Hyperbolic Neural Collaborative Recommender

Anchen Li, Bo Yang, Huan Huo, Hongxu Chen, Guandong Xu, *Member, IEEE*, and Zhen Wang

Abstract—Recently, deep learning techniques have yielded immense success on recommender systems. However, one weakness of most deep methods is that, users/items mutual semantic relationships, which are latent in the user-item interactions, are not distilled out explicitly. Moreover, most methods have been primarily focused on representation learning in euclidean geometry. Since recent studies have shown that the bipartite graph structure has the non-euclidean latent anatomy, euclidean embeddings may suffer from a certain degree of distortion. In this work, we present **H**yperbolic **N**eural **C**ollaborative **R**ecommender (HNCR), a deep hyperbolic representation learning method that exploits mutual semantic relationships among users/items for collaborative filtering tasks. HNCR first introduces a neighbor construction strategy to build user and item semantic neighborhoods. Then HNCR develops a framework based on deep learning and hyperbolic geometry to integrate constructed neighborhoods into recommendation. To evaluate our method, we conduct experiments on the four datasets. Experimental results show the superiority of HNCR compared with its euclidean counterpart and state-of-the-art recommendation baselines. The results also indicate that hyperbolic representations can reflect meaningful data insights.

Index Terms—Collaborative filtering, deep learning, hyperbolic geometry, representation learning, recommender system

1 Introduction

In recent years, recommender systems have become a cornerstone of many online services and applications (e.g., Amazon, Netflix, and Youtube) for providing users with personalized suggestions of products or contents. As dominant and effective techniques in recommender systems, collaborative filtering (CF) approaches, which assume behaviorally similar users may have similar preferences for items, exploit the user-item interactions to drive recommendations [18], [26], [27], [46].

The common paradigm for CF is to transform users and items into latent vectors (a.k.a. embeddings) and predict user preference based on learned embeddings. Early CF models (e.g., MF [22] and SVD [21]) employ the inner product between user embeddings and item embeddings to perform prediction. Nevertheless, such shallow representations and the simple inner product may lack expressiveness to learn user preferences. As deep learning techniques developed, some recommendation algorithms use neural networks for modeling interaction behaviors, which enhances the performance of shallow CF approaches [15], [16], [54].

Despite effectiveness, most deep CF recommenders lack the ability to explicitly distill user-user or item-item detailed semantic dependencies which are latent in the topology structure of user-item interactions to reveal user (or item) similarities. Some recent studies [8], [19] seek to construct neighbors through co-occurrence relations (i.e., co-engage between users or co-engaged between items), while we argue that it is insufficient for these methods to mine the high-order dependency among users (or items). Let us take a user-movie interaction scenario as an example. As shown in Fig. 1, we believe that there is a semantic dependency between non-co-occurrence users a and d because these two users have the same co-occurrence neighbors (i.e., users b and c) and interacted with co-occurrence items (i.e., movies B and C). Another line of work adopting graph convolutions provides a possibility to capture this high-order semantic dependency [7], [47]. However, since not all high-order neighbors are meaningful [28], relevant messages from long-range nodes may be mixed with some noise from proximal nodes in multi-layer graph convolution operations. To better characterize user preferences and item properties, recommender systems require a fine-grained relationship mining algorithm.

In addition, most deep learning approaches for CF primarily operate in euclidean space. From the perspective of the graph, user-item connections could be considered as a bipartite graph, which is a typical complex network [11]. The characteristics of the complex network have been widely studied, and it is known that they are closely related to hyperbolic geometry [23]. Moreover, real-world user-item interactions often follow the power-law distribution. Recent studies reveal that data with a power-law distribution could be more effectively embedded in hyperbolic spaces rather than euclidean spaces [34]. This is because euclidean space

- *Anchen Li and Bo Yang are with the College of Computer Science and Technology, Jilin University, Changchun, Jilin 130012, China. E-mail: liac20@mails.jlu.edu.cn, ybo@jlu.edu.cn.*
- *Huan Huo, Hongxu Chen, and Guandong Xu are with the University of Technology Sydney, Ultimo, NSW 2007, Australia. E-mail: {Huan.Huo, Hongxu.Chen, Guandong.Xu}@uts.edu.au.*
- *Zhen Wang is with the Center for OPTical IMagery Analysis and Learning (OPTIMAL) and School of Mechanical Engineering, Northwestern Polytechnical University, Xi'an, Shaanxi 710072, China. E-mail: zhenwang0@gmail.com.*

Manuscript received 3 June 2021; revised 4 September 2022; accepted 30 October 2022. Date of publication 11 November 2022; date of current version 8 August 2023.
This work was supported in part by the Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin Univeriity, in part by the National Key R&D Program of China under Grants 2021ZD0112501 and 2021ZD0112502, in part by the National Natural Science Foundation of China under Grants U22A2098, 62172185, and 61876069, and in part by the Jilin Province Natural Science Foundation under Grant 20200201036JC.
(Corresponding author: Bo Yang.)
Recommended for acceptance by P. Castells PhD.
Digital Object Identifier no. 10.1109/TKDE.2022.3221386

1041-4347 © 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.
Authorized licensed use limited to: JILIN UNIVERSITY. Downloaded on April 02,2024 at 07:18:22 UTC from IEEE Xplore. Restrictions apply.

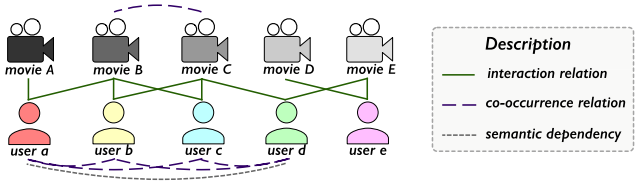


Fig. 1. A toy user-movie interaction scenario.

grows linearly while hyperbolic space grows exponentially [34]. These studies inspire us to explore whether deep CF approaches can benefit from hyperbolic geometry.

In this article, we consider both user-user and item-item semantic dependencies explicitly and utilize deep learning techniques and hyperbolic geometry to integrate these dependencies into recommendation. Indeed, it is a non-trivial task because of two key challenges. First, as mentioned, since users/items mutual semantic relationships are latent in the user-item connections, it is difficult to extract such relationships directly. How to detailed understand these correlations and build high-quality semantic neighbor sets is the first challenge. Second, because the mathematical operations (e.g., addition and multiplication) in hyperbolic geometry are different from those in euclidean geometry, most well-established deep euclidean recommenders cannot be directly utilized for hyperbolic space. Moreover, some operations in hyperbolic space do not satisfy basic mathematical properties, e.g., the addition operation is not commutative nor associative [10], which might affect the model efficiency. Thus, how to design our hyperbolic model elegantly and effectively is the second challenge.

