Collaborative Social Metric Learning in Trust Network for Recommender Systems

Taehan Kim, Korea University, South Korea
Wonzoo Chung, Korea University, South Korea*

ABSTRACT

In this study, a novel top-K ranking recommendation method called collaborative social metric learning (CSML) is proposed, which implements a trust network that provides both user-item and user-user interactions in simple structure. Most existing recommender systems adopting trust networks focus on item ratings, but this does not always guarantee optimal top-K ranking prediction. Conventional direct ranking systems in trust networks are based on sub-optimal correlation approaches that do not consider item-item relations. The proposed CSML algorithm utilizes the metric learning method to directly predict the top-K items in a trust network. A new triplet loss is further proposed, called socio-centric loss, which represents user-user interactions to fully exploit the information contained in a trust network, as an addition to the two commonly used triplet losses in metric learning for recommender systems, which consider user-item and item-item relations. Experimental results demonstrate that the proposed CSML outperformed existing recommender systems for real-world trust network data.

KEYWORDS
Homophily, Item Recommendation, Metric Learning, Recommender Systems, Social Recommendation, Trust Network, User-Item Relation, User-User Relation

INTRODUCTION

In the last decades, the internet has explosively expanded. In a vast sea of information, technologies of big data engineering, machine learning, and deep learning, which solve problems in several areas such as fake news detection, healthcare, computer vision, and recommendation, prosper but following computational load remains a burden (Hao et al., 2022; Li et al., 2022; Tembhurne et al., 2022; Wang et al., 2021; Yu & Reiff-Marganiec, 2022). Meanwhile, internet surfers find it difficult to choose what they want from a large amount of varied information. Therefore, offering content lists of what they might want becomes more important.

Recommender systems are widely used by various web sites, such as Amazon, YouTube, and Netflix to help users find contents they might wish to interact with. Top-K personalized recommendation is conventionally performed by recommender systems to satisfy each user’s individual preference for various items (Li et al., 2022; Wang et al., 2021; Wu et al., 2016; Xue et al., 2019).

DOI: 10.4018/IJSWIS.316535  
*Corresponding Author

This article published as an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0/) which permits unrestricted use, distribution, and production in any medium, provided the author of the original work and original publication source are properly credited.
Social recommendations use additional information on social relations encoded as a network to improve recommendation performance (Gao et al., 2012; Tang et al., 2013a). Trust networks are social networks involving both a user-item network and user-user relations (Ma et al., 2009; Tang et al., 2013a). Because trust networks have the property that people with similar tastes naturally tend to gather or associate, called homophily (Gao et al., 2012; McPherson et al., 2001; Tang et al., 2013a), enhancement of item recommendation performance in trust networks has been verified with real-world data, such as Ciao, Epinions, Yelp, and Gowalla (Ardissono & Mauro, 2020; Fan et al., 2019; Tang et al., 2012a, 2012b, 2013a; Wang et al., 2020b).

Existing recommendation methods in trust networks for top-K recommendations are based on two approaches: rating prediction and direct ranking (Aggarwal, 2016, pp. 345–361). Rating prediction approaches offer predicted ratings for unrated items of a user and recommend top-K items to users based on their ratings (Ardissono & Mauro, 2020; Chanyoung et al., 2016; Hao et al., 2020; Wang et al., 2020b; Wang et al., 2021; Zhang et al., 2020). However, the best K-rated items do not always match with the best top-K recommendation; direct ranking methods are therefore preferred (Cremonesi et al., 2010; Yang et al., 2012). Moreover, rating prediction is generally based on explicit data representing specific and explicit preferences of users (e.g., from 1- to 5-star ratings on movies) and cannot be applied to implicit data, which includes a wider range of data, such as clicks, purchases, and views. In the process of learning user-user interactions, several models utilize a user-user cosine similarity matrix which causes a computational burden (Ardissono & Mauro, 2020; Tang et al., 2013a; Zhang et al., 2020).

