HyperSoRec: Exploiting Hyperbolic User and Item Representations with Multiple Aspects for Social-aware Recommendation

HAO WANG, DEFU LIAN, University of Science and Technology of China, China
HANGHANG TONG, University of Illinois at Urbana-Champaign, USA
QI LIU, ZHENYA HUANG, ENHONG CHEN*, University of Science and Technology of China, China

Social recommendation has achieved great success in many domains including e-commerce and location-based social networks. Existing methods usually explore the user-item interactions or user-user connections to predict users' preference behaviors. However, they usually learn both user and item representations in Euclidean space, which has large limitations for exploring the latent hierarchical property in the data. In this paper, we study a novel problem of hyperbolic social recommendation, where we aim to learn the compact but strong representations for both users and items. Meanwhile, this work also addresses two critical domain-issues, which are under-explored. First, users often make trade-off with multiple underlying aspect factors to make decisions during their interactions with items. Second, users generally build connections with others in terms of different aspects, which produces different influences with aspects in social network. To this end, we propose a novel graph neural network (GNN) framework with multiple aspect learning, namely HyperSoRec. Specifically, we first embed all users, items and aspects into hyperbolic space with superior representations to ensure their hierarchical properties. Then, we adapt a GNN with novel multi-aspect message-passing-receiving mechanism to capture different influences among users. Next, to characterize the multi-aspect interactions of users on items, we propose an adaptive hyperbolic metric learning method by introducing learnable interactive relations among different aspects. Finally, we utilize the hyperbolic translational distance to measure the plausibility in each user-item pair for recommendation. Experimental results on two public datasets clearly demonstrate that our HyperSoRec not only achieves significant improvement for recommendation performance but also shows better representation ability in hyperbolic space with strong robustness and reliability.

CCS Concepts: • **Information systems** → **Recommender systems**; **Data mining**; • **Computing methodologies** → **Neural networks**;

Additional Key Words and Phrases: Hyperbolic social recommendation; Multi-aspect user influence; Multi-aspect item interaction

ACM Reference Format:
Hao Wang, Defu Lian, Hanghang Tong, and Qi Liu, Zhenya Huang, Enhong Chen. 2021. HyperSoRec: Exploiting Hyperbolic User and Item Representations with Multiple Aspects for Social-aware Recommendation. *ACM Transactions on Information Systems* 1, 1, Article 1 (January 2021), 29 pages. https://doi.org/10.1145/3463913

*Corresponding author.

Authors' addresses: Hao Wang, Defu Lian, Anhui Province Key Laboratory of Big Data Analysis and Application, University of Science and Technology of China, Hefei, Anhui, 230027, China, wanghao3@mail.ustc.edu.cn,liandefu@ustc.edu.cn; Hanghang Tong, University of Illinois at Urbana-Champaign, IL, 61801, USA, htong@illinois.edu; Qi Liu, Zhenya Huang, Enhong Chen, Anhui Province Key Laboratory of Big Data Analysis and Application, University of Science and Technology of China, Hefei, Anhui, 230027, China, qiliuql@ustc.edu.cn,huangzhy@ustc.edu.cn,cheneh@ustc.edu.cn.

1046-8188/2021/1-ART1 $15.00
https://doi.org/10.1145/3463913

1 INTRODUCTION

It becomes more and more difficult for users nowadays lied in the situation of information explosion to make decisions [32]. To alleviate this information overload, recommender systems play a dominant role, which aims to provide users with personalized services by suggesting suitable items (e.g., product) instead of letting them self-seeking. In recent years, they have become the cornerstone for improving users' experience in many applications, such as e-commerce [43, 59, 76], location-based social network [35, 36], and tourism [40, 54], showing much proliferation.

In recommender systems, the key issue is to design an optimal algorithm that can predict users' preferences on items, where it is necessary to learn good representations for both users and items to describe their interactions [20]. Along this line, traditional methods explore the user-item interactions by projecting both users and items into latent space with low-dimensional representations considering their linear relationship [49, 56, 57] or non-linear relationship [16, 18, 20, 37]. Then, inspired by social theories indicating that users' preferences are highly related to their social relations (e.g., friends) [39, 46], many efforts have been devoted to exploiting users' connections for social-aware recommendation. Generally, they assume that a certain user's preference can be affected by her friends' opinions or decisions, which leads to many typical methods by introducing some social factors, such as social regularizations [24, 25, 44, 77, 83] and social features [7, 10, 55, 66, 68]. Recently, considering the fact that users' influence may not only affect their local neighbors but also propagate farther over the user-user connection network, research work further incorporate the graph neural networks to capture the utility of this diffusion for social recommendation, such as NGCF [69], GraphRec [11] and SocialGCN [72] and MCNE [67].

Though these methods have achieved great success, they usually learn both user and item representations in Euclidean space, which cannot fully capture the beneficial latent structural properties existing in relational user and item data. Specifically, first, user-item interaction graph (the degree of each user or item node in the graph) generally follows the intrinsic power-law distribution, which can often be traced back to hierarchical structures [50]. Second, user-user connection network also exhibits an underlying tree-like structure, which demonstrates that the number of users that may be connected to the central user grows exponentially. Therefore, there exist a few users with large number of degrees but many ones lying in the boundary of network [50]. Therefore, both user-item interactions and user-user connections can form the n-ary trees, where the number of nodes at distance r from the root grows exponentially as n^r. As many work suggest, such tree-like data cannot be effectively embedded in Euclidean space but are capable of being modeled in the more reliable hyperbolic space [15]. Let us take an intuitive example in Fig. 1 to explain both spaces that helps understanding. In Fig. 1(a) with a two-dimensional Euclidean space, given the radius r, the space circumference and area can be calculated as $2\pi r$ and πr^2 respectively. In such space, the number of nodes should grows polynomially to the center with respect to the radius r. Therefore, the general representation ability of Euclidean space can be summarized as square-level, which may cause high distortion embeddings if we model the tree-like relational user-item or user-user data [13, 42]. In contrast, in Fig. 1(b), given the radius r, a two-dimensional hyperbolic space (with curvature $\xi^2, \xi > 0$) has the circumference and area as $2\pi \sinh(\xi r)$ and $2\pi(\cosh(\xi r) - 1)$, respectively, both of which are exponential with respect to radius r [13, 50]. Obviously, hyperbolic space has a stronger representation ability (with exponential-level) than Euclidean space since it has a larger space given the same radius, and therefore, more nodes could be embraced. As a result, such hyperbolic space is more suitable for modeling this relational user and item data in social recommendation, which is prone to preserve the inherent tree-like hierarchical relationship in same dimensional space compared to Euclidean space. Based on this intuition, in this paper, we study a novel problem of hyperbolic social recommendation, where we aim to propose a

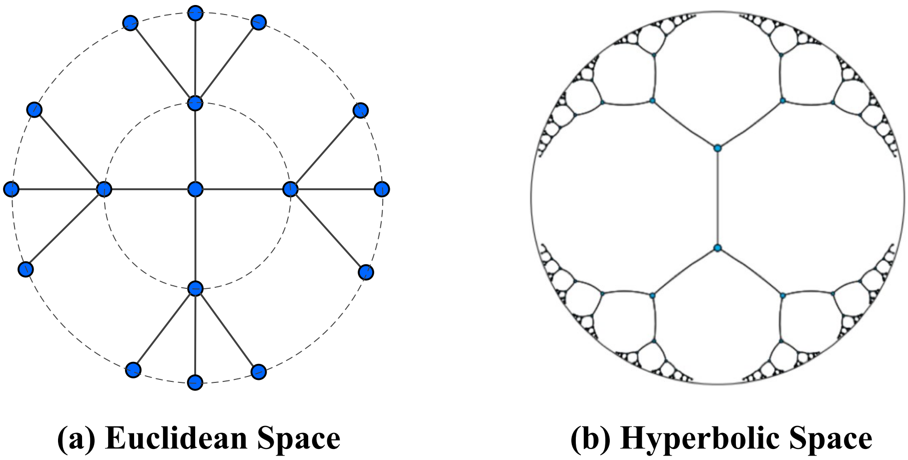


Fig. 1. An illustration example of a 2-dimensional Euclidean and hyperbolic space.

principled way for learning compact but strong user and item representations in hyperbolic space to improve the recommendation performance.

Unfortunately, there are many technical and domain challenges along this line. First, compared with commonly Euclidean space, it is even harder to learn user and item representations in hyperbolic space since we should simultaneously capture their hierarchy properties and similarity relationship. Second, users' interactions are usually influenced by different aspect factors. For example, as shown in Fig. 2, "User a" considers three underlying aspect factors to select a mobile phone including "Price", "Brand" and "Appearance", and then makes the final decision since she focuses more on "Price" aspect. Therefore, it is a non-trivial problem to explore such multi-aspect preference learning for user-item interactions in hyperbolic space. Third, in the user-user network, social users usually build connections with multiple friends and adopt their opinions on different aspects. For example, in Fig. 2, "user a" generally takes the advice from her "friend b" on the "Brand" opinions of mobile phones but trust "friend c" more on "Price" comments. So how to distinguish such differnt multi-aspect influences among user-user connections also bring us a critical challenge for designing a hyperbolic social recommendation model in practice.

To address above challenges, we propose a novel graph neural network framework with multi-aspect learning for hyperbolic social recommendation, namely HyperSoRec. Specifically, we first embed users, items and aspects with compact but strong embeddings, which are prone to preserve their inherent hierarchy properties in hyperbolic space, where we develop several specific operations based on hyperboloid model to ensure the necessary vectorial transformations for these representations. Then, we adapt a modified graph neural network framework with novel multi-aspect message-passing-receiving mechanism to distinguish users' influences with respect to different underlying aspects during the social diffusion and propagation process over the user-user network. Next, to characterize the effects of users' preferences on items with different aspects, we propose an adaptive hyperbolic metric learning method by introducing learnable interactive relations. At last, we calculate the plausibility score in hyperbolic space by using translational distance for each user-item pair. We conduct extensive experiments on two public datasets for different tasks. Experimental results not only demonstrate the significant recommendation performance of HyperSoRec but also shows the better representation ability for users and items with strong robustness and reliability in hyperbolic space. To the best of our knowledge, this is the first attempt

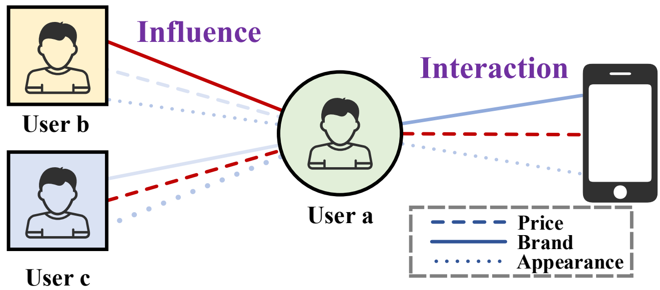


Fig. 2. Multiple aspects illustration in user connections and behaviors. Different lines represent different aspects. Brighter color means the higher effect.

to explore hyperbolic social recommendation considering both multi-aspect users' influences and item interactions simultaneously. In summary, the contributions of this article are as follows:

- In this paper, we propose a novel problem that learns compact but representative embeddings of users and items in hyperbolic space for social recommendation, which could better maintain the latent hierarchy properties between them.
- In order to address the multi-aspect users' influence and item interaction problem, we design a modified graph neural network with multi-aspect message-passing-receiving mechanism in hyperbolic space to capture different users' influences on multiple aspects, and propose a hyperbolic metric learning method to characterize the multi-aspect interactions of users on items by defining a learnable interactive relation for each specific user-item pair.
- Extensive experiments on public datasets have validated that HyperSoRec could outperform the state-of-the-art baselines with a significant margin. Besides, we further conduct embedding visualization and several comparison experiments to intuitively illustrate the effectiveness and robustness of HyperSoRec in hyperbolic space.

The rest of this paper is organized as follows. In Section 2, we introduce the related work. Then we present some preliminaries of this work in Section 3 including our problem definition and some basic knowledge. Next, Section 4 introduces technical parts of HyperSoRec model in detail and Section 5 presents the experimental results. Finally, conclusions are given in Section 6.

2 RELATED WORK

In this section, we summarize the related work into the following categories, i.e., traditional recommendation, social-aware recommendation and hyperbolic learning.

2.1 Traditional Recommendation

Recommender system is a popular topic in information retrieval and data mining domain. which has achieved great success in various applications, such as e-commerce [59, 76, 79], location-based social network [35, 36, 78], tourism [14, 40, 54] and intelligent education [21, 84]. The primary goal of it is to design an optimal algorithm that can recommend the best items to users instead of letting them self-seeking. Traditionally, research work aim to explore the user-item interactions for recommendation based on users' explicit feedback (e.g., rating) [59] or implicit feedback (e.g., click) [9, 57]. Among them, factorization models play the dominant role in the earlier time, which project users and items into latent space for describing user-item preference relationships [49, 56]. For example, Rendle et la. [57] proposed a BPR model to learn the relative preference of a user over pairs of items. Factorization Machines (FM) were proposed to model the higher-order user-item relationships considering rich side features [56]. Despite achieving great success, these models just capture user-item interactions with linear relationship (i.e., inner product), which may ignore the utility of complex user-item interactive relationship in practice.

Recently, inspired by remarkable representation performance of deep learning in various domain, such as computer vision [8] and nature language processing [47], researchers have attempted to utilize neural network architectures for recommender system [16, 18, 20, 37]. For example, He et al. [20] presented a neural collaborative filtering (NCF) model to explore the non-linear complicated user-item relationships combined with matrix factorization and feed forward neural network. One step further, NFM [18] and xDeepFM [37] were proposed to improve the recommendation performance by considering higher-order feature interactions and explicit-implicit feature interactions, respectively. Moreover, to enhance the ability of feature selection in user-item latent space, many recent work designed neural attention mechanisms to measure the feature importance for recommendation, such as AFM [74] and LRML [63].