To solve these technical challenges, we present a new approach **H**yperbolic **N**eural **C**ollaborative **R**ecommender (HNCR). There are two phases in our HNCR. In the first phase, HNCR devises a neighbor construction strategy to find user and item semantic neighborhoods. To be specific, HNCR first utilizes the implicit feedback to build a relational graph for users and items respectively. We design a delicate weighted edge approach to represent the relationship strength of nodes. Then each relational graph is projected in a latent space by node embedding approaches to construct user and item neighborhoods with semantic dependencies. With the help of node embedding techniques, complicated topology patterns in relational graphs can be presented as intuitive geometry in the latent space, which provides a fine-grained way to understand node relations. In the second phase, based on deep learning and hyperbolic geometry, HNCR carries a recommendation framework. To support the vector operations in hyperbolic space, we introduce Möbius gyrovector space operations [41], [42] in recommendation framework. Specifically, gyrovector operations are leveraged to integrate the semantic neighborhood as well as the interaction history into the hyperbolic embedding modeling. We further utilize the Möbius logarithmic map and exponential map to accelerate our method. In this way, our framework not only preserves the characteristics of hyperbolic space but also enhances learning efficiency. Based on the benchmark datasets, we conduct experiments to evaluate HNCR. The results show the superiority of HNCR compared with its euclidean counterpart and several strong CF approaches in common evaluation scenarios (i.e., click-through rate prediction tasks and top-K recommendation). Further analyses indicate that using hyperbolic geometry to learn representations can better organize the underlying hierarchical structure in the user-item interaction. We make the following contributions:

- We propose a novel method HNCR, which considers user-user and item-item semantic relations and makes use of hyperbolic geometry and deep learning techniques for the CF tasks.
- We propose a neighbor construction strategy based on implicit feedback to construct user and item semantic neighborhoods.
- We propose a deep hyperbolic framework, which uses gyrovector space operations to integrate the semantic neighbors and interaction history for recommendation.
- Extensive experiments on four datasets demonstrate the effectiveness and rationality of HNCR.

The rest of our work is structured as follows. The background is discussed in Section 2. Section 3 presents our method HNCR. In Section 4, we conduct experiments to show HNCR's effectiveness. We give a review of studies related to our HNCR in Section 5, followed by conclusions and future works in Section 6.

2 Preliminaries

This work investigates the notion of modeling user and item hyperbolic embeddings for recommendation. To better understand our approach, we cover the important background of the hyperbolic geometry and gyrovector space.

2.1 Hyperbolic Geometry

As a Riemannian manifold with negative curvature, hyperbolic space has received increasing momentum from the academic and industrial worlds. There are two important properties of hyperbolic space [34]: (i) compared with euclidean space, hyperbolic space grows faster. Specifically, euclidean space grows linearly, while hyperbolic space grows exponentially; and (ii) data of power-law distribution is well-suited to be embedded into hyperbolic space. To describe hyperbolic space, five frequently used hyperbolic models are designed [5]. These hyperbolic methods are connected and can be transformed into each other. A detailed introduction of these models can be referred to the previous literature [5].

Among these models, we use the Poincaré ball model for our approach HNCR, because gradient-based optimizers are suitable for optimizing this model [34]. Specifically, the n-dimensional Poincaré ball model is defined as follows:

$$\mathbb{D}_c^n = \left\{ (x_1, ..., x_n) : x_1^2 + \cdots + x_n^2 < \frac{1}{c} \right\}, \tag{1}$$

where c is the constant curvature. Specially, we can obtain an open unit ball in $\mathbb{R}^n$ when $c = 1$. The Riemannian metric tensor of the Poincaré ball is $g_{\mathbf{x}}^{\mathbb{D}} = (\lambda_{\mathbf{x}}^c)^2 g^{\mathbb{E}}$, in which $\lambda_{\mathbf{x}}^c = 2 \cdot (1 - c\|\mathbf{x}\|^2)^{-1}$; $\mathbf{x} \in \mathbb{D}^n$; $g^{\mathbb{E}} = \mathbf{I}$ is the euclidean metric.

With constant curvature -1, the distance between node embeddings $\mathbf{p} \in \mathbb{D}_c^n$ and $\mathbf{q} \in \mathbb{D}_c^n$ is:

$$d_{\mathbb{D}}(\mathbf{p}, \mathbf{q}) = \operatorname{arcosh}\left(1 + \frac{2\|\mathbf{p} - \mathbf{q}\|^2}{(1 - \|\mathbf{p}\|^2)(1 - \|\mathbf{q}\|^2)}\right), \tag{2}$$

Authorized licensed use limited to: JILIN UNIVERSITY. Downloaded on April 02,2024 at 07:18:22 UTC from IEEE Xplore. Restrictions apply.

where arcosh denotes the inverse hyperbolic cosine operation and $\|\cdot\|$ denotes the euclidean norm.

2.2 Gyrovector Spaces

For supporting vector operations in hyperbolic geometry, gyrovector space framework provides a feasible way [57]. To be specific, these gyrovector operations are built on a Poincaré ball $\mathbb{D}_c^n$. For $\mathbf{p} \in \mathbb{D}_c^n$, $\mathbf{q} \in \mathbb{D}_c^n$, $\alpha \in \mathbb{R}$, and $\mathbf{T} \in \mathbb{R}^{n' \times n}$, some mathematical operations are presented as:

- The *Möbius addition* of $\mathbf{p}$ and $\mathbf{q}$ is formulated as:

$$\mathbf{p} \oplus_c \mathbf{q} = \frac{(1 + 2c\langle \mathbf{p}, \mathbf{q} \rangle + c\|\mathbf{q}\|^2)\mathbf{p} + (1 - c\|\mathbf{p}\|^2)\mathbf{q}}{1 + 2c\langle \mathbf{p}, \mathbf{q} \rangle + c^2\|\mathbf{p}\|^2\|\mathbf{q}\|^2}. \tag{3}$$

  Here, we utilize $\langle \cdot, \cdot \rangle$ to denote the euclidean inner product. Generally, the Möbius addition does not satisfy associative and commutative laws.
- The *Möbius scalar multiplication* of α and $\mathbf{p}$ is formulated as:

$$\alpha \otimes_c \mathbf{p} = \frac{1}{\sqrt{c}} \tanh\big(\alpha \ \operatorname{artanh}\big(\sqrt{c}\|\mathbf{p}\|\big)\big) \frac{\mathbf{p}}{\|\mathbf{p}\|}, \tag{4}$$

  When $\mathbf{p} = \mathbf{0}$, $\alpha \otimes_c \mathbf{p} = \mathbf{0}$. The Möbius scalar multiplication satisfies associativity.
- The *Möbius vector multiplication* of $\mathbf{T}$ and $\mathbf{p}$ is formulated as:

$$\mathbf{T} \otimes_c \mathbf{p} = \frac{1}{\sqrt{c}} \tanh\left(\frac{\|\mathbf{T}\mathbf{p}\|}{\|\mathbf{p}\|} \operatorname{artanh}\big(\sqrt{c}\|\mathbf{p}\|\big)\right) \frac{\mathbf{T}\mathbf{p}}{\|\mathbf{T}\mathbf{p}\|}. \tag{5}$$

  The Möbius vector multiplication is associative.
- The *distance* between $\mathbf{p}$ and $\mathbf{q}$ is formulated as:

$$d_c(\mathbf{p}, \mathbf{q}) = \frac{2}{\sqrt{c}} \operatorname{artanh}(\sqrt{c}\|-\mathbf{p} \oplus_c \mathbf{q}\|). \tag{6}$$

  When $c = 1$, we retrieve the Equation (2).