Existing direct ranking approaches in the trust network (Krohn-Grimberge et al., 2012; Rafailidis & Crestani, 2016, 2017; Zhao et al., 2019) are based on finding the probability that a certain user prefers a certain item from the correlation (computed by the dot product) between a user and an item. However, these dot-product-based methods have fundamental limitations: (1) owing to a lack of similarity propagation process (Hsieh et al., 2017), the similar items of a pivoted item should also be similar to each other, and (2) failing to capture the similarities among users (Zhang et al., 2018). Furthermore, all these algorithms ignore item-item relations in the network, which can improve the ranking performance.

In this paper, we propose a direct top-K ranking recommender system in trust networks based on metric learning, called Collaborative Social Metric Learning (CSML). As in a metric learning recommendation algorithm, users and items are represented as $d$-dimensional vectors and related pairs are gathered closely and unrelated pairs are repelled promptly using triplet loss. CSML considers relations between users, or user-user relations, and the related user-user pair is defined as two users who share a “trustor-trustee” relation determined by the trust network, whereas unrelated user-user pairs refer the pairs of users without this relation. Consequently, CSML utilizes the sum of the following three triplet losses to achieve a direct top-K ranking recommendation: first, user-centric loss (Hsieh et al., 2017; Li et al., 2020) that learns user-item relations; second, item-centric loss (Li et al., 2020) that learns item-item relations, and, finally, sociocentric loss, proposed in this paper, which learns a user-user relation. After metric learning with respect to the abovementioned loss, the K-items having the least distance to a user are recommended to the user.

The contribution of this study is to provide a metric learning framework for direct ranking K-recommendations in trust networks. The proposed CSML algorithm inherits the advantages of the metric learning recommendation system, that is, the ability to use both explicit and implicit data, and a similarity propagation process. Furthermore, by exploiting the user-user interaction in a trust network, the CSML produces state-of-the-art recommendation performance.

The remainder of this study is organized as follows. Below, related works on recommender systems in trust networks and metric learning are briefly reviewed. In the Preliminaries section, the metric learning method in recommendation is described with mathematical notations. In the Proposed Method section that follows, the CSML algorithm is presented, and the loss function is given in detail. In the Experiments section, simulation results of the proposed algorithm on real-world data
sets are presented. In the Complexity Analysis section, it is verified that as CSMLs do not utilize a cosine similarity matrix, this model reduces the computation load. The final section provides some concluding remarks.

RELATED WORKS

Recommendation With Trust Network

User networks in a trust network help to improve the performance of the top-K item recommendation. Methods to apply the knowledge of user networks are a key focus of existing works. A top-K recommendation with social information (TRecSo) was proposed using multiple embedded vectors for each user as trustor, trustee, and neighbor, and for each item (as items in which users and their neighbors are currently interested). This then aggregates all the vectors by a function to predict the rating (Chanyoung et al., 2016). TRecSo also learns user-user interaction with an indegree and outdegree of each user in a user network. Dimension-wise attention social recommendation (DASR) utilizes user-user networks as component-wise weights to predict ratings (Wang et al., 2020a). Diversity balancing for two-stage collaborative filtering (DBTS) utilizes both direct and indirect connected users and their trust degrees, with cosine similarities of users to obtain weights to predict ratings. It additionally proposes a reranking method for diversity (Zhang et al., 2020). LOCABAL+ proposes a weight matrix called a user correlation preference matrix to capture user-user interaction and utilizes each user’s reputations as weights in losses to learn both user-item and user-user interactions (Ardissono & Mauro, 2020). Learning to rank with trust and distrust (LTRW), Rafailidis & Crestani (2017) additionally considered the positive and negative items of neighbors and nonneighbors to predict the rankings of items. Trust-enhanced collaborative filtering (TECF) (Wang et al., 2020b) learns user networks in interest and identity representation learning (IIRL) to calculate user preferences to recommend points-of-interests (POIs) for a given user with geographical and temporal influence (Xu et al., 2018).

Metric Learning in Recommendation

Existing methods of metric learning in recommendation systems mainly learn user-item interaction and item-item interactions. Collaborative metric learning (CML) learns user-item interaction and that similarity propagation holds by triangle inequality (Hsieh et al., 2017). TransCF introduces translation vectors to gather information on users and items within a given neighborhood (Chanyoung et al., 2018). However, CML and TransCF do not consider item-item interaction. Symmetric metric learning with adaptive margin (SML) additionally considered item-item relations and proposed margins as personalized learnable parameters (Li et al., 2020).