2.2 Social Recommendation

Besides user-item interactions, many social scientists indicate that users' preferences are highly related to their social relations (e.g., friend, follow), which motivates many efforts that exploit users' social connections for improving the recommendation performance [46]. Generally, they assume that a user's preference can be affected by her neighbor friends' opinions and decisions in the social network. Along this line, on one hand, some work empirically design some social regularizations controlling that similar users share similar preferences in the factorization models [24, 25, 44, 77, 83]. For example, TrustMF [77] and ContextMF [25] consider the utility of social context and mutual trust to measure the influences between users with each other, respectively. On the other hand, a popular fashion suggests to incorporate the utility of social network for recommendations in deep learning models, where the users' relations can be viewed as a kind of beneficial auxiliary explicit information describing the relationships between user-user and user-item in the high-level embedding space [7, 62, 68]. Moreover, to capture the multi-aspect effect between social users, Chen et al. [7] proposed to use memory network with attention mechanism for the social-aware recommendation. These models directly explore the utility of first-order local neighbors' influences for social users in the domain.

Considering the fact that users' social influences may not only affect their local neighbors but also propagate farther over the user-user connection network, researchers have noticed the potentials of using graph neural networks for social recommendations [67, 69, 72, 80]. Generally, such methods treat the user-item interactions and user-user social network as the principled graph structure, where users or items can be viewed as nodes, and then leverage the graph neural network (GNN) [60] or graph convolution network (GCN) [28] to model the message passing and diffusion of social users over the network to generate the node embedding. For example, Fan et al [11] proposed a GraphRec model to capture the interactions and opinions in the user-item graph. Wu et al. [71, 72] designed SocialGCN and DiffNet++ architecture for modeling the social diffusion over the user-user graph. To further improve the performance of graph neural network adopted in recommendation problems, ESFR [81] incorporated the adversarial strategy with it, and LightGCN [19] only retained the graph convolution operation and discarded the feature transformations and non-linear activation. Considering the different-type social relationship, Xu et al. [75] proposed a relation-aware GCN model to distinguish the different connection relationship between social users, and Wang et al. [67] designed a conditional GNN for learning multiple similarities between users in both user-item and user-user graph for social recommendation. Different from them, we focus on the different aspects implied in a single relationship among users' connections.

In summary, existing methods usually model users and items in Euclidean space, where the learned representations are limited for capturing latent hierarchical properties. In this work, we target at learning both user and item representations in hyperbolic space, where we hope to keep such properties of both user-user connections and user-item interactions in social recommendations.

2.3 Hyperbolic Learning

Hyperbolic geometry learning is a kind of attractive topic which targets at learning representation of relational data to capture the inherent hierarchical structure [13, 30]. It can describe an embedding space with exponential-level representation ability, where a two-dimensional hyperbolic space (with curvature $\xi^2, \xi > 0$), as an example, has the circumference and area of radius r as $2\pi \sinh(\xi r)$ and $2\pi(\cosh(\xi r) - 1)$, respectively, both of which are exponential with respect to radius r [50]. Compared to the square-level Euclidean space (with the circumference $2\pi r$ and area πr^2, with respect to radius r), hyperbolic space owns stronger representation ability since it is capable of containing more points in the space with same dimensions [4]. In the literature, there are many popular models describing hyperbolic space as a Riemannian manifold[38], such as Poincaré ball model, Hyperboloid model[30] and Beltrami-Klein model [50, 58]. Readers who are interested in more details can refer to the corresponding works.

In the real world, there exist many relational data including biological protein graph [45], social network [77], and word frequencies [50], etc. Specifically, such relational data can be approximated with tree-like structures (n-ary trees), where the number of nodes at distance r from the root grows exponentially (as n^r), which can be effectively and smoothly modeled in hyperbolic space [15]. Holding with such strong representation ability, researchers have explored the potentials of hyperbolic space for many applications in different domain, like computer vision [15], natural language processing [50] and graph learning [6], etc, showing its effectiveness for learning hierarchical structures of complex relational data. For example, Gulcehre et al [15] proposed a hyperbolic attention network for many NLP tasks including visual question answering and machine translation. Chami et al. [6] explored an effective way to embed graph in the hyperbolic space. Recently, noticing the potentials of its ability for learning user-item complex interactions, some researchers have attempted to incorporate hyperbolic learning for recommender systems, such as [5, 12, 48, 61, 64, 65]. Based on Poincaré metric [13], [64], [65], and [12] embed users and items into hyperbolic space for recommending the items or next POI. Moreover, Schmeier et al [61] proposed a parametric empirical Bayes approach to estimate the link reliability between entities, and Mirvakhabova et al [48] adopted the Poincaré model with the variation auto-encoder for topk recommendation problem. Different from them, to the best of our knowledge, we are the first to introduce the hyperbolic space into social recommendation problem with GNN framework.

Our work improves such studies for social recommendations as follows. First, we propose a general hyperbolic framework by a principled way to learn user and item representations, where the latent hierarchy properties of both user-user connections and user-item interactions can be captured simultaneously. Second, since users can connect with each other and interact with items by multiple latent aspects, we explore both multi-aspect user influences and item interactions in hyperbolic space for social-aware recommendation.

3 PRELIMINARIES

In this section, we first formally present our problem of hyperbolic social recommendation. Then we introduce some basic knowledge including hyperboloid model and graph neural network in order to better understand our work.

3.1 Problem Definition

In the social platforms, there are a set of users $U = \{u_1, u_2, ..., u_{|U|}\}$ and a set of items $V = \{v_1, v_2, ..., v_{|V|}\}$, where $|U|$ and $|V|$ are the number of users and items respectively. Users often perform two kinds of activities including interacting with items and connecting with other users. Generally speaking, in different platforms, users show different interactions with items, e.g., users can watch/rate movies in Netflx or click/buy a clothes in Taobao [57]. Meanwhile, users can build different connections with each other, e.g., users can trust others in Epinions or follow others in Weibo [24]. Without loss of generality, we record the user-item interactions as a matrix $R \in \mathbb{R}^{|U|\times|V|}$ that reflects the users' preferences on items. If user u_i have interacted with item v_j, the corresponding matrix element value $R_{i,j}$=1, otherwise $R_{i,j}$=0. Besides, we denote user-user connections $E = \{e_{a,b}\}_{a,b=1}^{|U|}$ as the social relationship of graph $G = (U, E)$. If user u_a and u_b are linked, the value of edge $e_{ab} = 1$, otherwise $e_{ab} = 0$. As mentioned in Section 1, there exists latent hierarchy properties both in the user-item interaction matrix and user-user social graph, respectively. Therefore, it is necessary to jointly learn such latent hierarchical properties of users' influence and preference relationship for social recommendation in hyperbolic space. Formally, we define our hyperbolic social recommendation problem as follows:

Definition 1. ***(Hyperbolic Social Recommendation).*** *Given the user-item interaction matrix R and social relationship graph G, we aim to learn a function: $f(R, G) \to \hat{R}$ to predict the missing value in R, where the function $f(R, G)$ should measure the similarity of users and items in hyperbolic space.*

3.2 Hyperboloid Model

In this section, we will briefly introduce some basic concepts of hyperboloid model (also named Lorentz model), which is necessary for our work. This is the basis preliminary for describing the necessary hyperbolic space that we will use [42, 51].

Specifically, we should first introduce the basic process of **Lorentzian inner product** $\langle\cdot,\cdot\rangle_L$ for two vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^{n+1}$, which is defined as follows:

$$\langle \mathbf{x}, \mathbf{y}\rangle_L = -\mathbf{x}_0\mathbf{y}_0 + \sum_{i=1}^{n} \mathbf{x}_n\mathbf{y}_n. \tag{1}$$

Then the **hyperboloid model** which describes an n-dimensional hyperbolic space can be defined as a Riemannian manifold $(\mathbb{H}^n, g_x^{\mathbb{H}})$, where notation $\mathbb{H}^n = \{\mathbf{x} \in \mathbb{R}^{n+1} : \langle \mathbf{x}, \mathbf{x}\rangle_L = -1, \mathbf{x}_0 > 0\}$ denotes the upper sheet of a two-sheeted n-dimensional hyperboloid[1], and $g_x^{\mathbb{H}} = \mathrm{diag}([-1, 1, ..., 1])$ is a positive-definite Riemannian metric tensor, which can calculate the length and angle between tangent vectors on the manifold[4, 58]. Without loss of generality, in the following, we use $\mathbb{H}^n$ to represent our hyperbolic space for simplification. The shortest path between two points in this hyperbolic space is defined as a geodesic. It can be seen as the generalization of a straight-line in Euclidean space [51]. Specifically, the induced **distance function** of two points $(\mathbf{x}, \mathbf{y})$ derived from the geodesic between them is defined as:

$$d_L(\mathbf{x}, \mathbf{y}) = \mathrm{arcosh}(-\langle \mathbf{x}, \mathbf{y}\rangle_L). \tag{2}$$

Furthermore, for a certain point $\mathbf{x} \in \mathbb{H}^n$ in a hyperbolic space, we can define its corresponding tangent space $\mathcal{T}_x\mathbb{H}^n = \{\mathbf{v} \in \mathbb{R}^{d+1} : \langle \mathbf{v}, \mathbf{x}\rangle_L = 0\}$ on the manifold as the first-order linear approximation of $\mathbb{H}^n$ around point $\mathbf{x}$. Then we can utilize the **exponential and logarithmic map** operations [13] to map points between tangent space and hyperbolic space. Formally, both mapping operations,

[1]Here we consider $\langle \mathbf{x}, \mathbf{x}\rangle_L = -1$ and explore the trainable curvature as further work.

Table 1. The key mathematical notations.

Notation	Description
$\mathbb{H}^d$	a hyperbolic space $\mathbb{H}^d$ of dimension d
$U^{\mathbf{E}} \in \mathbb{R}^{\lvert U \rvert \times d}$	the user embeddings in the Euclidean space of dimension d
$V^{\mathbf{E}} \in \mathbb{R}^{\lvert V \rvert \times d}$	the item embeddings in the Euclidean space of dimension d
$S_r = \{r_m^{\mathbf{E}} \in \mathbb{R}^d \lvert m \in 1, ..., M\}$	a set of aspect embeddings in the Euclidean space of dimension d
$U^{0,\mathbf{H}} \in \mathbb{H}^d$	the user embeddings in the hyperbolic space of dimension d
$V^{\mathbf{H}} \in \mathbb{H}^d$	the item embeddings in the hyperbolic space of dimension d
$S_r^{\mathbf{H}} = \{r_m^{\mathbf{H}} \in \mathbb{H}^d \lvert m = 1, .., M\}$	a set of aspect embeddings in the hyperbolic space of dimension d
$\mathcal{T}_o\mathbb{H}^d$	the tangent space at origin o with dimension d
L	the layer number of graph neural network
M	the number of aspects

i.e., can be defined as:

$$\mathcal{T}_x\mathbb{H}^n \rightarrow \mathbb{H}^n := \exp_{\mathbf{x}}(\mathbf{v}) = \cosh(\|\mathbf{v}\|_L)\mathbf{x} + \sinh(\|\mathbf{v}\|_L)\frac{\mathbf{v}}{\|\mathbf{v}\|_L}, \tag{3}$$

$$\mathbb{H}^n \rightarrow \mathcal{T}_x\mathbb{H}^n := \log_{\mathbf{x}}(\mathbf{y}) = \frac{\operatorname{arcosh}(-\langle\mathbf{x},\mathbf{y}\rangle_{\mathrm{L}})}{\sqrt{\langle\mathbf{x},\mathbf{y}\rangle_L^2 - 1}}(\mathbf{y} + \langle\mathbf{x},\mathbf{y}\rangle_L\mathbf{x}), \tag{4}$$

where $\|\mathbf{v}\|_L = \sqrt{\langle\mathbf{v},\mathbf{v}\rangle_L}$ is the norm of vector. Specifically, **exponential map** $\exp_{\mathbf{x}}(\mathbf{v})$ projects a tangent vector $\mathbf{v} \in \mathcal{T}_x\mathbb{H}^n$ at point $\mathbf{x}$'s tangent space into hyperbolic space $\mathbb{H}^n$, and **logarithmic map** $\log_{\mathbf{x}}(\mathbf{y})$ is the reverse projection operation to transform a vector at point $\mathbf{y} \in \mathbb{H}^n$ in hyperbolic space into the corresponding point $\mathbf{x}$'s tangent space $\mathcal{T}_x\mathbb{H}^n$. Based on such exponential and logarithmic maps, several operations in Euclidean space can be achieved in hyperbolic space [6, 13, 42], where the details will be discussed in Section 4. Besides, there is another important mapping operation, i.e., **parallel transport**, which is a generalization of translation in Riemannian geometry. It transports a tangent vector $\mathbf{v} \in \mathcal{T}_x\mathbb{H}^n$ in point $\mathbf{x}$'s tangent space to the tangent space $\mathcal{T}_y\mathbb{H}^n$ of another point $\mathbf{y}$, which is defined as follows:

$$P_{\mathbf{x}\rightarrow\mathbf{y}}(\mathbf{v}) = \mathbf{v} - \frac{\langle\log_{\mathbf{x}}(\mathbf{y}),\mathbf{v}\rangle_L}{d_L(\mathbf{x},\mathbf{y})^2}\big(\log_{\mathbf{x}}(\mathbf{y}) + \log_{\mathbf{y}}(\mathbf{x})\big). \tag{5}$$

Please note that in hyperbolic geometry, there exist many equivalent models of hyperbolic spaces such as Poincaré ball model and Beltrami-Klein model [50, 58]. However, designing deep hyperbolic method based on such models may cause normally compound numerical issues since it needs to apply multiple exponential and logarithmic maps. To avoid such issue, in this work, we adopt the hyperboloid model to make it easier to optimize our model parameters. Readers who are interested in the hyperboloid model can refer to [30, 58] for more detailed discussions.