In addition to the above mathematical operations, there are two mapping operations: Möbius exponential mapping and Möbius logarithmic mapping. The exponential mapping operation projects the nodes from tangent space $T_{\mathbf{x}}\mathbb{D}_c^n$, which is the first-order approximation of $\mathbb{D}_c^n$ around $\mathbf{x}$, to hyperbolic space $\mathbb{D}_c^n$. The logarithmic mapping operation projects the nodes from hyperbolic space $\mathbb{D}_c^n$ to tangent space $T_{\mathbf{x}}\mathbb{D}_c^n$.

Specifically, for $\mathbf{x} \in \mathbb{D}_c^n$, $\mathbf{y} \in \mathbb{D}_c^n \backslash \{\mathbf{0}\}$, and $\mathbf{z} \in T_{\mathbf{x}}\mathbb{D}_c^n \backslash \{\mathbf{0}\}$, the two mappings are defined as follows:

- The *Möbius exponential mapping* $\exp_{\mathbf{x}}^c : T_{\mathbf{x}}\mathbb{D}_c^n \to \mathbb{D}_c^n$ for $\mathbf{z}$ is formulated as follows:

$$\exp_{\mathbf{x}}^c(\mathbf{z}) = \mathbf{x} \oplus_c \left(\tanh\left(\sqrt{c}\frac{\lambda_{\mathbf{x}}^c\|\mathbf{z}\|}{2}\right)\frac{\mathbf{z}}{\sqrt{c}\|\mathbf{z}\|}\right). \tag{7}$$

- The *Möbius logarithmic mapping* $\log_{\mathbf{x}}^c : \mathbb{D}_c^n \to T_{\mathbf{x}}\mathbb{D}_c^n$ for $\mathbf{y}$ is formulated as follows:

$$\log_{\mathbf{x}}^c(\mathbf{y}) = \frac{2}{\sqrt{c}\lambda_{\mathbf{x}}^c} \operatorname{artanh}(\sqrt{c}\|-\mathbf{x} \oplus_c \mathbf{y}\|) \frac{-\mathbf{x} \oplus_c \mathbf{y}}{\|-\mathbf{x} \oplus_c \mathbf{y}\|}. \tag{8}$$

In this way, we can move the embeddings between tangent space and hyperbolic space with the help of mapping operations.

3 Methodology

We first give the notations and the problem formulation. Next, we present our HNCR, which contains two phases, i.e., neighborhood construction and recommendation framework. Finally, we present the complexity analysis of HNCR.

3.1 Notations and Problem Formulation

We focus on recommender systems with implicit feedback. Let $\mathcal{U} = \{u_1, u_2, ..., u_M\}$ and $\mathcal{V} = \{v_1, v_2, ..., v_N\}$ be the user set and item set, respectively, in which M and N represent the number of users and items, respectively. We utilize $\mathbf{Y} \in \mathbb{R}^{M \times N}$ to denote user-item interaction data, whose element $y_{ai} \in \{0, 1\}$ indicates whether user u_a has purchased or clicked on item v_i before.

Given the sets $\mathcal{U}$, $\mathcal{V}$, and training data from interaction matrix $\mathbf{Y}$, HNCR's first phase outputs user semantic neighbor set $\mathcal{N}_u = \{\mathcal{N}_u(1), \mathcal{N}_u(2), ..., \mathcal{N}_u(M)\}$ and item semantic neighbor set $\mathcal{N}_v = \{\mathcal{N}_v(1), \mathcal{N}_v(2), ..., \mathcal{N}_v(N)\}$, in which $\mathcal{N}_u(m)$ and $\mathcal{N}_v(n)$ are neighborhoods for u_m and v_n, respectively. Details of constructing $\mathcal{N}_u$ and $\mathcal{N}_v$ are introduced in Section 3.2. In the second phase of HNCR, given the constructed neighborhood data $\mathcal{N}_u$, $\mathcal{N}_v$ and training data from interaction matrix $\mathbf{Y}$, our recommendation framework aims to predict user u_a's preference for item v_i that he has never interacted with before. The details of the second phase of HNCR are described in Section 3.3.

3.2 Phase I: Neighborhood Construction

This subsection introduces our construction method based on implicit feedback. Such an approach consists of the following steps: we first construct the relational graphs of users and items, then project constructed relational graphs into latent spaces, and finally construct user and item semantic neighborhoods from the latent spaces. To avoid label leakage, we only utilize the training data for construction. Fig. 2 shows the three steps of neighborhood construction. In what follows, we introduce the process of the user side because the user's and item's processes are similar.

3.2.1 Relational Graph Construction

A user relational graph $G_u = (\mathcal{U}, \mathcal{E}_u)$ is constructed by transforming the user-item interactions for identifying user-user relations. If two users have interacted with the common items, they are connected in the graph. Moreover, relational graph G_u is constructed as a weighted graph. The edge $e^u \in \mathcal{E}_u$ is assigned a weight $w^u > 0$ to show the relational strength. We design a delicate approach to define the weight, which can represent user detailed relationships. Specifically, we define the weight w_{ab}^u for the edge e_{ab}^u between u_a and u_b as $w_{ab}^u = h_{ab}^u \cdot c_{ab}^u$, which is determined by two aspects, i.e., the historical behavior factor h_{ab}^u and the popularity of co-interacted items factor c_{ab}^u.

For the first aspect, if the historical behaviors of u_a and u_b are similar, the weight w_{ab}^u should be large, and vice versa. We use the cosine similarity to define h_{ab}^u, as follows:

Authorized licensed use limited to: JILIN UNIVERSITY. Downloaded on April 02,2024 at 07:18:22 UTC from IEEE Xplore. Restrictions apply.