PRELIMINARIES

Notations and Definitions

In this section, we introduce some basic notation and definitions to describe metric learning-based algorithms in recommendation systems. Let $\mathcal{U}$ be a set of users and $\mathcal{V}$ be a set of items.

Let $I := \{(u, v) \mid u \in \mathcal{U}, v \in \mathcal{V}\} \subseteq \mathcal{U} \times \mathcal{V}$ be set of all user-item pairs that have interaction and $I^- := \mathcal{U} \times \mathcal{V} \setminus I$. These are the only available information in most conventional recommendation systems.

A trust network is characterized by additional information, typically including user-user relations. Let $T$ be a matrix representing the user-user relation, called user-trust matrix, defined by $T_{ij} = 1$ if user $i$ is the trustor and the user $j$ is the trustee, otherwise $T_{ij} = 0$ (note that $T$ may not be symmetric). The trustor-trustee relation is determined by the data, for example, listed or followed.
users can be trustees. Most existing recommendation systems exploit these user-user relations to improve recommendation performance.

**Metric Learning in Recommendation**

In metric learning-based recommendation systems, every element \( e \in \mathcal{U} \cup \mathcal{V} \) is embedded in a \( d \)-dimensional metric space with squared Euclidean distance (Chanyoung et al., 2018; Hsieh et al., 2017; Li et al., 2020). Let \( z_a \) and \( z_b \) denote the embedded vectors of \( a \) and \( b \in \mathcal{U} \cup \mathcal{V} \), respectively. The distance between \( a \) and \( b \) is given as:

\[
d(a, b) = \| z_a - z_b \|_2^2
\]

(1)

The aim of metric learning-based recommendation systems is to place embedded user vectors and item vectors, such that: i) for a given user, the related items are located closer than the unrelated items to the user, and, at the same time, ii) for a given item, related users are located closer than their unrelated items to the item.

Rigorously speaking, given user \( u \):

\[
d(u, j) \geq d(u, i) + m
\]

(2)

for all \((u, i) \in I, (u, j) \in I^-\) with a nonnegative margin \( m \) and given item \( i \):

\[
d(i, j) \geq d(i, u) + n
\]

(3)

for all \((u, i) \in I, (u, j) \in I^-\) with a nonnegative margin \( n \). To implement such a requirement, the following triplet losses are considered.

**User-Item Interaction Learning**

Given a triple \((u, i, j)\), where \((u, i) \in I, (u, j) \in I^-\), the triplet loss \(\max(m + d(u, i) - d(u, j), 0)\), called user-centric loss, is introduced to achieve Equation 2 via a stochastic gradient update of embedded vectors (Hsieh et al., 2017; Li et al., 2020). Owing to the triangle inequality, similar items tend to gather together and, consequently, the similarity propagation process holds (Hsieh et al., 2017).

**Item-Item Interaction Learning**

In order to achieve Equation 3, item-centric loss, defined as loss \(\max(n + d(i, u) - d(i, j), 0)\) for a given triple \((u, i, j)\), is used (Li et al., 2020).

**Adaptive Margins**

It is known that personalized adaptive margins for each user in user-centric loss and each item in item-centric loss achieve better performance in item recommendation (Li et al., 2020). To optimize the individual margins \(m_u, n_i\), the following loss function is used with the constraint that \(0 < m_u, n_i \leq 1\):
\[ \mathcal{L}_{AM} = - \left( \sum_{u \in \mathcal{U}} \frac{m_u}{|\mathcal{U}|} + \sum_{i \in \mathcal{I}} \frac{n_i}{|\mathcal{I}|} \right) \] (4)

**PROPOSED METHOD**

**User-User Interaction Learning**

Because the trustor and trustee in a trust network have similar tastes or interests, called *homophily*, a trustor vector and a trustee vector should be located close to each other in a metric space from the perspective of metric learning (Aggarwal, 2016, pp. 345–361; Eunjoon et al., 2011; Gao et al., 2012; Tang et al., 2012a, 2012b, 2013a). In other words, given a user, it is desirable to place a trustee closer to the user than the so-called *unconnected* users (neither trustor nor trustee). For a given user \( u \), the following equation holds:

\[ d(u, w) \geq d(u, v) + h \] (5)

for all users \( v, w \), such that \( T_{u, w} = T_{u, v} = 0 \) and \( T_{u, v} = 1 \) with a positive margin \( h \).