3.3 Graph Neural Network

In recent years, graph neural network (GNN) is a kind of hot technique, which has attracted a lot of attentions from both academia and industry, because it can effectively capture the structure information of graph to learn better node embeddings [17, 28, 31]. Generally, GNN produces several graph layers with the message-passing-receiving mechanism, which can iteratively aggregating neighbors' information, so as to embed each node with a low-dimensional vector. Specifically, given a graph $G = (U, E)$, the single-layer network contains two necessary operations, i.e., message

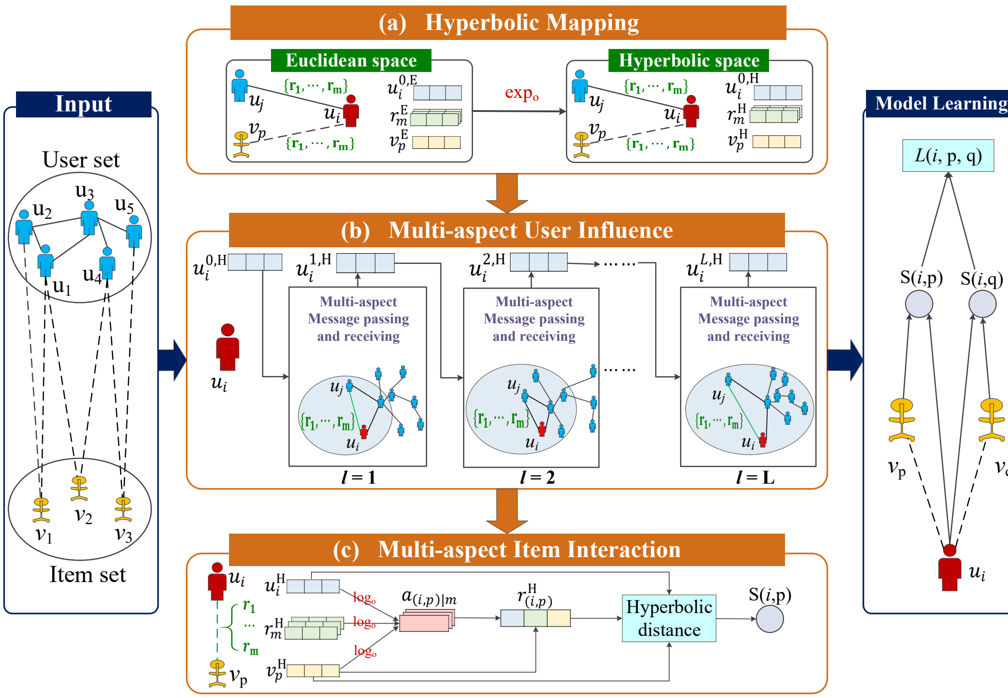


Fig. 3. HyperSoRec framework with multiple aspects for hyperbolic social recommendation.

passing ($v \longrightarrow e$) and message receiving ($e \longrightarrow v$), which are defined as follows:

$$v \longrightarrow e : h_i^l = W^l x_i^{l-1} + b^l, \tag{6}$$

$$e \longrightarrow v : x_i^l = \sigma(h_i^l + \sum_{j \in \mathcal{N}(i)} a_{ij} h_j^l), \tag{7}$$

where W^l and b^l are the weight and bias parameters at layer l, $\mathcal{N}(i) = \{j : (i, j \in E)\} \cup \{i\}$ is the a set of neighbors of node v_i, notation a_{ij} denotes the weight relationship between node v_i and node v_j, and $\sigma(\cdot)$ is a non-linear activation function, e.g., Sigmoid function and ReLU function [1]. Through Eq. (6) and Eq. (7), the node v_i first sends its message h_i^l to its surrounding neighbors, and receives the incoming messages to update its embedding vector x_i^l in the next layer. By further stacking multiple layers to perform several message-passing-receiving operations, we can capture the high-order structural information of graph for generating the final node representations, where the message of nodes can be iteratively propagated over the network. In this work, our model improves this typical message-passing-receiving mechanism, in order to capture the multi-aspect influences among users in hyperbolic space. We will discuss the technical details in Section 4.3.

4 HYPERSOREC FRAMEWORK

In this section, we first briefly illustrate the framework overview of our proposed model HyperSoRec. Then we introduce the technical details of each part. Finally, we present how to optimize model parameters in hyperbolic space.

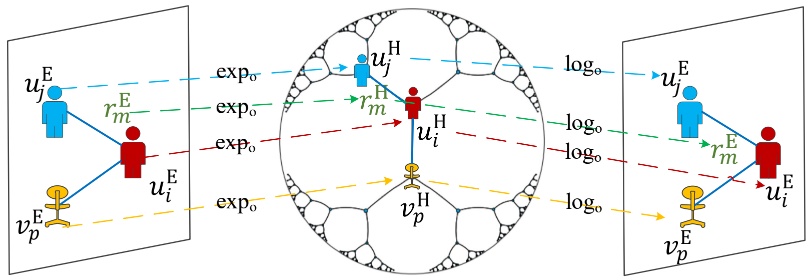


Fig. 4. The hyperbolic mapping process of embeddings between the Euclidean space and hyperbolic space

4.1 Framework Overview

In this paper, we propose a novel graph neural network framework with multi-aspect learning for hyperbolic social recommendation, namely HyperSoRec. Fig. 3 presents its architecture, which mainly consists of three components, i.e., *Hyperbolic Mapping*, *Multi-aspect User Influence Layer* and *Multi-aspect Item Interaction Layer*. Specifically, we first embed users, items along with multiple aspects in a low-dimensional Euclidean space, and project them into a hyperbolic space. Then, we design a graph neural network with novel multi-aspect message-passing-receiving mechanism in hyperbolic space, in order to capture the different influences of users on multiple aspects. Furthermore, we propose a novel hyperbolic metric learning method by introducing attention network for characterizing the multi-aspect interactions of users on items, so as to obtain the learnable relations with respect to specific user-item pairs. At last, we utilize the translational distance in hyperbolic space to calculate the plausibility score of users' preference for recommendation. In the following, we will elaborate the technical details of each component.

4.2 Hyperbolic Embedding Mapping

First of all, given the user-item interaction matrix R and user-user social graph G, the *Mapping Layer* (Fig. 3 (a)) aims to project all Euclidean embeddings of users and items into a hyperbolic space based on hyperboloid model, in order to preserve latent hierarchical properties between them. Specifically, for all users and items, we first project them into a Euclidean space, which are denoted as two embedding matrices, i.e., user Euclidean embedding matrix $U^{0,\mathbf{E}} \in \mathbb{R}^{|U|\times d_0}$ and item Euclidean embedding matrix $V^{\mathbf{E}} \in \mathbb{R}^{|V|\times d_0}$, where d_0 is the dimension of embedding vector. Meanwhile, we assume there exist M aspects among user-user connections and user-item interactions that do have effects, and also embed them in the same space with aspect Euclidean embeddings $S_r = \{r_m^{\mathbf{E}} \in \mathbb{R}^{d_0} | m \in 1, ..., M\}$, where notation M is the number of aspects. After all initializations, we transform all these Euclidean embeddings into the tangent space $\mathcal{T}_o\mathbb{H}^{d_0}$ of origin $o := \{1, 0, ..., 0\} \in \mathbb{H}^{d_0}$ in d_0-dimensional hyperbolic space. Specifically, for a Euclidean embedding $x^{\mathbf{E}}$ (of each user, item or aspect), such process of transformation operation is described as follows:

$$\mathrm{Proj}_{\mathcal{T}_o\mathbb{H}^{d_0}}(x^{\mathbf{E}}) = (0, x^{\mathbf{E}}), \tag{8}$$

where the $x^{\mathbf{E}}$ is the original vector representation of each user, item or aspect x in Euclidean space. First of all, in Eq. (8), we add an additional dimension with a value of 0 to obtain the projected Euclidean embeddings $(0, x^{\mathbf{E}})$ in tangent space at origin o, which satisfies the requirement that $\langle (0, x^{\mathbf{E}}), o\rangle_L = 0$. Then we can utilize exponential map (Eq. (3)) at origin o to map this projected Euclidean embedding $(0, x^{\mathbf{E}})$ into the hyperbolic embedding $x^{\mathbf{H}}$ in hyperboloid model as follows:

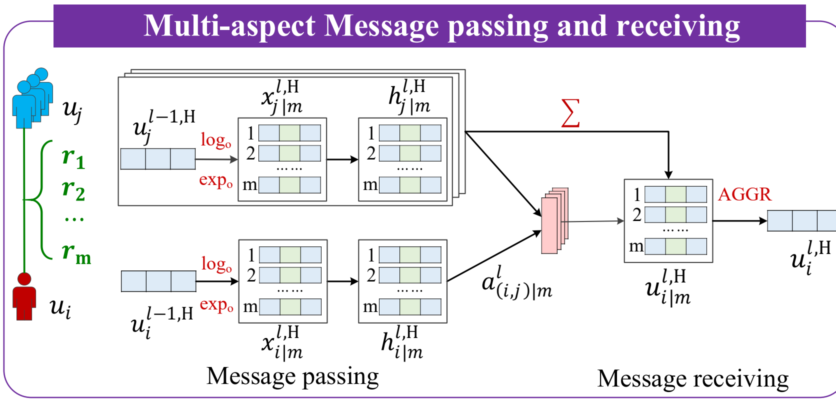


Fig. 5. Multi-aspect message-sending-receiving mechanism.

$$\mathbf{x}^{\mathrm{H}} = \exp_{\mathbf{o}}\big((0, \mathbf{x}^{\mathrm{E}})\big) = \Big(\cosh(\|\mathbf{x}^{\mathrm{E}}\|_2), \sinh(\|\mathbf{x}^{\mathrm{E}}\|_2)\frac{\mathbf{x}^{\mathrm{E}}}{\|\mathbf{x}^{\mathrm{E}}\|_2}\Big). \tag{9}$$

According to these operations, we can map all users, items and aspects to obtain the corresponding hyperbolic embeddings: $U^{0,\mathrm{H}} \in \mathbb{H}^{d_0}$, $V^{\mathrm{H}} \in \mathbb{H}^{d_0}$ and $S_r^{\mathrm{H}} = \{r_m^{\mathrm{H}} \in \mathbb{H}^{d_0} | m = 1, .., M\}$. The hyperbolic mappling process is illustrated in Fig. 4.

Particularly, in practical scenarios, the aspects can be defined as users' profiles (e.g., gender, age), items' attributes (e.g. category, brand) and some other factors (e.g., price, appearance), which are formalized with an inductive problem under our framework. (Please note that exploring such induction is not the main focus of this work, where we will leave it in the future.)

Through the *Hyperbolic Mapping* layer, we can obtain compact but representative embeddings for users, items and aspects in hyperbolic space, so as to maintain their latent hierarchical properties between them as suggested by [13]. We will further discuss this in later Section 5.3.2.

4.3 Modeling Multi-aspect User Influences

After the *Mapping Layer*, now we should deal with the problem of how to model users' influence propagation among their social connections. As shown in Fig. 2, users usually build connections with different friends and adopt their opinions in terms of different aspects (e.g., price, brand). Therefore, we propose a graph neural network (L graph layers) with novel multi-aspect message-passing-receiving mechanism in hyperbolic space to capture such differnt multi-aspect influences among users during propagation process (Fig. 3 (b)). Here, we still meet a technical challenge that the traditional hyperboloid model does not define the necessary processes as similar as Euclidean vectorial operations, like vector multiplication and addition, etc. Therefore, we cannot directly apply the hyperboloid model in our HyperSoRec. To address this issue, we develop and implement the similar vectorial operations by using the exponential and logarithmic maps (Eq. (3), Eq. (4)) inspired by previous work [13, 42]. Specifically, Fig. 5 presents the detailed technique of our proposed multi-aspect message-passing-receiving mechanism, which consists of two necessary processes, i.e., multi-aspect message passing and multi-aspect message receiving.

4.3.1 Multi-aspect Message Passing. Different from traditional message passing Eq. (6), our multi-aspect message passing process makes an assumption, that a certain user would have different influences on the surrounding neighbors with respect to different aspects. Therefore, it needs to distinguish and quantify the messages on different aspects in this process. Specifically, at l layer of HyperSoRec, given a certain aspect m of user i, we define a set of aspect-specific parameters for it to obtain its message representation $\mathbf{h}_{i|m}^{l,\mathrm{H}}$ on m-th aspect to be passed to user i' neighbors, which is defined as follows:

$$\mathbf{h}_{i|m}^{l,\mathrm{H}} = W_m^{l,\mathrm{E}} \otimes \mathbf{u}_i^{l-1,\mathrm{H}} \oplus \mathbf{b}_m^{l,\mathrm{E}}, \tag{10}$$

where $\mathbf{u}_i^{l-1,\mathrm{H}} \in \mathbb{H}^{d_{l-1}}$ is hyperbolic embedding of user i at previous layer $l-1$, matrix $W_m^{l,\mathrm{E}} \in \mathbb{R}^{d_{l-1} \times d_l}$ and vector $\mathbf{b}_m^{l,\mathrm{E}} \in \mathbb{R}^{d_l}$ are Euclidean parameters with respect to m-th aspect.

