018*, 147–156.

- Do, T.-T.; Tran, T.; Reid, I.; Kumar, V.; Hoang, T.; and Carneiro, G. 2019. A theoretically sound upper bound on the triplet loss for improving the efficiency of deep distance metric learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 10404–10413.
- Feng, S.; Li, X.; Zeng, Y.; Cong, G.; Chee, Y. M.; and Yuan, Q. 2015. Personalized ranking metric embedding for next new poi recommendation. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.
- He, X.; Liao, L.; Zhang, H.; Nie, L.; Hu, X.; and Chua, T.-S. 2017. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*, 173–182. International World Wide Web Conferences Steering Committee.
- He, X.; He, Z.; Song, J.; Liu, Z.; Jiang, Y.-G.; and Chua, T.-S. 2018. Nais: Neural attentive item similarity model for recommendation. *IEEE Transactions on Knowledge and Data Engineering* 30(12):2354–2366.
- Hsieh, C.-K.; Yang, L.; Cui, Y.; Lin, T.-Y.; Belongie, S.; and Estrin, D. 2017. Collaborative metric learning. In *Proceedings of the 26th International Conference on World Wide Web*, 193–201. International World Wide Web Conferences Steering Committee.
- Ji, G.; He, S.; Xu, L.; Liu, K.; and Zhao, J. 2015. Knowledge graph embedding via dynamic mapping matrix. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, 687–696.
- Khoshneshin, M., and Street, W. N. 2010. Collaborative filtering via euclidean embedding. In *Proceedings of the fourth ACM conference on Recommender systems*, 87–94. ACM.
- Kulis, B., et al. 2013. Metric learning: A survey. *Foundations and Trends® in Machine Learning* 5(4):287–364.
- Liu, X.; Vijaya Kumar, B.; You, J.; and Jia, P. 2017. Adaptive deep metric learning for identity-aware facial expression recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 20–29.
- Park, C.; Kim, D.; Xie, X.; and Yu, H. 2018. Collaborative translational metric learning. In *2018 IEEE International Conference on Data Mining (ICDM)*, 367–376. IEEE.
- Ram, P., and Gray, A. G. 2012. Maximum inner-product search using cone trees. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, 931–939. ACM.
- Rendle, S.; Freudenthaler, C.; Gantner, Z.; and Schmidt-Thieme, L. 2009. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*, 452–461. AUAI Press.
- Song, K.; Nie, F.; Han, J.; and Li, X. 2017. Parameter free large margin nearest neighbor for distance metric learning. In *Thirty-First AAAI Conference on Artificial Intelligence*.
- Tang, D.; Qin, B.; Liu, T.; and Yang, Y. 2015. User modeling with neural network for review rating prediction. In *IJCAI*, 1340–1346.
- Tay, Y.; Luu, A. T.; and Hui, S. C. 2018. Multi-pointer co-attention networks for recommendation. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2309–2318. ACM.
- Tay, Y.; Tuan, L. A.; and Hui, S. C. 2018. Latent relational metric learning via memory-based attention for collaborative ranking. In *Proceedings of the 27th International Conference on World Wide Web*.
- Wang, Z.; Zhang, J.; Feng, J.; and Chen, Z. 2014. Knowledge graph embedding by translating on hyperplanes. In *Twenty-Eighth AAAI conference on artificial intelligence*.
- Wang, J.; Zhou, F.; Wen, S.; Liu, X.; and Lin, Y. 2017. Deep metric learning with angular loss. In *Proceedings of the IEEE International Conference on Computer Vision*, 2593–2601.
- Wang, H.; Zhang, F.; Zhao, M.; Li, W.; Xie, X.; and Guo, M. 2019. Multi-task feature learning for knowledge graph enhanced recommendation. *arXiv preprint arXiv:1901.08907*.
- Weinberger, K. Q., and Saul, L. K. 2009. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research* 10:207–244.
- Zadeh, P.; Hosseini, R.; and Sra, S. 2016. Geometric mean metric learning. In *International Conference on Machine Learning*, 2464–2471.
- Zhang, S.; Yao, L.; Huang, C.; Xu, X.; and Zhu, L. 2018. Position and distance: Recommendation beyond matrix factorization. *CoRR* abs/1802.04606.
- Zhang, J.; Kalantidis, Y.; Rohrbach, M.; Paluri, M.; Elgammal, A.; and Elhoseiny, M. 2019. Large-scale visual relationship understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 9185–9194.
- Zhou, X.; Liu, D.; Lian, J.; and Xie, X. 2019. Collaborative metric learning with memory network for multi-relational recommender systems. *arXiv preprint arXiv:1906.09882*.