**Socio Centric Loss**

To achieve Equation 6, we propose a triplet loss function, called *sociocentric-loss* and defined as follows. Given a triple \((u, v, w)\), where \( u, v, w \in \mathcal{U} \) with \( T_{u, v} = 1 \) and \( T_{u, w} = T_{v, w} = 0 \), which is defined as follows:

\[ \mathcal{L}_s = \max(h + d(u, v) - d(u, w), 0) \] (6)

Because this *sociocentric loss* should be minimized simultaneously with conventional *user-centric* and *item-centric* metric learning losses, the final loss is given as the sum of all three losses.

**Adaptive Margin for Sociocentric Loss**

The margin for *sociocentric loss* can be personalized and adaptive, that is, for each user \( u \), an individual margin \( h_u (0 < h_u \leq 1) \) can be considered. The adaptive margin loss term, in this case, is modified as:

\[ \mathcal{L}_{AM} = - \left( \sum_{u \in \mathcal{U}} \frac{m_u + h_u}{|\mathcal{U}|} + \sum_{i \in \mathcal{I}} \frac{n_i}{|\mathcal{I}|} \right) \] (7)

where \( 0 < m_u, n_i, h_u \leq 1 \).

**Loss Function for CSML**

Finally, the loss function for CSML is given as the weighted sum of *user-centric*, *item-centric*, and *sociocentric losses* with modified adaptive margin loss:
\[
L_{\text{total}} = \sum_{(u,i,j) \in D} \max \left( m + d(u,i) - d(u,j), 0 \right) \\
+ \lambda \sum_{(u,i,j) \in D} \max \left( n + d(i,u) - d(i,j), 0 \right) \\
+ \rho \sum_{(u,v) \in S} \max \left( h + d(u,v) - d(u,w), 0 \right) \\
+ \gamma L_{\text{mAM}} \\
\]

s.t., \( 0 < m, n, h, \lambda, \gamma, \rho \leq 1 \) \hspace{1cm} (8)

where \( D \) and \( S \) are valid triple sets obtained from the training data and \( \lambda, \gamma, \rho \) are the hyperparameters set by grid search. At the end of each minibatch, all the embedded vectors are normalized simultaneously to prevent overfitting:

\[
z_a \rightarrow \frac{z_a}{\max \left( \| z_a \|_1 \right)} \text{ for all } a \in \mathcal{U} \cup \mathcal{V} \hspace{1cm} (9)
\]

**Gradient Updates for CSML**

Given item \( v \in \mathcal{V} \), the update of \( z_v \) with respect to \( L_{\text{total}} \) is exactly the same as the conventional metric learning recommendation system (Li et al., 2020). For a given user \( u \in \mathcal{U} \), to update \( z_u \), the following three updates should be counted, in addition to the conventional metric updates for users:

1. \( u \) is a trustor of \( v_1, \cdots, v_{K_u} \) and has unconnected users \( w_1, \cdots, w_{M_u} \):

\[
d_{u1} = 2\rho \sum_{i=1}^{N_u} \sum_{j=1}^{M_u} \left( z_{w_j} - z_{v_i} \right) \hspace{1cm} (10)
\]

2. \( u \) is a trustee of \( a_1, \cdots, a_{K_u} \):

\[
d_{u2} = 2\rho \sum_{i=1}^{K_u} \left( z_u - z_{a_i} \right) \hspace{1cm} (11)
\]

3. \( u \) is an unconnected user of \( b_1, \cdots, b_{K_u} \):

\[
d_{u3} = 2\rho \sum_{i=1}^{K_u} \left( z_{b_i} - z_u \right) \hspace{1cm} (12)
\]

Consequently, the final update equation for \( z_u \) in terms of the user-item and item-item relations is given as:
\[ z_u^{new} = z_u^{old} \]

\[-\eta \left(d_{u1} + d_{u2} + d_{u3}\right)\]

\[+ 2 \sum_{(u,i,j) \in D} (z_j - z_i)\]

\[+ 2\lambda \sum_{(u,i,j) \in D} (z_u - z_i)\]  \hspace{1cm} (13)

where \( \eta \) is the learning rate. Furthermore, the update for the personalized margin \( h_u \) is calculated as:

\[ h_u^{new} = \max \left( h_u^{old} + \eta \frac{1}{|U|}, 1 \right) \]  \hspace{1cm} (14)