In Eq. (10), please note that operations $\otimes$ and $\oplus$ respectively represent matrix vector multiplication and bias addition in hyperbolic space, which need be defined and implemented in this work. Specifically, for the multiplication implementation: $\otimes$, we first utilize the logarithmic map operation ($\log_o(\cdot)$, Eq. (4)) to project the hyperbolic embedding $\mathbf{u}_i^{l-1,\mathrm{H}}$ into the tangent space $\mathcal{T}_o\mathbb{H}^{d_{l-1}}$ of origin o, and then perform matrix vector multiplication in this Euclidean tangent space. After that, we further utilize the exponential map operation ($\exp_o(\cdot)$, Eq. (3)) to transform it into a new hyperbolic embedding $\mathbf{t}_{i|m}^{l,\mathrm{H}} \in \mathbb{H}^{d_l}$, this process is described as follows:

$$\mathbf{t}_{i|m}^{l,\mathrm{H}} = W_m^{l,\mathrm{E}} \otimes \mathbf{u}_i^{l-1,\mathrm{H}} := \exp_{\mathbf{o}}(W_m^{l,\mathrm{E}} \log_{\mathbf{o}}(\mathbf{u}_i^{l-1,\mathrm{H}})). \tag{11}$$

It's worth noting that $\log_o(\cdot)$ map is processed in the hyperbolic space $\mathbb{H}^{d_{l-1}}$ of previous layer $l-1$, but the $\exp_o(\cdot)$ map is processed in the next layer l's hyperbolic space $\mathbb{H}^{d_l}$.

Next, for the bias addition operation: $\oplus$, we build on the derivation from previous work [13, 42]. We also first define the Euclidean parameter $\mathbf{b}_m^{l,\mathrm{E}}$ as the vector in tangent space $\mathcal{T}_o\mathbb{H}^{d_l}$ of origin o. Then we parallel transport (Eq. (5)) it to another tangent vector space of the target point $\mathbf{t}_{i|m}^{l,\mathrm{H}}$, and then utilize exponential map operation ($\exp(\cdot)$, Eq. (3)) to bring this point back into the hyperbolic space $\mathbb{H}^{d_l}$, which is defined as follows:

$$\mathbf{h}_{i|m}^{l,\mathrm{H}} = \mathbf{t}_{i|m}^{l,\mathrm{H}} \oplus \mathbf{b}_m^{l,\mathrm{E}} := \exp_{\mathbf{t}_{i|m}^{l,\mathrm{H}}}(P_{\mathbf{o} \to \mathbf{t}_{i|m}^{l,\mathrm{H}}}(\mathbf{b}_m^{l,\mathrm{E}})), \tag{12}$$

where $P_{\mathbf{o} \to \mathbf{t}_{i|m}^{l,\mathrm{H}}}(\cdot)$ is the definition of parallel transport from tangent space $\mathcal{T}_o\mathbb{H}^{d_l}$ of origin o to another tangent space $\mathcal{T}_{\mathbf{t}_{i|m}^{l,\mathrm{H}}}\mathbb{H}^{d_l}$ of point $\mathbf{t}_{i|m}^{l,\mathrm{H}}$ in the hyperbolic space $\mathbb{H}^{d_l}$.

Finally, through Eq. (11) and Eq. (12), we can obtain multiple message passing representations $\{\mathbf{h}_{i|1}^{l,\mathrm{H}}, ..., \mathbf{h}_{i|M}^{l,\mathrm{H}}\}$ on different aspects for user i at a specific layer l of HyperSoRec.

4.3.2 Multi-aspect Message Receiving. Similarly, our multi-aspect message receiving further assumes that a user would receive the messages from her neighbors' friends on different aspects, which is also superior to traditional message receiving mechanism Eq. (6). Therefore, we need to quantify how much she could update the information from her neighbors on each aspect in hyperbolic space. Mathematically, for a certain user i, we first define her neighbor set $\mathcal{N}(i)$ consisting of her sampled neighbors (with a fixed size) and herself. We will make detailed discussion about the number of neighbor set in the experiment Section 5.3.2. Then, we can update her hyperbolic embedding $\mathbf{u}_{i|m}^{l,\mathrm{H}}$ about aspect m at layer l as:

$$\mathbf{u}_{i|m}^{l,\mathrm{H}} = \mathrm{AGGR}_m^l(i) := \exp_{\mathbf{h}_{i|m}^{l,\mathrm{H}}}\left(\sigma\Big(\sum_{j \in \mathcal{N}(i)} a_{(i,j)|m}^l \log_{\mathbf{h}_{\mathrm{i|m}}^{\mathrm{l,H}}}(\mathbf{h}_{j|m}^{l,\mathrm{H}})\Big)\right), \tag{13}$$

where $\mathbf{h}_{j|m}^{l,\mathrm{H}}$ is the hyperbolic message representation of her neighbor j on aspect m at layer l by Eq.(10). Value $a_{(i,j)|m}^l$ means the corresponding influence weight of neighbor j on m-th aspect.

In Eq. (13), by utilizing the logarithmic map operation ($\log(\cdot)$, Eq. (4)) to project all the passing message representation $\mathbf{h}_{j|m}^{l,\mathbf{H}}$ of neighbor j on m-th aspect into the same tangent space $\mathcal{T}_{\mathbf{h}_{i|m}^{l,\mathbf{H}}}\mathbb{H}^{d_l}$ of current point $\mathbf{h}_{i|m}^{l,\mathbf{H}}$, we can make a better approximation in Euclidean space for hyperbolic space to achieve lower distortion [6, 13]. Then in this tangent space, we accumulate the passing message on m-th aspect from all neighbors for user i according to the weight scores $a_{(i,j)|m}^{l}$, and utilize the non-linear activation function $\sigma(x) = \mathrm{ReLU}(x) = max(0, x)$ to update the representation of user node i. Furthermore, we perform the exponential map operation ($\exp(\cdot)$, Eq. (3)) to transform it back to the hyperbolic space, in order to obtain her updated hyperbolic embedding $\mathbf{u}_{i|m}^{l,\mathbf{H}}$ on m-th aspect at next layer l.

Next, for the definition of $a_{(i,j)|m}$, we design an attention network by softmax operation based on the hyperbolic distance $d_L(\cdot,\cdot)$ (Eq. (2)) between two user nodes i and j on aspect m as:

$$a_{(i,j)|m}^{l} = \mathrm{SOFTMAX}_{j\in\mathcal{N}(i)}\big(\beta \cdot d_L(\mathbf{h}_{i|m}^{l,\mathbf{H}}, \mathbf{h}_{j|m}^{l,\mathbf{H}}) + \gamma\big), \tag{14}$$

where β and γ are scalar Euclidean parameters. By Eq. (14), we can make the nodes closer in hyperbolic space have more larger value. Then these neighbors with larger weights would produce more influence with respect to aspect m for message receiving information on user i.

By Eq. (13) and Eq. (14), we can obtain a set of user i's hyperbolic embeddings $\{\mathbf{u}_{i|1}^{l,\mathbf{H}}, ..., \mathbf{u}_{i|M}^{l,\mathbf{H}}\}$ on all M aspects. Then we conduct a simple cumulative pooling operation to aggregate these representations for the final hyperbolic embedding $\mathbf{u}_{i}^{l,\mathbf{H}}$ of user i, which is described as follows:

$$\mathbf{u}_{i}^{l,\mathbf{H}} = \mathrm{POOLING}^{l}(i) = \exp_{\mathbf{o}}\Big(\sum_{m=1}^{M}\log_{\mathbf{o}}(\mathbf{u}_{i|m}^{l,\mathbf{H}})\Big). \tag{15}$$

In Eq. (15), since the addition operation in hyperbolic space does not satisfy the commutativity or associativity [13, 82], we have to calculate these embeddings by order like $\Big(\big((\mathbf{u}_{i|1}^{l,\mathbf{H}} \oplus \mathbf{u}_{i|2}^{l,\mathbf{H}}) \oplus \mathbf{u}_{i|3}^{l,\mathbf{H}}\big) \oplus ...\Big)$. Thus we utilize logarithmic and exponential maps (Eq. (4) and Eq. (3)) to conduct the cumulative pooling operation in the tangent space $\mathcal{T}_o\mathbb{H}^{d_l}$, in order to accelerate our algorithm. After that, through such entire multi-aspect message receiving operation Eq.(13) and Eq.(15), we can capture the different influences of neighbor nodes on multiple aspects, and update the hyperbolic embedding $\mathbf{u}_{i}^{l,\mathbf{H}}$ of user i at next layer l.

Finally, through Eq. (10)~Eq. (15), we are able to capture the users' influences on M aspects in immediate neighborhood. Then we can continuously stack L layers to characterize the propagation of surrounding L-order users' influences (Fig. 3 (b)). At last, we denote the output hyperbolic embedding of user i at last layer L as its final representation for simplicity, i.e., $\mathbf{u}_{i}^{\mathbf{H}}$ $(\in \mathbb{H}^{d}) = \mathbf{u}_{i}^{L,\mathbf{H}}$.

4.4 Modeling Multi-aspect Item Interactions

Next, our goal is to estimate the preference relationship of user-item pairs. Please recall that in Fig. 2, users usually consider multiple underlying aspects (e.g., price, brand) to make the final decisions. Therefore, motivated by this intuition, we introduce a novel adaptive hyperbolic metric learning method to calculate the plausibility of a specific user-item pair (u, v), which is based on the translational distance $u + r \approx v$, where r denotes relational vector between user u and item i, by considering the utility of such multi-aspect item interactions (Fig. 3 (c)).

4.4.1 Multi-aspect Interactive Relation. For a certain user-item pair (i, p), we first define its multi-aspect interactive vector $\mathbf{r}^{\mathbf{H}}_{(i,p)}$ as follows:

$$\mathbf{r}^{\mathbf{H}}_{(i,p)} = \exp_{\mathbf{o}}\Big(\sum_{m=1}^{M} a_{(i,p)|m} \log_{\mathbf{o}}(\mathbf{r}^{\mathbf{H}}_{m})\Big), \tag{16}$$

where $\mathbf{r}^{\mathbf{H}}_{m}$ is the corresponding hyperbolic embedding of m-th aspect and $a_{(i,p)|m}$ is the weight score of user i to item p on this aspect. Similarly, we utilize the logarithmic and exponential maps (Eq. (4) and Eq. (3)) to aggregate hyperbolic embeddings from different aspects, and obtain the final specific interactive vector $\mathbf{r}^{\mathbf{H}}_{(i,p)}$ for each user-item pair (i, p).

We also introduce an attention network to calculate weight score $a_{(i,p)|m}$ in Eq. (16) as:

$$a_{(i,p)|m} = \mathrm{MLP}\Big(\big(\log_{\mathbf{o}}(\mathbf{u}^{\mathbf{H}}_{i}) \odot \log_{\mathbf{o}}(\mathbf{v}^{\mathbf{H}}_{p})\big) \| \log_{\mathbf{o}}(\mathbf{r}^{\mathbf{H}}_{m})\Big), \tag{17}$$

where $\odot$ denotes the element-wise operation, $(\cdot\|\cdot)$ denotes the concatenation of vectors, MLP is the Euclidean Multi-layer Perception. We utilize logarithmic map to project hyperbolic embeddings of user i, item p and each aspect relation $r^{\mathbf{H}}_{m}$ into the Euclidean tangent space $\mathcal{T}_{o}\mathbb{H}^{d}$. Then we conduct the element-wise operation to obtain the combined embedding of user and item, and concatenate it with the relation vector as the input of attention network. By Eq. (17), we can assign greater weights to the aspects that are more important for users in the interactive relations, in order to make the users and items more similar on the corresponding aspects.

4.4.2 Score Function. After obtaining the hyperbolic interaction relation $\mathbf{r}^{\mathbf{H}}_{(i,p)}$ for each user-item pair (i, p), we then derive its score function with translation distance formula (Eq. (2)) in hyperbolic space to obtain the final plausibility, which is defined as follows:

$$s(i,p) = d_{L}\big(\mathbf{u}^{\mathbf{H}}_{i} \oplus \mathbf{r}^{\mathbf{H}}_{(i,p)}, \mathbf{v}^{\mathbf{H}}_{p}\big) = \operatorname{arcosh}\big(-\langle \mathbf{u}^{\mathbf{H}}_{i} \oplus \mathbf{r}^{\mathbf{H}}_{(i,p)}, \mathbf{v}^{\mathbf{H}}_{p}\rangle_{L}\big), \tag{18}$$

where $\mathbf{v}^{\mathbf{H}}_{p} \in \mathbb{H}^{d}$ is hyperbolic embedding of item p. Here, we first add user vector $\mathbf{u}^{\mathbf{H}}_{i}$ and interactive relation vector $\mathbf{r}^{\mathbf{H}}_{(i,p)}$ to obtain the transitional embedding in hyperbolic space, and then calculate the hyperbolic distance $d_{L}\big(\mathbf{u}^{\mathbf{H}}_{i} \oplus \mathbf{r}^{\mathbf{H}}_{(i,p)}, \mathbf{v}^{\mathbf{H}}_{p}\big)$ between this translational embedding and item embedding $\mathbf{v}^{\mathbf{H}}_{p}$, which is regarded as the plausibility score for the user-item pair (i, p). It's worth mentioning that notation $\oplus$ is addition operation defined in hyperboloid model, so we need to first map the hyperbolic embeddings $\mathbf{u}^{\mathbf{H}}_{i}$ and $\mathbf{r}^{\mathbf{H}}_{(i,p)}$ into the tangent space and utilize the same operation defined in Eq. (12), so as to achieve the accumulation operation of them in hyperbolic space.

4.5 Model Learning

In this subsection, we will describe the details of model learning of HyperSoRec including objective function, training optimization and model complexity.