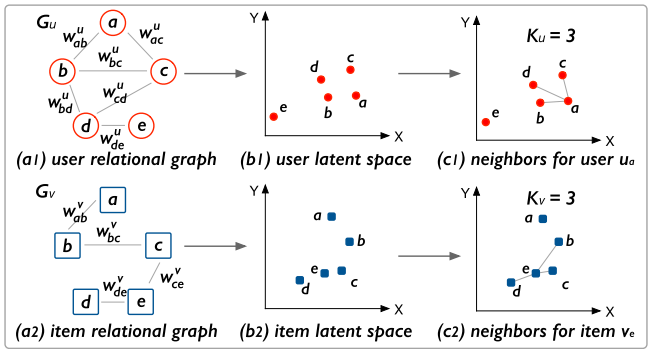


Fig. 2. The illustration of the three steps in neighbor construction. In (a1) and (a2), relational graphs are constructed. In (b1) and (b2), relational graphs are projected into the latent spaces. In (c1) and (c2), semantic neighborhoods are constructed.

$$h^u_{ab} = \frac{\mathbf{Y}_a \cdot \mathbf{Y}_b}{|\mathbf{Y}_a||\mathbf{Y}_b|}, \tag{9}$$

where $\mathbf{Y}_a$ and $\mathbf{Y}_b$ are the corresponding rows in interaction data $\mathbf{Y}$ for u_a and u_b, respectively.

For the second aspect, if the co-interacted items are unpopular, the weight w^u_{ab} should be large, and vice versa. This is because unpopular items can better reflect users' personalized preferences [14]. We define c^u_{ab}, as follows:

$$c^u_{ab} = \frac{2}{|C_{ab}|} \cdot \sum\nolimits_{v_i \in C_{ab}} \frac{1}{|I_v(i)|}, \tag{10}$$

where C_{ab} contains items that rated by both u_a and u_b (i.e., the intersection of items that two users have engaged with), and $I_v(i)$ contains users that rated v_i before.

Figs. 2(a1) and 2(a2) show the process of constructing user and item relational graphs according to Fig. 1. The benefit of this step is, both co-occurrence relationships (from one-hop neighborhood) and high-order semantic relationships (from multi-hop neighborhood) are reflected in the graphs.

3.2.2 Relational Graph Mapping

Node embedding techniques is capable of understanding graph structure data by revealing the latent space underlying it [35]. In this step, we leverage the node embedding approach to project the constructed relational graph into a low-dimensional space. Here, a mapping function $f_u : u \rightarrow z^u$ is utilized to project a user node $u \in \mathcal{U}$ in the user relational graph G_u into a latent representation vector $z^u \in \mathbb{R}^{l_u}$, where l_u is the vector dimension for the user latent space. According to the assumption of node embedding techniques [17], [33], the intuitive geometry in the latent space reflects complex topology patterns and characteristics of the user relational graph. A recent study reveals that a common node embedding model which preserves graph structures is capable of being effective [35]. Since node embedding techniques are not the main concern of our study, we use the embedding model LINE [39] which is suitable for weighted graphs.

Figs. 2(b1) and 2(b2) show the relational graph mapping processes. We use two-dimensional latent spaces for illustration. Though user a and user d are not in co-occurrence relationships, the distance between them in the low-dimensional space might be close because they are one-hop neighbors of user b and user c.

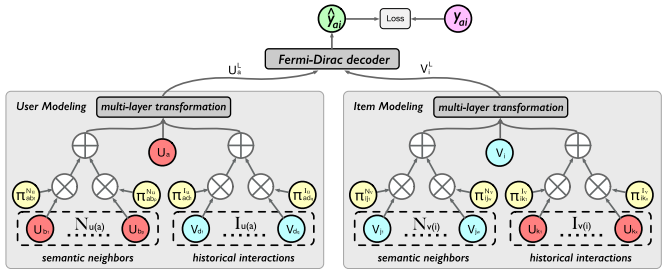


Fig. 3. Hyperbolic recommendation framework of HNCR.

3.2.3 Semantic Neighborhoods Construction

We build the semantic neighborhoods based on the latent space. With the help of network embedding techniques, for a target user, users with semantic relationships will appear near to her, while the users with irrelevant or noise information will appear far away from her. Therefore, we leverage the nearest neighbor approach to construct user neighborhoods $\mathcal{N}_u$. Specifically, user u_a's relational data $\mathcal{N}_u(a)$ contains K_u nearest nodes in the user latent space.

Figs. 2(c1) and 2(c2) show the processes of constructing the semantic neighborhoods for user u_a and item v_e. User u_a's neighborhoods is $\mathcal{N}_u(a) = \{u_b, u_c, u_d\}$ when $K_u = 3$, and item v_e's neighborhoods is $\mathcal{N}_v(e) = \{v_b, v_c, v_d\}$ when $K_v = 3$.

3.3 Phase II: Recommendation Framework

This subsection first introduces HNCR's recommendation framework and then presents the framework optimization method.

Fig. 3 shows the framework architecture, which is based on hyperbolic geometry. The framework contains the following components: an embedding layer, an aggregation layer, and a prediction layer. First, we embed entities in hyperbolic space by the embedding layer. Next, the aggregation layer refines the hyperbolic representations by considering the information of constructed semantic neighbors and historical behaviors. Finally, by taking the refined representations as inputs, the prediction layer learns user-item interactions and outputs predicted scores. Details of our framework are provided as follows.

3.3.1 Embedding Layer

Low-dimensional embedding vectors are utilized to encode users and items in the embedding layer. To be specific, such a layer takes one-hot representations of user u_a and item v_i as inputs and outputs their embeddings $\mathbf{u}_a$ and $\mathbf{v}_i$. We will learn these embedding representations in hyperbolic space $\mathbb{D}^d_c$.

3.3.2 Aggregation Layer

We design an aggregator that aggregates constructed semantic neighborhoods and historical behaviors in hyperbolic space for better modeling user and item embeddings. In what follows, since user and item modelings are symmetric, we mainly introduce the modeling process for users.

Given user u_a, we first aggregate the representations of her semantic neighborhoods into a single embedding. Möbius addition provides a way to compute the combination of

Authorized licensed use limited to: JILIN UNIVERSITY. Downloaded on April 02,2024 at 07:18:22 UTC from IEEE Xplore. Restrictions apply.

hyperbolic embeddings. Specifically, the aggregation is formulated as follows:

$$\mathbf{u}_a^N = \sum\nolimits_{u_b \in \mathcal{N}_u(a)}^{\oplus_c} \pi_{ab}^{\mathcal{N}_u} \otimes_c \mathbf{u}_b. \tag{11}$$

Here, we utilize $\sum^{\oplus_c}$ to denote the Möbius addition accumulation. We also design an attention mechanism to calculate π which indicates the importance of different neighbors. Specifically, factor $\pi_{ab}^{\mathcal{N}_u}$ is defined as:

$$\pi_{ab}^{\mathcal{N}_u} = \frac{\exp(-d_c(\mathbf{u}_a, \mathbf{u}_b)/\tau)}{\sum_{u_{b'} \in \mathcal{N}_u(a)} \exp(-d_c(\mathbf{u}_a, \mathbf{u}_{b'})/\tau)}, \tag{12}$$

where τ is the temperature parameter which is used for producing a softer distribution over neighbors. Since distance in hyperbolic space satisfies the triangle inequality, our attention mechanism preserves the transitivity among entities [10], [58].