The gradient update process of CSML is illustrated in Figure 1. Denote user(trustor) as \( u \), trustee as \( v \), unconnected user as \( w \), positive item as \( i \), and negative item as \( j \). By using one-hot vector as a lookup index, users and items are embedded as vectors with random initialization. With three losses, which are user-centric loss, item-centric loss, and sociocentric loss, each embedding in training step \( t \) is updated as displayed in training step \( t+1 \).

**EXPERIMENTS**

**Data Set**

The data sets we experimented with included Ciao (Tang et al., 2012a, 2012b) and Yelp-Hotel (data that are last updated on February 21, 2020) (Ardissono & Mauro, 2020). These data sets are widely used in recommender systems with a trust network after appropriate pre-processing (Tang et al., 2012a, 2012b, 2013a; Wang et al., 2020b).

Figure 1. Visualized Gradient Update Process of CSML; from One-hot Vector to Gradient Update
The Ciao data set were preprocessed to meet the condition that each user has at least five interaction items, and each item has at least five interacting users. We filter out trustees who do not have common items for a given user. After preprocessing, the Ciao consists of 5,858 users, 10,685 items, 141,900 user-item interactions, 5,300 trustors, 5,763 trustees, and 53,136 user-user interactions. The sparsity of user-item interactions is 0.0027 and user-user interactions is 0.0015.

The Yelp-Hotel data are preprocessed, such that each user had at least 10 interaction items as considered in Ardissono & Mauro (2020), and trustees who do not have common items for a given user are filtered out. After preprocessing, the Yelp-Hotel data consist of 1,100 users, 1,123 items, 18,875 user-item interactions, 934 trustors, 934 trustees, and 15,734 user-user interactions. The sparsity of user-item interactions is 0.0153 and user-user interactions is 0.0180.

The trust network was then formed with the preprocessed data and, consequently, some users did not have trustees. For each user, we divided the interacting items into training, validation, and testing sets with a ratio of 64:16:20. Owing to the size of the data, a massive computational load is often required to list all the negative items for each user $u$. To reduce the computational burden, we applied negative sampling (Goldberg & Levy, 2014; Le & Mikolov, 2014) by randomly selecting 499 negative items and 499 unconnected users, as used in previous works (Hsieh et al., 2017; Li et al., 2020; Rendle et al., 2009).

**Evaluation Measures**

To evaluate the performance of a top-$K$ recommendation, we used common metrics precision (P@$K$), recall (R@$K$), mean reciprocal rank (MRR, M@$K$), and normalized discounted cumulative gain (NDCG, N@$K$), which are expressed as follows (Hsieh et al., 2017; Li et al., 2020; SeongKu et al., 2019; Wang et al., 2013):

\[
P@K = \frac{1}{|U|} \sum_{u \in U} \sum_{i \in P_u^K} \frac{ind(i, t_u)}{|P_u^K|}
\]  
\[
R@K = \frac{1}{|U|} \sum_{u \in U} \sum_{i \in P_u^K} \frac{ind(i, t_u)}{|t_u|}
\]  
\[
M@K = \frac{1}{|U|} \sum_{u \in U} \sum_{i \in P_u^K} \frac{ind(i, t_u)}{|p_i|}
\]  
\[
N@K = \frac{1}{|U|} \sum_{u \in U} \sum_{i \in P_u^K} \frac{ind(i, t_u)}{\log_2(p_i + 1)}
\]

where:
\[ \text{ind}(i, t_u) = \begin{cases} 1, & \text{if } i \in t_u \\ 0, & \text{otherwise} \end{cases} \] (19)

\( P_u^K \) is the top-K recommended item for user \( u \), \( t_u \) is the set of interaction items of user \( u \) in test set, and \( p_i \) is the rank of item \( i \) in \( P_u^K \).