4.5.1 Objective Function. As we focus on the implicit feedbacks of users, we utilize the widely-used hinge loss [73, 80] to learn score function for model learning between users and items (Fig. 3), which is illustrated as follows:

$$\mathcal{L}(i,p,q) = \sum_{(i,p)\in\mathcal{D}} \sum_{(i,q)\notin\mathcal{D}} \max\big(0, \lambda + s(i,p)^{2} - s(i,q)^{2}\big), \tag{19}$$

where $\mathcal{D}$ is the set of all user-item pairs, q is the sampled negative item that user i haven't interacted, and λ is the margin for separating the hyperbolic distance between the positive (p) and negative (q) sample pairs. Please note that we utilize the same relation vector $\mathbf{r}^{\mathbf{H}}_{(i,p)}$ for the negative sample pair (i, q), which is motivated by our empirical results that can achieve better performance and convergence of HyperSoRec. By optimizing the objective function Eq. (19), we can make the distance between users and items in positive pair (i, p) closer, and separate the distance in negative pairs (i, q) farther. Benefit from the characteristics of hyperbolic space, we can move a point to a certain distance with a smaller force than Euclidean space [13, 50], which can make HyperSoRec better learn the compact hyperbolic embeddings of users and items that are prone to preserve the latent hierarchy properties among them.

4.5.2 Optimization. Please note that HyperSoRec contains both Euclidean parameters $\theta^{\mathbf{E}}$ (like $W_m^{l,\mathbf{E}}, \mathbf{b}_m^{l,\mathbf{E}}$, etc) and hyperbolic parameters $\theta^{\mathbf{H}}$ (like $\mathbf{u}_u^{\mathbf{H}}, \mathbf{v}_p^{\mathbf{H}}, \mathbf{r}_m^{\mathbf{H}}$, etc), and therefore, we conduct the different optimization methods for these two types of parameters. We derive their Euclidean gradients $\nabla\mathcal{L}(\theta)$ for all the model parameters $\theta = \{\theta^{\mathbf{E}}, \theta^{\mathbf{H}}\}$. On one hand, for Euclidean parameters $\theta^{\mathbf{E}}$, we can directly use the stochastic gradient algorithm (SGD) [3] for optimization. On the other hand, for hyperbolic parameters $\theta^{\mathbf{H}}$, we adopt the Riemannian stochastic gradient algorithm (RSGD) [30] for optimization as:

$$\theta_{t+1}^{\mathbf{H}} = \exp_{\theta_t^{\mathbf{H}}}(-\eta \cdot \mathrm{grad}\mathcal{L}(\theta_t^{\mathbf{H}})), \tag{20}$$

where η is learning rate, and $\mathrm{grad}\mathcal{L}(\theta_t^{\mathbf{H}})$ is the gradient of hyperbolic parameter $\theta_t^{\mathbf{H}}$ defined in the Riemannian manifold. In order to obtain it, we first multiply it's Euclidean gradient $\nabla\mathcal{L}(\theta_t^{\mathbf{H}})$ by Lorentz metric $g_x^{\mathbb{H}}$ (defined in Section 3.2) to obtain the steepest descent direction $\mathbf{h}_t$, and then project it into the corresponding tangent space of current parameter $\theta_t^{\mathbf{H}}$ to get the Riemannian gradient, which is defined as follows:

$$\mathrm{grad}\mathcal{L}(\theta_t^{\mathbf{H}}) = \mathrm{proj}_{\theta_t^{\mathbf{H}}}(\mathbf{h}_t) = \mathbf{h}_t + \langle\theta_t^{\mathbf{H}}, \mathbf{h}\rangle_L \theta_t^{\mathbf{H}}. \tag{21}$$

After that, we can combine the learning rate η with exponential map to obtain the updated hyperbolic parameter $\theta_{t+1}^{\mathbf{H}}$. In order to find a better optimal solution and accelerate the model convergence, we adopt the Adam [27] and RAMSGrad [2] algorithms to optimize the Euclidean and hyperbolic parameters in HyperSoRec, respectively.

4.5.3 Time Complexity. In Section 4.3, we can observe that the computational complexity of generating hyperbolic embeddings for all users is very high. Although in hyperbolic space, we can still adopt the standard mini-batch training [17, 67] for acceleration to alleviate this problem, so as to apply HyperSoRec to the large-scale social networks. With this mini-batch setting, the complexity of HyperSoRec is fixed at $O(C\,B\,\Pi_{i=1}^{L}|\mathcal{N}_i|)$, where C is the number of negative item samples, B is the number of nodes in each batch, L is the layer number of our GNN , and $|\mathcal{N}_i|$ is the number of sampling neighbors in each layer. In general optimization situation, we often set $L = 2$ to achieve satisfactory results. Such time complexity is acceptable. Thus HyperSoRec could be applied to the real-world recommendation systems. More discussions of HyperSoRec model settings can be found in Section 5.3.2.

In summary, our proposed HyperSoRec framework mainly has the following advantages. First, it provides a principled way to learn compact but representative embeddings of users and items for hyperbolic social recommendation, which could preserve their latent hierarchy properties in hyperbolic space. Second, HyperSoRec holds a GNN with novel multi-aspect message-sending-receiving mechanism in hyperbolic space to capture different users' influences on multiple aspects. Third, HyperRoRec incorporates a novel adaptive hyperbolic metric learning method to model the multi-aspect item interactions for recommendation. Last but not least, all the Euclidean and hyperbolic parameters in HyperSoRec are optimized simultaneously in a unified learning framework, and could be applied to the large social networks under an end-to-end mini-batch training strategy.

Table 2. Statistics of the Datasets.

Datasets	#Users	#Links	Tasks	#Items	#Interactions
Ciao	4,321	121,408	Beauty	9,243	23,091
			Book	12,409	21,105
			Travel	11,899	20,857
Epinions	10,459	280,258	Game	6,804	30,417
			Electronics	12,425	30,429
			Travel	11,885	38,578
Yelp	17,237	143,765	/	38,342	204,448

5 EXPERIMENTS

In this section, we will conduct extensive experiments to evaluate the performance of HyperSoRec framework. Specifically, we first describe the datasets and experimental setup (Section 5.1). Then, we demonstrate the effectiveness of HyperSoRec compared with several baselines (Section 5.2). At last, we provide detailed analyses about HyperSoRec (Section 5.3).

5.1 Experimental Dataset and Setup

5.1.1 Datasets. In the experiments, we use three publicly available datasets, i.e., Ciao, Epinions[2] and Yelp [72]. Specifically, Ciao and Epinions are two popular who-trust-whom online social platforms, which both record two kinds of users' behaviors. First, users can consume products which belong to several different categories (e.g., "Book" in Ciao and "Game" in Epinions). Second, users can browse others' comments on products and then "trust" ones who write good comments, which establishes the trust network among users. Moreover, Yelp is a well-known online location-based social network, where users can make friends with each other and rate the restaurants that they have consumed. For these datasets, we make the following assumptions in this paper: First, users usually take advices from their trusted friends when consuming a product or restaurant. Second, users may trust others or make friends with respect to different aspects, e.g., a user can trust the one who usually makes plausible opinions on the "price" of products, but trusts another who are familiar with products' "brand". Third, users generally consume products or rate positively to restaurants with respect to different aspects as well, e.g., a user may buy a book since she like the "author". Therefore, it is necessary to combine both users' behaviors with considering the multiple aspects when generating the recommendations.

For Ciao and Epinions datasets, it worth mentioning that users' consumption behaviors for different categories may be related to different aspects. For example, we may consider the aspects like "author" or "style" when choosing a "Book" but focus on the aspects like "distance" or "cost" for "Travel". Therefore, in order to avoid such confusion, we respectively select three representative categories in both datasets, i.e., "Beauty", "Book", "Travel" in Ciao and "Game", "Electronics", "Travel" in Epinions, and conduct the recommendation experiments on them as different tasks. In Yelp, we do not split the data and use all users' rating behaviors as one task. Moreover, to ensure the reliability of experimental results, we filtered out the users that had less than 2 social links and 2 item records. The detailed statistics of all datasets after preprocessing are presented in Table 2.

5.1.2 Comparison Methods. To demonstrate the effectiveness of HyperSoRec, we select several state-of-the-art methods from three perspectives. Specifically, we first choose two models only considering user-item interactions for recommendation, i.e., BPR and LRML. Then we select two

[2] https://www.cse.msu.edu/~tangjili/trust.html

Table 3. Characteristics of all models.

Model	User-item Interaction		User-user Influence		Embedding Space	
	Consider?	Multiple aspects?	Consider?	Multiple aspects?	Euclidean	Hyperbolic
BPR [57]	√	×	×	-	√	×
LRML [63]	√	√	×	-	√	×
HyperBPR [64]	√	×	×	-	×	√
HyperML [65]	√	×	×	-	×	√
FM [56]	√	×	√	×	√	×
NMF [20]	√	×	√	×	√	×
GraphRec [11]	√	×	√	√	√	×
SocialGCN [72]	√	×	√	√	√	×
LightGCN [19]	√	×	√	√	√	×
HyperSoRec(E)	√	√	√	√	√	×
HyperSoRec-I	√	√ (Average)	√	√ (Eq. (17))	×	√
HyperSoRec-U	√	√ (Eq. (14))	√	√ (Average)	×	√
HyperSoRec	√	√	√	√	×	√

hyperbolic models capturing the latent hierarchy properties existing in user-item interactions, i.e., HyperBPR and HyperML. We also introduce five typical social-aware algorithms incorporating social connections for recommendation, i.e., FM, NMF, GraphRec, SocialGCN and LightGCN. The details of them are as follows:

- **BPR**[57]: BPR is a typical latent factor method for modeling users' implicit feedback on items. It designs a pairwise raking function to learn the preferences of a user over pairs of items.
- **LRML**[63]: LRML employs a augmented memory module to learn the latent relations of each user-item pair, and utilizes the metric learning method to optimize the model for recommending items to users.
- **HyperBPR**[64]: HyperBPR designs the distance function based on Poincaré model to measuring user-item pairs in a hyperbolic space, and takes use of the criterion in BPR model to optimize recommendations.
- **HyperML**[65]: HyperML explores the metric learning in hyperbolic space based on Möbius gyrovector spaces of Poincaré model for personalized ranking recommendation.
- **FM**[56]: FM has shown strong performance for personalized recommendation, where higher-order interactions of features are considered. For our problem, we utilize the adjacency matrix of users' social relationship as its own attribute features.
- **NMF**[20]: NMF utilizes the deep neural network to capture the higher-order user-item feature interactions, along with matrix factorization to improve recommendation performance.
- **GraphRec**[11]: GraphRec is the latest graph neural network framework for social recommendation, which jointly captures the interactions and opinions between users and items.
- **SocialGCN**[72]: SocialGCN is a state-of-the-art model with a layer-wise propagation structure to model the recursive dynamic users' influences in social recommendation.
- **LightGCN**[19]: LightGCN is the most competitive graph-based model for social recommendation recently. It only retains the essential convolution operations and abandons the feature transformation and nonlinear activation in common graph neural networks.

Moreover, to highlight the effectiveness of each part in our HyperSoRec including hyperbolic representation, multi-aspect user influence and multi-aspect item interaction, we introduce the following variants of HyperSoRec as:

- **HyperSoRec(E)**: HyperSoRec(E) can be viewed as a simplified version of HyperSoRec, that performs all operations and embeddings in Euclidean space. It still considers both multi-aspect users' influences and item interactions when generating recommendations.
- **HyperSoRec-I**: HyperSoRec-I is a reduced version of HyperSoRec. Here, we keep the same multi-aspect user-user influence learning as HyperSoRec, but ignore the different multi-aspect importances of user-item interaction learning. Specifically, we utilize an average method to replace the weights calculated by the attention network in Eq. (17).
- **HyperSoRec-U**: HyperSoRec-U is another reduced version. We keep the same multi-aspect user-item interaction learning as HyperSoRec, but regard the user-user influences on multiple aspects as the same (averaging scores) instead of the attention weights calculated by Eq. (14) .

For better illustration, we summarize the characteristics of these models in Table 3.

5.1.3 Evaluation Protocols. We conduct recommendation experiments on Yelp and each task of Ciao and Epinions datasets. For each task or dataset, to start up the experiments, we randomly select 70% of users' consumption data as training set, 10% as validation set, and the remaining 20% as test set. Then, we evaluate the ranking performance for the recommendation of all models, i.e., we target at providing a ranking list with recommended products for each user [26, 33]. To obtain more rigorous experiments results, as Krichene et al. [29] suggested, we replace the sampling strategy used in previous work [34, 72], and regard all the items that the user has not interacted with as the candidates, so as to alleviate the biased results. Furthermore, for the evaluation metric, we selected two widely used top@K ranking metrics including Recall and NDCG. Besides, we truncate the ranked list with different top@K values K=[5, 10, 20] for both metrics, and observe the similar trends in these results. Therefore, we only report the experimental results of K=5 for better illustration as the representative. Finally, we repeat each experiment 10 times independently and report the average ranking results to ensure the reliability.

5.1.4 Parameter Setting. There are several hyper-parameters to be specified in HyperSoRec framework. First, we set the number of graph network layer for our multi-aspect influence propagation part (Section 4.3) as L=2, where the corresponding dimensions of each layer are defined as [100, 50, 50], with the sampled neighbor sizes $\mathcal{N}(i)$ of each layer are [20, 15]. Then we set the dimension of item and aspect embeddings as 50 in accordance with user embeddings at the last network layer. (We will make the detailed analyses to show the effectiveness of embedding size in HyperSoRec in Section 5.3.2). Next, as for the attention network implementation in HyperSoRec, we leverage 2 layer feed forward neural network for the calculation (Eq. (14) and Eq. (17)). At last, We also make the grid search for the hinge loss parameter λ in Eq. (19) from the set [1.0, 2.0,..., 5.0] and select the best one in the experiments of each task.