For the second aggregation, it accounts for the user's historical behaviors. Specifically, we aggregate user's interacted items:

$$\mathbf{u}_a^I = \sum\nolimits_{v_d \in I_u(a)}^{\oplus_c} \pi_{ad}^{I_u} \otimes_c \mathbf{v}_d, \tag{13}$$

where $I_u(a)$ is the set of items that user u_a has engaged with before, and $\pi_{ad}^{I_u}$ is item d's interest influence score to user a, which can be defined as:

$$\pi_{ad}^{I_u} = \frac{\exp(-d_c(\mathbf{u}_a, \mathbf{v}_d)/\tau)}{\sum_{v_{d'} \in I_u(a)} \exp(-d_c(\mathbf{u}_a, \mathbf{v}_{d'})/\tau)}. \tag{14}$$

The last step in aggregation layer is to combine user embedding $\mathbf{u}_a$, her semantic neighbor embedding $\mathbf{u}_a^N$, and her historical preference representation $\mathbf{u}_a^I$ into a single representation. We design a multi-layer structure to obtain sufficient representation power, which is formulated as follows:

$$\mathbf{u}_a^{(L)} = \mathcal{M}_u^{(L)}(\mathcal{M}_u^{(L-1)}(\cdots \mathcal{M}_u^{(1)}(\mathbf{u}_a^{(0)}))), \tag{15}$$

$$\mathbf{u}_a^{(0)} = \mathbf{u}_a \oplus_c \mathbf{u}_a^N \oplus_c \mathbf{u}_a^I, \tag{16}$$

$$\mathcal{M}_u^{(l)}(\mathbf{u}_a^{(l-1)}) = \sigma\left(\mathbf{M}_u^{(l)} \otimes_c \mathbf{u}_a^{(l-1)}\right), \ l \in [1, L], \tag{17}$$

in which L denotes the layer number, $\mathbf{M}_u : \mathbb{R}^d \to \mathbb{R}^d$ denotes the linear map, and σ is the LeakyReLU function [32].

3.3.3 Prediction Layer

The Fermi-Dirac decoder [23], [34] is utilized to model interactions and calculate probability scores. To be specific, by taking the refined embeddings of user u_a and item v_i from the aggregation layer as inputs, the prediction layer is defined as:

$$\hat{y}_{ai} = \frac{1}{e^{(d_c(\mathbf{u}_a^{(L)}, \mathbf{v}_i^{(L)}) - r)/t} + 1}, \tag{18}$$

in which r and t are hyper-parameters.

3.3.4 Framework Optimization

This subsection first presents the acceleration strategy and then discusss the objective function of HNCR.

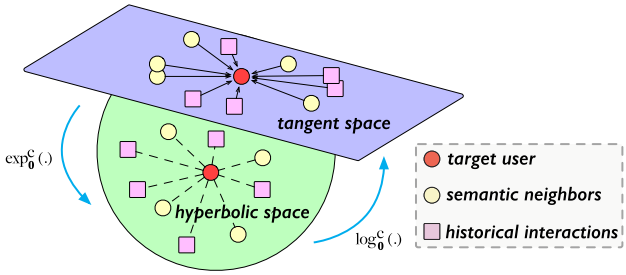


Fig. 4. Illustration of the acceleration strategy. Möbius logarithmic map $\log_{\mathbf{0}}^c$ and Möbius exponential map $\exp_{\mathbf{0}}^c$ are utilized to move the embeddings between tangent space and hyperbolic space.

Acceleration Strategy. In the aggregation layer, we utilize the Möbius addition to aggregate semantic neighbors and historical interactions for embedding learning. Since Möbius addition operation does not satisfy commutative and associative law [10], [58], the accumulation operation in Equations (11), (13), and (16) should be calculated by order (we omit the attention score π for simplicity), as follows:

$$\begin{aligned}\mathbf{u}_a^{(0)} = \mathbf{u}_a \oplus_c \left(\left(\left(\mathbf{u}_{b_1} \oplus_c \mathbf{u}_{b_2}\right) \oplus_c \mathbf{u}_{b_3}\right) \oplus_c \cdots\right) \\ \oplus_c \left(\left(\left(\mathbf{v}_{d_1} \oplus_c \mathbf{v}_{d_2}\right) \oplus_c \mathbf{v}_{d_3}\right) \oplus_c \cdots\right).\end{aligned} \tag{19}$$

As is known to all, there exist some active users and items that have many interactions in real recommendation scenarios. It is time-consuming to calculate Equation (13), which will reduce the efficiency of our HNCR. In addition, the Möbius addition accumulation is sensitive to the order of terms. Therefore, we propose to use an acceleration strategy to alleviate these problems.

Following the methods in [25], [58], we use Möbius logarithmic map and exponential map operations for the calculation. To be specific, we first leverage Möbius logarithmic map operation to project hyperbolic embeddings into tangent space and perform the embedding aggregation in this space. Then, we leverage Möbius exponential mapping for projecting the aggregated embeddings to hyperbolic space. Fig. 4 depicts this acceleration strategy. Take user u_a as an example, the process can be written as:

$$\begin{aligned}\mathbf{u}_a^{(0)} = \exp_{\mathbf{0}}^c\Bigg(\log_{\mathbf{0}}^c(\mathbf{u}_a) + \sum\nolimits_{u_b \in \mathcal{N}_u(a)} \pi_{ab}^{\mathcal{N}_u} \cdot \log_{\mathbf{0}}^c(\mathbf{u}_b) \\ + \sum\nolimits_{v_d \in I_u(a)} \pi_{ad}^{I_u} \cdot \log_{\mathbf{0}}^c(\mathbf{v}_d)\Bigg).\end{aligned} \tag{20}$$

Here, we utilize the origin as the target point in the aggregation process for HNCR, which ensures the mapping operation's simplicity and symmetry [10], [57]. To be specific, for $\mathbf{x} \in \mathbb{D}_c^d$ and $\mathbf{y} \in T_{\mathbf{x}}\mathbb{D}_c^d$, Möbius logarithmic map $\log_{\mathbf{0}}^c$ and Möbius exponential map $\exp_{\mathbf{0}}^c$ are formulated as:

$$\log_{\mathbf{0}}^c(\mathbf{x}) = \operatorname{artanh}(\sqrt{c}\|\mathbf{x}\|)\frac{\mathbf{x}}{\sqrt{c}\|\mathbf{x}\|}, \tag{21}$$

$$\exp_{\mathbf{0}}^c(\mathbf{y}) = \tanh(\sqrt{c}\|\mathbf{y}\|)\frac{\mathbf{y}}{\sqrt{c}\|\mathbf{y}\|}. \tag{22}$$

Since the addition operation satisfies commutative and associative law in the tangent space, we can perform the embedding accumulation in a parallel way in Equation (20). In this way, the new aggregation operation is more efficient and not sensitive to the order of neighbors. In practice, we

Authorized licensed use limited to: JILIN UNIVERSITY. Downloaded on April 02,2024 at 07:18:22 UTC from IEEE Xplore. Restrictions apply.

implemented the aggregation operation by Equations (19) and (20), respectively. Taking the dataset Ciao we used in the experiment as an example, the recommendation framework of HNCR cannot converge within several hours with Equation (19), while only need about twelve minutes to converge with Equation (20), conducted on a Linux server with Intel(R) Xeon(R) Gold CPU. Therefore, we replace Equation (19) with Equation (20) for aggregation.