**Evaluation Measures (Comparative Models)**

- **BPR** (Rendle et al., 2009): Implicit ranking model based on dot product.
- **DASR** (Wang et al., 2020a): Utilizes component-wise learnable weights for direct ranking model.
- **DBTS** (Zhang et al., 2020): Rating to ranking model with reranking.
- **LOCABAL** (Tang et al., 2013a): Rating to ranking model using connectivity of user-user network with PageRank.
- **LOCABAL+** (Ardissono & Mauro, 2020): Rating to ranking model with the reputations of user and neighbor. Due to the absence of user profiles, LOCABAL+ is only applied on Yelp-Hotel data.
- **CSML**: Proposed method that is metric learning based on user-centric loss, item-centric loss, and sociocentric loss with adaptive margin.

**Hyperparameters**

We implemented the CSML algorithm using PyTorch. We used Adam optimizer to learn the parameters with learning rate in \((0.01, 0.005, 0.001)\) (Kingma & Ba, 2015; Li et al., 2020). The hyperparameters used to optimize the models are determined as the follows. The embedding size \( d \) was set to be 100. The batch size was chosen from \((64, 128)\) by grid search. All the embedded vectors were randomly initialized with normal distribution with a mean of 0.1 and a variance of 0.3. All the margins \( m_a, n_i, h_b \) were initialized as 1. The weights of losses, \( \lambda, \gamma, \text{and } \rho \) were individually chosen from \((0.1, 1, 10)\) via a grid search.

**Results**

The performances of BPR, DASR, DBTS, LOCABAL, and CSML on Ciao data are displayed in Figure 2 and Table 1. CSML achieved the best results in both top-5 and top-10 recommendations on Ciao data. BPR achieved the second-best performances in terms of top-5 and top-10 recommendations on Ciao data.

For the Yelp-Hotel data, CSML also achieved the best performances for all evaluation metrics, as shown in Figure 3 and Table 2 in both top-5 and top-10 recommendations.

These results conclude that CSML successfully captured user-item and user-user interactions in a metric space to conduct personalized item recommendations. Even though CSML only uses binary feedbacks 0 and 1, its performance on the top- \( K \) recommendation task was better than that of the methods that use rating information. Furthermore, using the implicit direct ranking method with BPR achieved the second-best performances on both types of data. These results coincide with the fact that the best method for top- \( K \) is direct ranking.

**Ablation Study**

We now explore the contribution of the sociocentric loss in the performance improvement of the proposed algorithm. In Table 3, we set the base model (denoted as Base) that only considers user-item relations that are optimized by user-centric loss and item-centric loss with their corresponding adaptive margins. In order to evaluate the effect of an adaptive margin, in addition to the proposed
Figure 2. Performances of Algorithms on Ciao Data: P@5, P@10, R@5, R@10, M@5, M@10, N@5, and N@10

![Figure 2](image)

Table 1. Performance Table of Algorithms Experimented on Ciao

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>P@5</th>
<th>R@5</th>
<th>M@5</th>
<th>N@5</th>
<th>P@10</th>
<th>R@10</th>
<th>M@10</th>
<th>N@10</th>
</tr>
</thead>
<tbody>
<tr>
<td>BRP</td>
<td>0.0282</td>
<td>0.0277</td>
<td>0.0144</td>
<td>0.0357</td>
<td>0.025</td>
<td>0.0477</td>
<td>0.0086</td>
<td>0.0415</td>
</tr>
<tr>
<td>DASR</td>
<td>0.002</td>
<td>0.005</td>
<td>0.0001</td>
<td>0.0003</td>
<td>0.0015</td>
<td>0.0037</td>
<td>0.0002</td>
<td>0.0018</td>
</tr>
<tr>
<td>DBTS</td>
<td>0.0179</td>
<td>0.0241</td>
<td>0.0104</td>
<td>0.0274</td>
<td>0.0128</td>
<td>0.0342</td>
<td>0.0057</td>
<td>0.0295</td>
</tr>
<tr>
<td>LOCABAL</td>
<td>0.0033</td>
<td>0.0038</td>
<td>0.0018</td>
<td>0.0047</td>
<td>0.003</td>
<td>0.0069</td>
<td>0.0011</td>
<td>0.0056</td>
</tr>
<tr>
<td>CSML</td>
<td><strong>0.0322</strong></td>
<td><strong>0.0322</strong></td>
<td><strong>0.0169</strong></td>
<td><strong>0.0416</strong></td>
<td><strong>0.0259</strong></td>
<td><strong>0.0512</strong></td>
<td><strong>0.0097</strong></td>
<td><strong>0.0457</strong></td>
</tr>
</tbody>
</table>