In training stage, there are two types of model parameters to be initialized in Euclidean and Hyperbolic spaces respectively. Specifically, for the model parameters in Euclidean space, we initialize them with a Gaussian distribution with mean 0 and standard deviation 0.01. Then for the parameters in Hyperbolic space, we initialize them with a uniform distribution [-0.001, 0.001]. In addition, we set the learning rate as 0.003 and mini-batch size as 64. We also use dropout (probability value 0.3) to prevent HyperSoRec from overfitting.

In the following experiments, we implement HyperSoRec and all compared baselines by PyTorch. The parameters of all comparsion methods are set to be the same as the original settings stated in their papers and tuned to the best performance. To ensure the fairness, all the baselines and variants of HyperSoRec are implemented with same embedding sizes. We run all the experiments on a Linux server with four 2.0GHz Intel Xeon E5-2620 CPUs and a Tesla K80 GPU.

Table 4. Recommendation performance results on all datasets with Recall@5 metric.

Datasets	Ciao			Epinions			Yelp
Tasks	Beauty	Book	Travel	Electronics	Travel	Game	
BPR	0.0121	0.0258	0.0161	0.0135	0.0267	0.0534	0.0092
LRML	0.0128	0.0262	0.0166	0.0141	0.0284	0.0561	0.0094
HyperBPR	0.0135	0.0266	0.0172	0.0146	0.0272	0.0544	0.0089
HyperML	0.0138	0.0265	0.0177	0.0145	0.0269	0.0558	0.0090
FM	0.0168	0.0272	0.0184	0.0217	0.0356	0.0574	0.0101
NMF	0.0172	0.0280	0.0191	0.0224	0.03794	0.0587	0.0102
GraphRec	0.0176	0.0294	0.0201	0.0247	0.0402	0.0595	0.0108
SocialGCN	0.0176	0.0301	0.0206	0.0246	0.0392	0.0582	0.0113
LightGCN	0.0178	0.0311	0.0210	0.0251	0.0405	0.0594	0.0117
HyperSoRec	**0.0185**	**0.0328**	**0.0216**	**0.0271**	**0.0427**	**0.0606**	**0.0126**

5.2 Experimental Results

5.2.1 Recommendation Performance Comparison. Table 4 and Table 5 report the overall recommendation performance by ranking metrics on both datasets for each task or dataset. We can conclude several observations as follows: First, HyperSoRec consistently achieves significant improvements for all recommendation tasks on both datasets, especially on the NDCG metric. It demonstrates that HyperSoRec can better characterizes multi-aspect users' influences and multi-aspect item interactions in hyperbolic space for social-aware recommendation. Second, compared with the typical models in Euclidean space (i.e., BPR and LRML), hyperbolic models (i.e., HyperSoRec, HyperBPR, HyperML) perform better results. This proves that hyperbolic space is prone to capture the latent hierarchy properties implied in user-item data, so as to effectively enhance representation ability of the learned hyperbolic embeddings of users and items in a compact space. Third, we notice that social-based models (HyperSoRec, FM, NMF, GraphRec, SocialGCN, LightGCN) perform better than those (BPR, LRML, HyperBPR, HyperML) with just considering user-item interactions. This phenomenon demonstrates that exploring users' social connections into modeling can help learn users' preferences, so that benefits the recommendation performance. Last but not least, compared with all social models, traditional models (NMF, FM) do not perform as well as graph-based ones (SocialGCN, GraphRec, LightGCN), which proves the effectiveness of graph neural network for capturing the utility of social influence propagation in user-user connections. What's more, HyperSoRec performs even better since it not only considers both multi-aspect users' influences and multi-aspect item interactions, but also leverages a more suitable hyperbolic space for recommendation. In summary, all the results clearly show the significant performance of HyperSoRec in the social-aware recommendation task.

5.2.2 Ablation Study. In order to highlight the effectiveness of each part in HyperSoRec, we further present the recommendation performance results of it with three variants including HyperSoRec(E), HyperSoRec-I and HyperSoRec-U on all tasks of Ciao and Epinions datasets in Table 6 and Table 7. From the figures, we can conclude the following observations. First, we find models in hyperbolic space (HyperSoRec, HyperSoRec-I, HyperSoRec-U) perform better than Euclidean-based variant HyperSoRec(E), especially on NDCG metric. It proves the effectiveness of hyperbolic representation ability for learning latent hierarchal relationship existing in user-user connections and user-item interactions. Second, we notice that HyperSoRec achieves better performance than HyperSoRec-I on all datasets. This demonstrates that users' preferences on multiple aspects are

Table 5. Recommendation performance results on all datasets with NDCG@5 metric.

Datasets	Ciao			Epinions			Yelp
Tasks	Beauty	Book	Travel	Electronics	Travel	Game	
BPR	0.0124	0.0210	0.0147	0.0105	0.0233	0.0464	0.0090
LRML	0.0139	0.0244	0.0142	0.0111	0.0238	0.0499	0.0092
HyperBPR	0.0204	0.0328	0.0224	0.0153	0.0367	0.0692	0.0115
HyperML	0.0205	0.0322	0.0235	0.0152	0.0366	0.0723	0.0118
FM	0.0141	0.0228	0.0162	0.0145	0.0300	0.0428	0.0095
NMF	0.0149	0.0225	0.0170	0.0137	0.0312	0.0467	0.0097
GraphRec	0.0156	0.0236	0.0171	0.0166	0.0332	0.0476	0.0105
SocialGCN	0.0154	0.0244	0.0173	0.0167	0.0319	0.0474	0.0110
LightGCN	0.0162	0.0299	0.0177	0.0160	0.0321	0.0471	0.0120
HyperSoRec	**0.0260**	**0.0398**	**0.0330**	**0.0325**	**0.0525**	**0.0775**	**0.0153**

Table 6. Recommendation performance results of HyperSoRec variants with Recall@5.(p-value<4.38e-2)

Datasets	Ciao			Epinions		
Tasks	Beauty	Book	Travel	Electronics	Travel	Game
HyperSoRec(E)	0.0179	0.0307	0.0209	0.0266	0.0409	0.0598
HyperSoRec-I	0.0182	0.0321	0.0214	0.0264	0.0411	0.0598
HyperSoRec-U	0.0184	0.0315	0.0212	0.0268	0.0418	0.0597
HyperSoRec	**0.0185**	**0.0328**	**0.0216**	**0.0271**	**0.0427**	**0.0606**

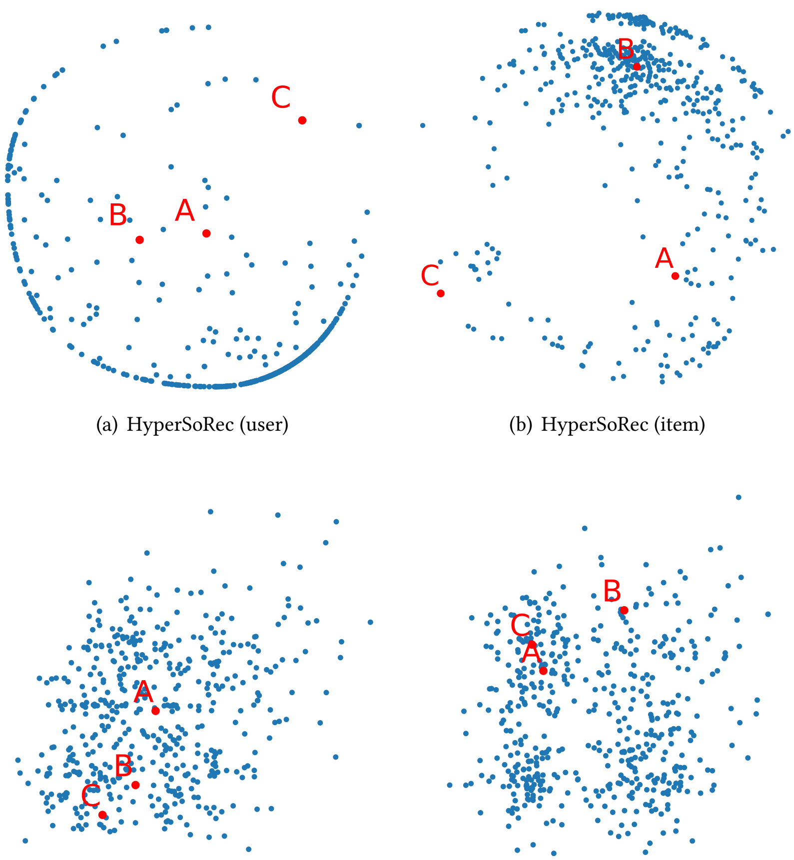
Table 7. Recommendation performance results of HyperSoRec variants with NDCG@5.(p-value<2.08e-3)

Datasets	Ciao			Epinions		
Tasks	Beauty	Book	Travel	Electronics	Travel	Game
HyperSoRec(E)	0.0154	0.0298	0.0179	0.0178	0.0341	0.0480
HyperSoRec-I	0.0249	0.0380	0.0311	0.0224	0.0521	0.0772
HyperSoRec-U	0.0255	0.0396	0.0327	0.0226	0.0519	0.0773
HyperSoRec	**0.0260**	**0.0398**	**0.0330**	**0.0325**	**0.0525**	**0.0775**

different, and therefore, HyperSoRec can further improve the model performance by taking into account the unequal weights of multi-aspect users' preferences with the attention network . Third, compared with the variant HyperSoRec-U, HyperSoRec consistently gains the best results. This evidence proves it is necessary to distinguish different influences of users on multiple aspects, which can help achieve better recommendations based on different semantic social connections. From all observations, we can reach out the agreement that HyperSoRec is an effective framework for social-aware recommendation, which learns the representative embeddings of users and items in the compact hyperbolic space, and jointly explores multi-aspect users' influences and multi-aspect item interactions.

5.3 Model Analysis

In this subsection, we discuss HyperSoRec from various perspectives including hyperbolic embedding visualization, parameter sensitivity and multi-aspect attention illustration.

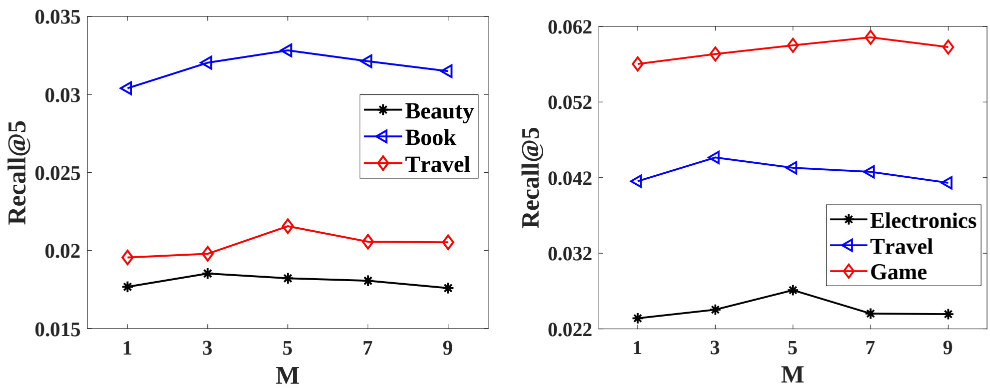


(a) HyperSoRec (user) (b) HyperSoRec (item)

(c) HyperSoRec(E) (user) (d) HyperSoRec(E) (item)

Fig. 6. Hyperbolic Embedding Visualization: Top two figures illustrate the user and item representations in hyperbolic space by HyperSoRec respectively. Bottom two figures are the user and item representations in Euclidean space by HyperSoRec(E) respectively.

5.3.1 Hyperbolic Embedding Visualization. Here, we intuitively demonstrate the representation ability of HyperSoRec capturing the latent hierarchical property in the data. Specifically, we randomly sample 500 users and items on "Beauty" task in Ciao, and equivalently map their hyperbolic embeddings to the Poincaré disk for visualization [42, 50], which can help us to observe the relative relationship between embeddings more intuitively in two-dimensional space. We also introduce the corresponding embedding results of the variant HyperSoRec(E) for comparison. Moreover, we mark three points A, B and C from high to low according to the users' degree or items' frequency, in order to better illustrate their correlation in the 2-D space. Fig. 6 shows all the embedding visualizations. First, compared with HyperSoRec(E), just a few points of HyperSoRec are distributed around the circle center while more points are distributed at the boundary. This demonstrates that learning user-item representations in hyperbolic space is more suitable to keep hierarchical data than Euclidean space. This means that a small number of users with more degrees (items with higher frequency) should be closer to the center of embeddings, and vice versa for users with less



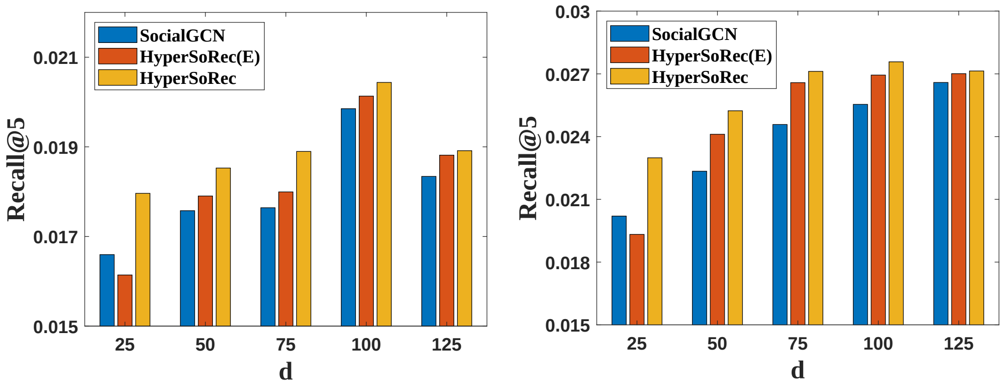
(a) Ciao (b) Epinions

Fig. 7. Result comparison of HyperSoRec with different aspect number M.

degrees (items with lower frequency). Second, we can observe the relative relationship among three points "A", "B" and "C" in HyperSoRec more specifically though Fig. 6(a) and Fig. 6(b). we observe that in HyperSoRec, users with more degrees (items with higher frequency) (i.e., "A") are closer to the center of circle than the ones with less degrees (lower frequency) (i.e., "C"). Such phenomenon indicates HyperSoRec could preserve the relative hierarchical relationship between users (items), even in the low-dimensional representation space. This indicates the learned hyperbolic embeddings in HyperSoRec has a parsimonious but stronger representation ability than the typical Euclidean embeddings by HyperSoRec(E), since it can effectively capture and preserve the latent hierarchy properties among users and items.