Objective Function. We utilize the cross-entropy loss function to train HNCR's recommendation framework:

$$\min_{\Theta} \mathcal{L} = -\sum_{(u,v,\tilde{v})\in\mathcal{D}} (y_{uv}\log(\hat{y}_{uv}) + (1-y_{u\tilde{v}})\log(1-\hat{y}_{u\tilde{v}})), \tag{23}$$

in which Θ denotes framework parameters. $\mathcal{D}$ denotes training triplet set: $\mathcal{D} = \{(u, v, \tilde{v}) \mid u \in \mathcal{U} \wedge v \in I_u \wedge \tilde{v} \in \mathcal{V} \backslash I_u\}$.

Riemannian stochastic gradient descent (RSGD) [4] is utilized for optimizing our framework due to the Riemannian manifold structure of Poincaré Ball. As similar to [34], [44], the process of the parameter update can be computed as follows:

$$\theta_{t+1} = \mathfrak{R}_{\theta_t}(-\eta_t \nabla_R \mathcal{L}(\theta_t)), \tag{24}$$

Here, we use $\mathfrak{R}_{\theta_t}$ and η_t to denote the retraction on $\mathbb{D}$ at θ and the learning rate at time t, respectively. We use ∇_R to denote the Riemannian gradient and it could be calculated according to the euclidean gradient ∇_E, as follows:

$$\nabla_R = \frac{(1-\|\theta_t\|^2)^2}{4}\nabla_E. \tag{25}$$

3.4 Complexity Analysis

We present the model size and time complexity of the proposed HNCR in this subsection.

3.4.1 Model Size

We illustrate HNCR's model size from two aspects, i.e., neighborhood construction and the recommendation model. (i) For the neighborhood construction, we leverage the embedding approach to project the constructed users' and items' relational graphs to the latent spaces. Therefore, the parameters in the embedding approach includes the user's and item's latent factor matrices: $\{\mathbf{Z}_u, \mathbf{Z}_v\}$, where $\mathbf{Z}_u \in \mathbb{R}^{|\mathcal{U}|\times l_u}$ and $\mathbf{Z}_v \in \mathbb{R}^{|\mathcal{V}|\times l_v}$. In practice, l_u and l_v do not need to be too larger. Empirically, when $l_u = l_v = 4$ or 8, our method achieves nice performance. (ii) For the recommendation framework, the parameters includes two parts: user and item embeddings $\mathbf{E} = \{\{\mathbf{u}_i\}_{i=1}^{|\mathcal{U}|}, \{\mathbf{v}_i\}_{i=1}^{|\mathcal{V}|}\}$ and the weight matrices $\mathbf{W} = \{\mathbf{M}_u^{(l)}, \mathbf{M}_v^{(l)}, \forall l \in \{1, \cdots, L\}\}$. Empirically, a small L makes the framework achieve nice performance. Therefore, compared with embedding parameter $\mathbf{E}$, weight parameter $\mathbf{W}$ is lighter and can be neglected.

3.4.2 Time Complexity

HNCR's cost is from two aspects: neighborhood construction and recommendation model. (i) For the relational graph construction, the cost is $O(N \cdot C_u + M \cdot C_v)$, where C_u and C_v denote the average number of user and item co-occurrence neighbors. For the relational graph mapping, the time complexity depends on the selected embedding method. In the process of obtaining semantic neighborhoods, the complexity is $O(M \cdot l_u)$ and $O(N \cdot l_v)$ for a user and an item. In practice, acceleration approaches [3], [29] can be leveraged to speed up the semantic neighborhood construction. Note that the neighbor construction process can be done offline in advance. (ii) The neighbor aggregation is the main operation in the recommendation framework. For a user, the cost of the attention operation is $O(K_u \cdot d + H_u \cdot d)$, in which K_u is the number of the semantic neighborhood for users, H_u denotes the average number of user interactions, and d is the embedding dimension. The cost in our multi-layer structure is $O(L \cdot d^2)$, where L denotes the layer number. For the item side in our aggregator, the cost is $O(K_v \cdot d + H_v \cdot d + L \cdot d^2)$, where K_v denotes the number of semantic neighborhoods for items and H_v is the average number of item interactions. In general, the time consumption of the whole training epoch is $O(Y \cdot ((K_u + H_u + K_v + H_v) \cdot d + L \cdot d^2))$, where Y is the interaction number.

4 Experiments

We evaluate the recommendation quality of our method HNCR in this section. We first present experimental settings, next analyze experimental results, then explore the impacts of HNCR's component designs and hyper-parameters, and finally conduct case study experiments for detailed model analysis.

4.1 Experimental Setup

In what follows, benchmark datasets, evaluation metrics, comparison approaches, and hyper-parameters settings are presented.

4.1.1 Datasets

Four datasets are employed to evaluate methods, including Ciao,[1] Yelp,[2] Epinion,[3] and Douban.[4] Since the user-item interaction is in the rating format, we convert the rating information to 1 as positive feedback. Then, for each user, the same amount of negative instances as their positive feedback are sampled randomly from unrated items. We transform the negative instances into 0. Table 1 reports the dataset statistics.

4.1.2 Evaluation for Recommendation

Two recommendation scenarios are used here to study and evaluate our HNCR, i.e., click-through rate (CTR) prediction and top-K recommendation. (i) AUC (i.e., area under the curve) and ACC (i.e., accuracy) are employed as evaluation metrics for the CTR prediction task. These two metrics are usually employed in CF with implicit feedback. (ii) Since the recommendation task can be considered a rank problem, Precision@K and Recall@K are employed as evaluation metrics. Because the ranking task is time-consuming, we follow the approaches in [15], [16], [48] to reduce computing consumption. To be specific, for each user, we first sample 1000

1. Ciao: http://www.cse.msu.edu/~tangjili/index.html
2. Yelp: http://www.yelp.com/
3. Epinion: http://alchemy.cs.washington.edu/data/epinions/
4. Douban: http://book.douban.com

Authorized licensed use limited to: JILIN UNIVERSITY. Downloaded on April 02,2024 at 07:18:22 UTC from IEEE Xplore. Restrictions apply.