Figure 3. Performances of Algorithms on Yelp-Hotel Data: P@5, P@10, R@5, R@10, M@5, M@10, N@5, and N@10

![Figure 3](image)
model (Base + sociocentric loss with adaptive margin, denoted as Base + SCL + AM), we also examined fixed margins applied to the sociocentric loss with Base (Base + SCL + FM).

Table 3 shows the performance of these three algorithms for Yelp-Hotel and Ciao data. For both types of data, the proposed algorithm outperforms Base, and Base + SCL + FM, which implies that an adaptive margin is more optimal than a fixed margin.

Ablation study verifies that not only user-item interactions but also user-user interactions are optimally learned through triple loss; sociocentric loss. By optimal user-user interaction capture, the improvement of recommendation performance is achieved.

**Complexity Analysis**

The time complexity of CSML is $O\left(d\left(\left|\mathcal{D}\right| + \left|\mathcal{S}\right|\right)\right)$. The time complexities of CML and SML are $O\left(d\left(\left|\mathcal{D}\right|\right)\right)$ and $O\left(d\left(\left|\mathcal{D}\right|\right)\right)$, respectively. Usually, the number of users is smaller than that of items in real-world data (Tang et al., 2012a, 2012b, 2013a; Wang et al., 2020b). Therefore, $\left|\mathcal{D}\right| + \left|\mathcal{S}\right|$ is smaller than $2\left|\mathcal{D}\right|$, and, consequently, $O\left(d\left(\left|\mathcal{D}\right| + \left|\mathcal{S}\right|\right)\right)$ is approximately the same as $O\left(d\left(\left|\mathcal{D}\right|\right)\right)$.

As negative sampling with size $k$ (small number) is applied, $\left|\mathcal{D}\right| = k\left|\mathcal{U} \times \mathcal{V}\right|$ and $O\left(d\left(\left|\mathcal{D}\right|\right)\right) = O\left(d\left(\left|\mathcal{U} \times \mathcal{V}\right|\right)\right) = \text{hold}$.

The time complexity of calculating a cosine similarity matrix is $\left|\mathcal{U}\right|\left|\mathcal{V}\right|\left|\mathcal{U}\right|$, hence, the models consisting of matrix factorization for rating prediction with a cosine similarity matrix have
Since calculating a cosine similarity matrix has a bigger time complexity, $O(|\mathcal{U}||\mathcal{V}| + d(|\mathcal{U} + \mathcal{V}|)) > O\left(d\left(|\mathcal{D}|\right)\right)$, CSML is verified as a simpler and lighter model.

CONCLUSION

In this study, we have proposed a novel algorithm called Collaborative Social Metric Learning for recommender systems in a trust network to perform top-K recommendation. CSML learns the relationships of user-item, item-item, and user-user relations optimally with three triplet losses and adaptive margins. The experimental results with two real-world data sets demonstrate the optimality of the CSML. Since CSML has a simple structure with less computational load, it can be widely used in big data situations where there is difficulty applying big and complex neural network models.

ACKNOWLEDGMENT

This work was supported by the BK21 Four program through the National Research Foundation (NRF) funded by the Ministry of Education of Korea and an Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by South Korea’s Ministry of Science and ICT (MSIT) (No. 2019-0-00079, Artificial Intelligence Graduate School Program (Korea University)).
REFERENCES


Taehan Kim is a graduate student of Master course in Department of Artificial Intelligence from Korea University, Seoul, South Korea. He graduated in Applied Statistics in 2018 from Yonsei University, Seoul, South Korea.

Wonzoo Chung is a Professor in Department of Artificial Intelligence from Korea University, Seoul, Korea. He received a PhD degree in Electrical Engineering from Cornell University, Ithaca, USA. He worked as Senior System Architect in Dotcast Inc, Seattle, WA, USA. His research interests include statistical machine learning, deep learning, and signal processing.