5.3.2 Parameter Sensitivity. We now investigate the effectiveness of three necessary model parameters: (1) the number of aspects M in users' influence and item interactions; (2) the embedding size d for user, item and aspect representations; (3) the sampled neighbor size $\mathcal{N}_i$ in each layer.

Sensitivity of aspect number M: As we mention in Section 1, users may connect with others and consume items both with respect to different underlying aspects. We first evaluate such claim. Specifically, in this part, we vary the aspect number in $\{1, 3, 5, 7, 9\}$ and present the results on all different tasks in Ciao and Epinions datasets in Fig. 7. From the figure, as the aspect number M increases, the performance of HyperSoRec firstly increases but decreases when its value surpasses 3, 5, 5 on "Beauty", "Travel" and "Book", in Ciao dataset respectively. The results on Epinions dataset perform the similar trends, where the best settings of HyperSoRec in "Electronics", "Travel" and "Game" are 5, 3, 7, respectively. This demonstrates that HyperSoRec with suitable setting of number M could effectively capture the utility of multi-aspect users' influences and multi-aspect item interactions. However, too large setting may also introduce some confusion of aspect learning, which reduces the performance. In addition, we also find an interesting observation. That is, if we ignore the aspect learning, i.e., aspect number $M = 1$, HyperSoRec just simply considers that there is only one comprehensive aspect existing user-user connections and user-item interactions, where such idea is similar to many previous work (Please see Table 3 for more details). However, with such setting, HyperSoRec cannot generate satisfied performance. This phenomenon proves the necessity of distinguishing the different influences and preferences on multiple aspects in social recommendation among users and items again.

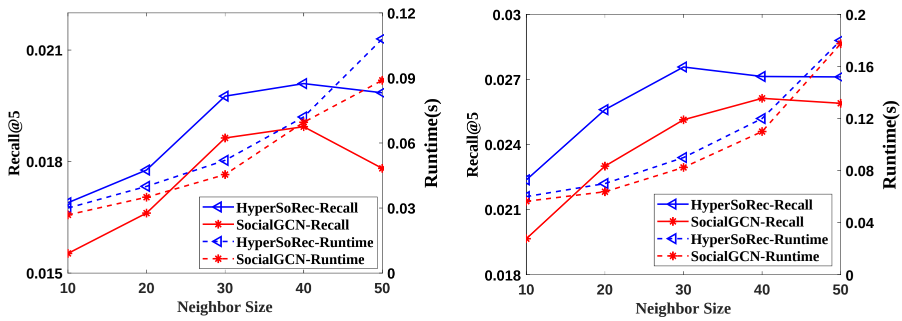


(a) Ciao (b) Epinions

Fig. 8. Result comparison of HyperSoRec with different embedding size d.

Sensitivity of embedding size d: The embedding size d plays an important role in the model, since it greatly affects the representation ability of learned embeddings. In this experiment, we select HyperSoRec, HyperSoRec(E) and one of the state-of-the-art model SocialGCN for comparison with the different size setting d={25, 50, 75, 100, 125}. For better illustration, we just report one task on each dataset (i.e., "Beauty" on Ciao and "Electronics" on Epinions) as the representatives since the result trends on other data tasks are similar after our experiments. The comparison results are illustrated in Fig. 8. There are several phenomena we can observe. First, HyperSoRec consistently performs the best in all settings, demonstrating that HyperSoRec, which is prone to capture the latent hierarchy property in hyperbolic space, has better representation ability and robustness than the Euclidean-based models (i.e., HyperSoRec, SocialGCN). Second, under the setting of embedding dimension with small value, i.e., d=25, HyperSoRec performs significantly better than others. This proves that HyperSoRec can gain better representation ability with low dimension but the Euclidean-based models do lose such ability. Thus, HyperSoRec can guarantee the compact but more representative ability of hyperbolic user and item embeddings for social recommendation. Third, the performance of both Euclidean-based models increase significantly when the embedding size increases, and gradually reduces the margin between HyperSoRec and them. This observation illustrates that Euclidean-based methods generally need more dimension setting than hyperbolic ones to learn the relative relationship between users and items for social-aware recommendation.

Sensitivity of sampled neighbor size $\mathcal{N}_i$: The sampled neighbor size $\mathcal{N}_i$ also has different impact on the efficiency and effectiveness of HyperSoRec. To further analyze the time efficiency from experimental perspective (the time complexity analyses can be found in Section 4.5.3), we conduct the experiments and introduce the baseline SocialGCN for comparison. Moreover, we report the representative ones (i.e., "Beauty" in Ciao and "Electronics" in Epinions) to illustrate the effect in a more intuitive way. In this experiment, we fix the number of sampled neighbors in each layer to the same, and vary the size in set {10, 20, 30, 40, 50}. Fig. 9 shows the performance and the corresponding runtime results of HyperSoRec and SocialGCN. From the figures, we can observe that as the number of sampled neighbors $\mathcal{N}_i$ increases, the margin between model performances gradually decrease, but the corresponding runtime of algorithm increases rapidly. Therefore, we select the number of sampled neighbor size as 20, in order to balance the performance and runtime in practical applications. Besides, although our HyperSoRec needs more running time than SocialGCN,



(a) Ciao (b) Epinions

Fig. 9. Result comparison of HyperSoRec with different sampled neighbor number $\mathcal{N}_i$.

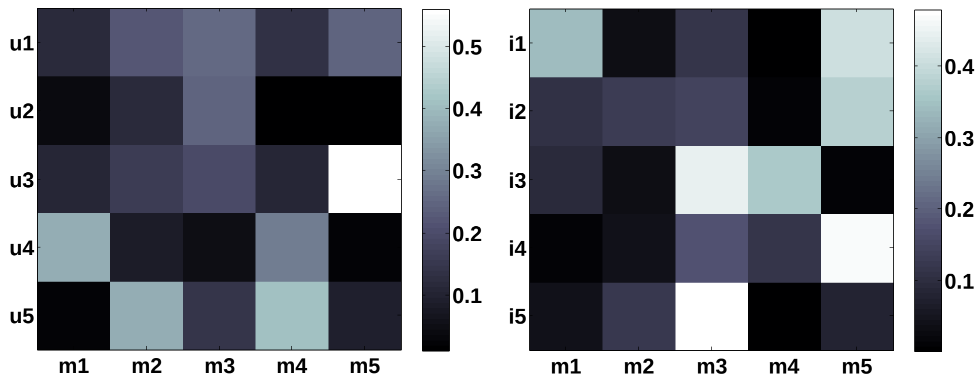
it can achieve better experimental performance. Moreover, as mentioned in Section 4.5.3, the complexities of HyperSoRec is acceptable in practice that is mainly determined by the number of graph layers and the sample size of neighbors. Therefore, we can make trade-off for HyperSoRec in real-world circumstances to have the most satisfied ability.

5.3.3 Aspect Attention Illustration. Our model HyperSoRec also endow a good ability of interpretability since it can analyze the multiple aspect influences between users and items in hyperbolic social recommendation. Fig. 10 provides a user study analysis by visualizing the attention scores in "Beauty" on Ciao dataset for illustration. Specifically, Fig. 10(a) shows how much she connects her 5 neighbors $(u_1, u_2, \cdots, u_5)$ in different aspects, i.e., Eq. (14). Fig. 10(b) presents how much she considers different aspects when she consumes 5 items $(i_1, i_2, \cdots, i_5)$, i.e., Eq. (17). For better illustration, we make some preprocessing as follows. First, we set the aspect number $m = 5$ without loss of generality. Second, we just select her 5 neighbors and 5 items since it is hard to illustrate clearly if we visualize her all neighbors and items in one figure. Moreover, we normalize both attention scores from Eq. (14) and Eq. (17).

From the figure, during the recommendation process, on one hand, we can observe that her five friends have different effects on her with different aspects. For example, user u_1 has dominant influence on aspect m_4, which means the user usually take advice from her friend u_1 on aspect m_4. On the other hand, the user also consumes different items considering different aspects, e.g., she likes the item i_5 due to the possible reason that she may be attracted by the aspect m_3. Generally, these observations can explain the results when we recommend an item to a target user in the recommender system, which demonstrate the effectiveness of multi-aspect learning in HyperSoRec.

6 CONCLUSIONS

In this paper, we presented a novel problem of hyperbolic social recommendation. Specifically, we proposed a novel hyperbolic graph neural network framework with multi-aspect learning (HyperSoRec). In this framework, we provided a principled way to learn compact but strong representations for users and items in hyperbolic space to preserve their inherent hierarchical properties. Then we respectively proposed a graph neural network with novel message-passing-receiving mechanism and an adaptive hyperbolic metric learning method to capture both multi-aspect user influences and item interactions. Extensive experiments demonstrated not only the better significant performance, but also the effectiveness and robustness for recommendation.



(a) User influence attention

(b) Item interaction attention

Fig. 10. Aspect influence visualization by attention weight of a case user on "Beauty" in Ciao dataset. Left figure shows how her five "friends" influences her on five aspects when she makes decisions. Right figure demonstrates that how she considers five aspects when consuming each item.

In this work, we focused on the implicit aspects beteen users' connections and interactions. Although HyperSoRec can enhance the experimental performance and demonstrate the necessity of modeling the relationship of users among multiple aspects, it cannot illustrate the specific meaning of each aspect. In the future, we are willing to incorporate more auxiliary information (e.g., knowledge graph, textual comments) to specify each aspect factors in HyperSoRec to improve the interpretability of model. Besides, we will further explore more complex user modeling in hyperbolic space, such as structure learning [23, 53, 85] and preference tracking [22, 41]. It will also be the future potential direction to investigate the curvature of hyperbolic embedding space [52] and effective optimiation algorithms [70] for practical applications.

ACKNOWLEDGEMENT

This research was partially supported by grants from the National Key Research and Development Program of China (No. 2016YFB1000904), the National Natural Science Foundation of China (No. U20A20229, 61976198 and 62022077). Hanghang Tong is partially supported by NSF (1947135, 2003924 and 1939725). Hao Wang gratefully acknowledges the support of the China Scholarship Council (No. 201906340183).

REFERENCES

[1] Forest Agostinelli, Matthew Hoffman, Peter Sadowski, and Pierre Baldi. 2014. Learning activation functions to improve deep neural networks. *arXiv preprint arXiv:1412.6830* (2014).

[2] Gary Bécigneul and Octavian-Eugen Ganea. 2018. Riemannian adaptive optimization methods. *arXiv preprint arXiv:1810.00760* (2018).

[3] Léon Bottou. 2010. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*. Springer, 177–186.

[4] James W Cannon, William J Floyd, Richard Kenyon, Walter R Parry, et al. 1997. Hyperbolic geometry. *Flavors of geometry* 31 (1997), 59–115.

[5] Benjamin Paul Chamberlain, Stephen R Hardwick, David R Wardrope, Fabon Dzogang, Fabio Daolio, and Saúl Vargas. 2019. Scalable hyperbolic recommender systems. *arXiv preprint arXiv:1902.08648* (2019).

[6] Ines Chami, Zhitao Ying, Christopher Ré, and Jure Leskovec. 2019. Hyperbolic graph convolutional neural networks. In *Advances in Neural Information Processing Systems*. 4869–4880.

[7] Chong Chen, Min Zhang, Yiqun Liu, and Shaoping Ma. 2019. Social attentional memory network: Modeling aspect-and friend-level differences in recommendation. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*. 177–185.

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 248–255.

[9] Jingtao Ding, Guanghui Yu, Yong Li, Xiangnan He, and Depeng Jin. 2020. Improving Implicit Recommender Systems with Auxiliary Data. *ACM Transactions on Information Systems (TOIS)* (2020).

[10] Yuxiao Dong, Nitesh V Chawla, Jie Tang, Yang Yang, and Yang Yang. 2017. User modeling on demographic attributes in big mobile social networks. *ACM Transactions on Information Systems (TOIS)* 35, 4 (2017), 1–33.

[11] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. 2019. Graph neural networks for social recommendation. In *The World Wide Web Conference*. 417–426.

[12] Shanshan Feng, Lucas Vinh Tran, Gao Cong, Lisi Chen, Jing Li, and Fan Li. 2020. HME: A Hyperbolic Metric Embedding Approach for Next-POI Recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1429–1438.

[13] Octavian Ganea, Gary Bécigneul, and Thomas Hofmann. 2018. Hyperbolic neural networks. In *Advances in neural information processing systems*. 5345–5355.

[14] Yong Ge, Hui Xiong, Alexander Tuzhilin, and Qi Liu. 2014. Cost-aware collaborative filtering for travel tour recommendations. *ACM Transactions on Information Systems (TOIS)* 32, 1 (2014), 1–31.

[15] Caglar Gulcehre, Misha Denil, Mateusz Malinowski, Ali Razavi, Razvan Pascanu, Karl Moritz Hermann, Peter Battaglia, Victor Bapst, David Raposo, Adam Santoro, et al. 2018. Hyperbolic attention networks. *arXiv preprint arXiv:1805.09786* (2018).

[16] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: a factorization-machine based neural network for CTR prediction. *arXiv preprint arXiv:1703.04247* (2017).