TABLE 1
Basic Statistics and Hyper-Parameter Settings for the Four Datasets

Dataset	# users	# items	# interactions	density	hyper-parameter settings
Ciao	7,267	11,211	147,995	0.181%	$d = 128$, $L = 1$, $K_u = K_v = 15$, $c = 1$, $\tau = 0.1$, $b = 1024$
Yelp	10,580	13,870	171,102	0.116%	$d = 128$, $L = 1$, $K_u = K_v = 15$, $c = 1$, $\tau = 0.1$, $b = 1024$
Epinion	20,608	23,585	454,022	0.093%	$d = 128$, $L = 1$, $K_u = K_v = 10$, $c = 1$, $\tau = 0.1$, $b = 1024$
Douban	12,748	22,347	785,272	0.275%	$d = 128$, $L = 1$, $K_u = K_v = 20$, $c = 1$, $\tau = 0.1$, $b = 1024$

items (negative samples) that have never engaged. Then the test items rank among negative samples.

4.1.3 Comparison Baselines

We compare HNCR with several groups of CF baselines, covering (i) MF-based models (i.e., SVD and WMF), (ii) deep learning models (i.e., NeuCF, CMN, and MMCF), (iii) GCN-based models (i.e., NGCF and LR-GCCF), and (iv) hyperbolic models (i.e., HyperBPR and HyperML). To examine the rationality of hyperbolic geometry in HNCR, we also prepare a euclidean counterpart ENCR. The characteristics of the baselines are given below:

- **SVD** is a classic MF approach that utilizes user history to enhance user embedding modeling [21].
- **WMF** is a classic MF approach that considers all missing data as negative instances and uniformly weights them [18].
- **NeuCF** is a deep CF approach that ensembles MF and a multi-layer perceptron [16].
- **CMN** is a memory-based approach that introduces memory networks in CF tasks. It designs memory slots for co-occurrence users for user embedding modeling [8].
- **MMCF** is a memory-based method that leverages multiplex memory networks to model the co-occurrence contexts [19].
- **NGCF** is a GCN-based approach that utilizes graph convolution operations to refine user and item representations [47].
- **LR-GCCF** is a GCN-based approach that simplifies graph convolution operations with the linear propagation layer [7].
- **HyperBPR** is a hyperbolic embedding method that learns user and item hyperbolic representations. It uses hyperbolic distance to learn user-item relationships and then designs a hyperbolic matching layer for studying user preferences for items [43].
- **HyperML** is a hyperbolic metric learning method that utilizes the operations of Möbius gyrovector spaces to devise a multi-task learning recommendation framework [44].
- **ENCR** is the euclidean counterpart of our HNCR. It takes the euclidean analogue of all the operations in HNCR. In addition, it removes two Möbius mapping operations (i.e., logarithmic mapping and exponential mapping).
- **HNCR** is the proposed approach.

It is noteworthy that MMCF, LR-GCCF, and HyperML are recently proposed strong recommendation algorithms.

4.1.4 Hyper-Parameter Settings

Each dataset is split into training, validation, and test sets according to the ratio 6:2:2. Adam optimizer is utilized to learn the parameters for all baselines. The learning rate η is searched in $\{0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05\}$ and the batch size b is searched in $\{256, 512, 1024, 2048\}$. We tested the embedding dimension d in $\{16, 32, 64, 128\}$ for all approaches. For our HNCR, the semantic neighbor size factor K_u, K_v is chosen from $\{5, 10, 15, 20, 25\}$ and the layer number L is selected in $\{1, 2, 3, 4\}$. In addition, we set curvature $c = 1$ and temperature $\tau = 0.1$. Table 1 shows our hyper-parameter settings. The effect of several key hyper-parameters will be studied in Section 4.6. In what follows, we report hyper-parameter settings of comparison methods. The layer sizes of the neural networks (e.g., the multi-layer perceptron in NeuCF, the memory network in CMN and MMCF, and the GCN layer in NGCF and LR-GCCF) are chosen in $\{1, 2, 3, 4\}$. For NGCF, the message dropout ratio is searched amongst $\{0.0, 0.1, \cdots 0.8\}$. For HyperML, the margin size is tuned amongst $\{0.1, 0.2, 0.5\}$ and the balancing factor in the multi-task learning is selected from $\{0.1, 0.2, 0.5, 1.0\}$. Some other hyper-parameters settings of baselines are either following their original papers or according to the empirical study.

4.2 Empirical Study

In this work, we design a hyperbolic recommendation framework. Researches discover that hyperbolic geometry is suitable for embedding the data with a power-law distribution [23], [34]. In what follows, we analyze the structural characteristic of user-item relationships and check whether the power-law distribution exists. We choose two real-world user-item interactions (i.e., Ciao and Epinion) for analysis and show the distribution of the user-item interaction number in Fig. 5. As shown in Figs. 5a and 5c, most users have very few interactions while a small portion of users has a huge number of interactions. Figs. 5b and 5d show that the interactions for items are similar to the users. We observe the interaction relations of users and items both follow the power-law distribution. Thus, hyperbolic spaces may be suitable for embedding the user-item bipartite graph in CF tasks.

4.3 Performance Comparison

We present and analyze the results of the CTR prediction (Table 2) and top-K recommendation (Fig. 6). Experimental results of AUC and ACC with varying embedding size d for the compared approaches are shown in Table 2. Because the model designs of WMF and HyperML are not appropriate for the CTR prediction tasks, we apply these

Authorized licensed use limited to: JILIN UNIVERSITY. Downloaded on April 02,2024 at 07:18:22 UTC from IEEE Xplore. Restrictions apply.

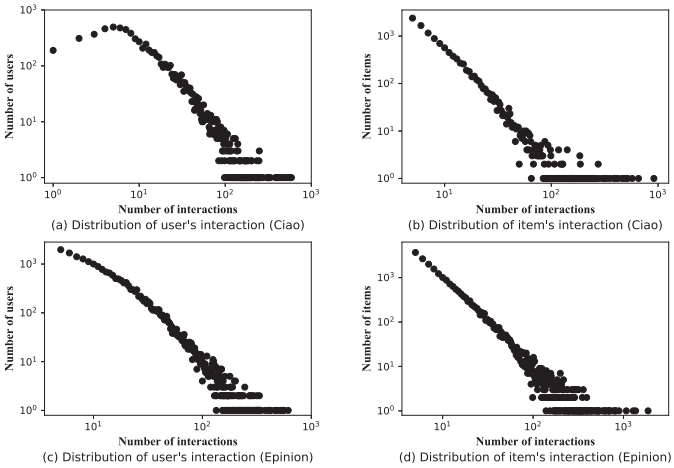


Fig. 5. Distributions of interactions on Ciao and Epinion datasets. The x-axis and y-axis show the interaction number associated with a user (or an item) and the number of users (or items), respectively.