[17] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Advances in neural information processing systems*. 1024–1034.

[18] Xiangnan He and Tat-Seng Chua. 2017. Neural factorization machines for sparse predictive analytics. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*. 355–364.

[19] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. 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*. 639–648.

[20] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*. 173–182.

[21] Zhenya Huang, Qi Liu, Yuying Chen, Le Wu, Keli Xiao, Enhong Chen, Haiping Ma, and Guoping Hu. 2020. Learning or Forgetting? A Dynamic Approach for Tracking the Knowledge Proficiency of Students. *ACM Transactions on Information Systems (TOIS)* (2020).

[22] Zhenya Huang, Qi Liu, Yuying Chen, Le Wu, Keli Xiao, Enhong Chen, Haiping Ma, and Guoping Hu. 2020. Learning or forgetting? A dynamic approach for tracking the knowledge proficiency of students. *ACM Transactions on Information Systems (TOIS)* 38, 2 (2020), 1–33.

[23] Zhenya Huang, Qi Liu, Weibo Gao, Jinze Wu, Yu Yin, Hao Wang, and Enhong Chen. 2020. Neural mathematical solver with enhanced formula structure. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1729–1732.

[24] Mohsen Jamali and Martin Ester. 2010. A matrix factorization technique with trust propagation for recommendation in social networks. In *Proceedings of the fourth ACM conference on Recommender systems*. 135–142.

[25] Meng Jiang, Peng Cui, Fei Wang, Wenwu Zhu, and Shiqiang Yang. 2014. Scalable recommendation with social contextual information. *IEEE Transactions on Knowledge and Data Engineering* 26, 11 (2014), 2789–2802.

[26] Binbin Jin, Defu Lian, Zheng Liu, Qi Liu, Jianhui Ma, Xing Xie, and Enhong Chen. 2020. Sampling-Decomposable Generative Adversarial Recommender. *Advances in Neural Information Processing Systems* 33 (2020).

[27] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[28] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[29] Walid Krichene and Steffen Rendle. 2020. On sampled metrics for item recommendation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1748–1757.

[30] Marc Law, Renjie Liao, Jake Snell, and Richard Zemel. 2019. Lorentzian Distance Learning for Hyperbolic Representations. In *International Conference on Machine Learning*. 3672–3681.

[31] Jundong Li, Ruocheng Guo, Chenghao Liu, and Huan Liu. 2019. Adaptive unsupervised feature selection on attributed networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.

92–100.
[32] Defu Lian, Qi Liu, and Enhong Chen. 2020. Personalized Ranking with Importance Sampling. In *WWW '20: The Web Conference 2020*.
[33] Defu Lian, Qi Liu, and Enhong Chen. 2020. Personalized ranking with importance sampling. In *Proceedings of The Web Conference 2020*. 1093–1103.
[34] Defu Lian, Yongji Wu, Yong Ge, Xing Xie, and Enhong Chen. 2020. Geography-aware sequential location recommendation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2009–2019.
[35] Defu Lian, Cong Zhao, Xing Xie, Guangzhong Sun, Enhong Chen, and Yong Rui. 2014. GeoMF: joint geographical modeling and matrix factorization for point-of-interest recommendation. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 831–840.
[36] Defu Lian, Kai Zheng, Yong Ge, Longbing Cao, and Xing Xie. 2018. GeoMF++: Scalable Location Recommendation via Joint Geographical Modeling and Matrix Factorization. *Acm Transactions on Information Systems* 36, 3 (2018), 1–29.
[37] Jianxun Lian, Xiaohuan Zhou, Fuzheng Zhang, Zhongxia Chen, Xing Xie, and Guangzhong Sun. 2018. xdeepfm: Combining explicit and implicit feature interactions for recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1754–1763.
[38] Tong Lin and Hongbin Zha. 2008. Riemannian manifold learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30, 5 (2008), 796–809.
[39] Xiao Lin, Min Zhang, Yiqun Liu, and Shaoping Ma. 2019. Enhancing Personalized Recommendation by Implicit Preference Communities Modeling. *ACM Transactions on Information Systems (TOIS)* 37, 4 (2019), 1–32.
[40] Qi Liu, Enhong Chen, Hui Xiong, Yong Ge, Zhongmou Li, and Xiang Wu. 2012. A cocktail approach for travel package recommendation. *IEEE Transactions on Knowledge and Data Engineering* 26, 2 (2012), 278–293.
[41] Qi Liu, Zhenya Huang, Yu Yin, Enhong Chen, Hui Xiong, Yu Su, and Guoping Hu. 2019. Ekt: Exercise-aware knowledge tracing for student performance prediction. *IEEE Transactions on Knowledge and Data Engineering* 33, 1 (2019), 100–115.
[42] Qi Liu, Maximilian Nickel, and Douwe Kiela. 2019. Hyperbolic graph neural networks. In *Advances in Neural Information Processing Systems*. 8228–8239.
[43] Yuanxing Liu, Zhaochun Ren, Wei-Nan Zhang, Wanxiang Che, Ting Liu, and Dawei Yin. 2020. Keywords Generation Improves E-Commerce Session-based Recommendation. In *Proceedings of The Web Conference 2020*. 1604–1614.
[44] Hao Ma, Haixuan Yang, Michael R Lyu, and Irwin King. 2008. Sorec: social recommendation using probabilistic matrix factorization. In *Proceedings of the 17th ACM conference on Information and knowledge management*. 931–940.
[45] Brian J. Mcgill, Brian J. Enquist, Evan Weiher, and Mark Westoby. 2006. Rebuilding community ecology from functional traits. (2006).
[46] Miller McPherson, Lynn Smith-Lovin, and James M Cook. 2001. Birds of a feather: Homophily in social networks. *Annual review of sociology* 27, 1 (2001), 415–444.
[47] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.
[48] Leyla Mirvakhabova, Evgeny Frolov, Valentin Khrulkov, Ivan Oseledets, and Alexander Tuzhilin. 2020. Performance of Hyperbolic Geometry Models on Top-N Recommendation Tasks. In *Fourteenth ACM Conference on Recommender Systems*. 527–532.
[49] Andriy Mnih and Russ R Salakhutdinov. 2008. Probabilistic matrix factorization. In *Advances in neural information processing systems*. 1257–1264.
[50] Maximillian Nickel and Douwe Kiela. 2017. Poincaré embeddings for learning hierarchical representations. In *Advances in neural information processing systems*. 6338–6347.
[51] Maximillian Nickel and Douwe Kiela. 2018. Learning Continuous Hierarchies in the Lorentz Model of Hyperbolic Geometry. In *International Conference on Machine Learning*. 3779–3788.
[52] Hongbin Pei, Bingzhe Wei, Kevin Chang, Chunxu Zhang, and Bo Yang. 2020. Curvature Regularization to Prevent Distortion in Graph Embedding. In *Advances in Neural Information Processing Systems*, Vol. 33. 20779–20790.
[53] Hongbin Pei, Bo Yang, Jiming Liu, and Kevin Chang. 2020. Active Surveillance via Group Sparse Bayesian Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020). https://doi.org/10.1109/TPAMI.2020.3023092
[54] Liu Qi, Ge Yong, Zhongmou Li, Enhong Chen, and Xiong Hui. 2011. Personalized Travel Package Recommendation. In *IEEE International Conference on Data Mining*.
[55] Zhaochun Ren, Shangsong Liang, Piji Li, Shuaiqiang Wang, and Maarten de Rijke. 2017. Social collaborative viewpoint regression with explainable recommendations. In *Proceedings of the tenth ACM international conference on web search and data mining*. 485–494.
[56] Steffen Rendle. 2010. Factorization machines. In *2010 IEEE International Conference on Data Mining*. IEEE, 995–1000.
[57] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618* (2012).

[58] William F Reynolds. 1993. Hyperbolic geometry on a hyperboloid. *The American mathematical monthly* 100, 5 (1993), 442–455.

[59] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*. 285–295.

[60] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2008. The graph neural network model. *IEEE Transactions on Neural Networks* 20, 1 (2008), 61–80.

[61] Timothy Schmeier, Joeseph Chisari, Sam Garrett, and Brett Vintch. 2019. Music recommendations in hyperbolic space: an application of empirical bayes and hierarchical poincaré embeddings. In *Proceedings of the 13th ACM Conference on Recommender Systems*. 437–441.

[62] Peijie Sun, Le Wu, and Meng Wang. 2018. Attentive recurrent social recommendation. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 185–194.

[63] Yi Tay, Luu Anh Tuan, and Siu Cheung Hui. 2018. Latent relational metric learning via memory-based attention for collaborative ranking. In *Proceedings of the 2018 World Wide Web Conference*. 729–739.

[64] Tran Dang Quang Vinh, Yi Tay, Shuai Zhang, Gao Cong, and Xiao-Li Li. 2018. Hyperbolic recommender systems. *arXiv preprint arXiv:1809.01703* (2018).

[65] Lucas Vinh Tran, Yi Tay, Shuai Zhang, Gao Cong, and Xiaoli Li. 2020. HyperML: A Boosting Metric Learning Approach in Hyperbolic Space for Recommender Systems. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 609–617.

[66] Hao Wang, Enhong Chen, Qi Liu, Tong Xu, Dongfang Du, Wen Su, and Xiaopeng Zhang. 2018. A united approach to learning sparse attributed network embedding. In *2018 IEEE International Conference on Data Mining (ICDM)*. IEEE, 557–566.

[67] Hao Wang, Tong Xu, Qi Liu, Defu Lian, Enhong Chen, Dongfang Du, Han Wu, and Wen Su. 2019. MCNE: An end-to-end framework for learning multiple conditional network representations of social network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1064–1072.

[68] Xiang Wang, Xiangnan He, Liqiang Nie, and Tat-Seng Chua. 2017. Item silk road: Recommending items from information domains to social users. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*. 185–194.

[69] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat Seng Chua. 2019. Neural Graph Collaborative Filtering. In *the 42nd International ACM SIGIR Conference*.

[70] Xiao Wang, Yiding Zhang, and Chuan Shi. 2019. Hyperbolic heterogeneous information network embedding. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 5337–5344.

[71] Le Wu, Junwei Li, Peijie Sun, Richang Hong, Yong Ge, and Meng Wang. 2020. DiffNet++: A Neural Influence and Interest Diffusion Network for Social Recommendation. *IEEE Transactions on Knowledge and Data Engineering* (2020).

[72] Le Wu, Peijie Sun, Richang Hong, Yanjie Fu, Xiting Wang, and Meng Wang. 2018. Socialgcn: An efficient graph convolutional network based model for social recommendation. *arXiv preprint arXiv:1811.02815* (2018).

[73] Yichao Wu and Yufeng Liu. 2007. Robust truncated hinge loss support vector machines. *J. Amer. Statist. Assoc.* 102, 479 (2007), 974–983.

[74] Jun Xiao, Hao Ye, Xiangnan He, Hanwang Zhang, Fei Wu, and Tat-Seng Chua. 2017. Attentional factorization machines: Learning the weight of feature interactions via attention networks. *arXiv preprint arXiv:1708.04617* (2017).

[75] Fengli Xu, Jianxun Lian, Zhenyu Han, Yong Li, Yujian Xu, and Xing Xie. 2019. Relation-aware graph convolutional networks for agent-initiated social e-commerce recommendation. In *Proceedings of the 28th ACM international conference on information and knowledge management*. 529–538.

[76] Feng Xue, Xiangnan He, Xiang Wang, Jiandong Xu, Kai Liu, and Richang Hong. 2019. Deep Item-based Collaborative Filtering for Top-N Recommendation. *ACM Transactions on Information Systems* 37, 3 (2019), 1–25.

[77] Bo Yang, Yu Lei, Jiming Liu, and Wenjie Li. 2016. Social collaborative filtering by trust. *IEEE transactions on pattern analysis and machine intelligence* 39, 8 (2016), 1633–1647.

[78] Cheng Yang, Maosong Sun, Wayne Xin Zhao, Zhiyuan Liu, and Edward Y Chang. 2017. A neural network approach to jointly modeling social networks and mobile trajectories. *ACM Transactions on Information Systems (TOIS)* 35, 4 (2017), 1–28.

[79] Yuan Yao, Wayne Xin Zhao, Yaojing Wang, Hanghang Tong, Feng Xu, and Jian Lu. 2017. Version-aware rating prediction for mobile app recommendation. *ACM Transactions on Information Systems (TOIS)* 35, 4 (2017), 1–33.

[80] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. 2018. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 974–983.

[81] Junliang Yu, Hongzhi Yin, Jundong Li, Min Gao, Zi Huang, and Lizhen Cui. 2020. Enhancing Social Recommendation with Adversarial Graph Convolutional Networks. *arXiv e-prints* (2020), arXiv–2004.

[82] Yiding Zhang, Xiao Wang, Xunqiang Jiang, Chuan Shi, and Yanfang Ye. 2019. Hyperbolic graph attention network. *arXiv preprint arXiv:1912.03046* (2019).

[83] Tong Zhao, Julian McAuley, and Irwin King. 2014. Leveraging social connections to improve personalized ranking for collaborative filtering. In *Proceedings of the 23rd ACM international conference on information and knowledge management*. 261–270.

[84] Wayne Xin Zhao, Wenhui Zhang, Yulan He, Xing Xie, and Ji-Rong Wen. 2018. Automatically learning topics and difficulty levels of problems in online judge systems. *ACM Transactions on Information Systems (TOIS)* 36, 3 (2018), 1–33.

[85] Dawei Zhou, Si Zhang, Mehmet Yigit Yildirim, Scott Alcorn, Hanghang Tong, Hasan Davulcu, and Jingrui He. 2021. High-Order Structure Exploration on Massive Graphs: A Local Graph Clustering Perspective. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 15, 2 (2021), 1–26.