two recommenders to the top-K recommendation. We have the following findings:

i) SVD performs poorly than other models on these four datasets, which is probably because the shallow representation lacks the expressiveness to model user-item relations. NeuCF performs better than SVD, which shows that nonlinear neural networks can enhance the recommendation performance. Nevertheless, both models consider user-item interactions as isolated instances and neglect relations in users and items.

ii) CMN and MMCF generally outperform SVD and NeuCF in most cases, indicating the effectiveness of considering co-occurrence relations in CF tasks. Besides, MMCF achieves better performance compared with CMN. This may be because MMCF accounts for both user and item co-occurrence relationships, while CMN only considers the user neighborhood.

iii) GCN-based models (i.e., NGCF and LR-GCCF) show strong performance among baselines. The results indicate that the neighborhood aggregation approach is efficient for the CF task, which is consistent with previous studies [7], [47].

iv) Among the shallow representation approaches, HyperBPR consistently outperforms SVD; meanwhile, our HNCR achieves better performance than the euclidean variant ENCR. These results indicate the advantages of hyperbolic geometry over euclidean geometry in recommender systems.

v) HNCR provides the best performance on different embedding size d among all the baselines in the CTR prediction task. For example, when $d = 64$, HNCR surpasses the best baselines by 3.23%, 3.58%, 2.30%, and 3.79% *w.r.t.* ACC in the four datasets.

For top-K recommendation, proper embedding sizes are set to ensure each approach achieves the best performance. Experimental results are shown in Fig. 6. Since CMN, NGCF, and ENCR underperform MMCF, LR-GCCF, and HNCR, respectively, they are not plotted for clarity. We have the observations:

i) The results of top-K recommendation are generally consistent with the CTR prediction. Deep learning and GCN-based approaches generally outperform MF-based approaches.

ii) Compared with the shallow euclidean representation approach (e.g., SVD), hyperbolic approaches (e.g., HyperML and HyperBPR) generally show better

TABLE 2
Comparisons of Methods With Different Embedding Size d for CTR Prediction Task on the Four Datasets

Models	Ciao								Yelp							
	AUC				ACC				AUC				ACC			
	d=16	d=32	d=64	d=128	d=16	d=32	d=64	d=128	d=16	d=32	d=64	d=128	d=16	d=32	d=64	d=128
SVD	0.7162	0.7206	0.7250	0.7218	0.6520	0.6556	0.6581	0.6546	0.8134	0.8156	0.8186	0.8198	0.7455	0.7469	0.7483	0.7522
NeuCF	0.7267	0.7334	0.7392	0.7371	0.6595	0.6638	0.6698	0.6684	0.8200	0.8237	0.8258	0.8255	0.7545	0.7563	0.7586	0.7579
CMN	0.7293	0.7355	0.7396	0.7365	0.6618	0.6650	0.6647	0.6643	0.8214	0.8245	0.8249	0.8262	0.7550	0.7584	0.7590	0.7596
MMCF	0.7518	0.7542	0.7569	0.7557	0.6761	0.6782	0.6829	0.6803	0.8215	0.8260	0.8277	0.8290	0.7576	0.7591	0.7625	0.7643
NGCF	0.7534	0.7585	0.7613	0.7659	0.6817	0.6861	0.6895	0.6926	0.8233	0.8268	0.8290	0.8297	0.7602	0.7630	0.7642	0.7634
LR-GCCF	0.7579	0.7621	0.7671	0.7694	0.6836	0.6895	0.6914	0.6952	0.8338	0.8343	0.8355	0.8374	0.7648	0.7672	0.7673	0.7691
HyperBPR	0.7556	0.7607	0.7658	0.7674	0.6859	0.6876	0.6900	0.6914	0.8226	0.8234	0.8262	0.8271	0.7604	0.7618	0.7627	0.7630
HNCR	**0.7851**	**0.7912**	**0.7985**	**0.8010**	**0.7083**	**0.7124**	**0.7137**	**0.7148**	**0.8545**	**0.8560**	**0.8591**	**0.8599**	**0.7929**	**0.7933**	**0.7948**	**0.7952**
ENCR	0.7643	0.7680	0.7719	0.7756	0.6903	0.6922	0.6930	0.6945	0.8347	0.8385	0.8399	0.8403	0.7721	0.7737	0.7740	0.7757

Models	Epinion								Douban							
	AUC				ACC				AUC				ACC			
	d=16	d=32	d=64	d=128	d=16	d=32	d=64	d=128	d=16	d=32	d=64	d=128	d=16	d=32	d=64	d=128
SVD	0.7938	0.8011	0.8102	0.8070	0.7231	0.7274	0.7324	0.7322	0.8225	0.8253	0.8279	0.8287	0.7488	0.7523	0.7546	0.7532
NeuCF	0.8074	0.8103	0.8132	0.8138	0.7316	0.7339	0.7352	0.7346	0.8302	0.8320	0.8365	0.8385	0.7561	0.7587	0.7618	0.7633
CMN	0.8080	0.8122	0.8138	0.8149	0.7363	0.7386	0.7380	0.7403	0.8350	0.8368	0.8390	0.8392	0.7587	0.7611	0.7620	0.7616
MMCF	0.8235	0.8276	0.8266	0.8257	0.7490	0.7514	0.7509	0.7500	0.8436	0.8455	0.8471	0.8488	0.7652	0.7668	0.7691	0.7718
NGCF	0.8143	0.8191	0.8224	0.8231	0.7388	0.7425	0.7440	0.7471	0.8411	0.8464	0.8483	0.8504	0.7645	0.7684	0.7697	0.7725
LR-GCCF	0.8266	0.8287	0.8319	0.8323	0.7492	0.7508	0.7535	0.7549	0.8448	0.8489	0.8512	0.8523	0.7686	0.7714	0.7722	0.7737
HyperBPR	0.8171	0.8190	0.8205	0.8212	0.7407	0.7430	0.7439	0.7447	0.8379	0.8415	0.8464	0.8470	0.7633	0.7662	0.7709	0.7704
HNCR	**0.8465**	**0.8492**	**0.8514**	**0.8529**	**0.7684**	**0.7697**	**0.7708**	**0.7712**	**0.8693**	**0.8754**	**0.8788**	**0.8810**	**0.7923**	**0.7981**	**0.8014**	**0.8024**
ENCR	0.8277	0.8294	0.8321	0.8330	0.7522	0.7533	0.7551	0.7558	0.8486	0.8502	0.8566	0.8581	0.7762	0.7789	0.7802	0.7818

Authorized licensed use limited to: JILIN UNIVERSITY. Downloaded on April 02,2024 at 07:18:22 UTC from IEEE Xplore. Restrictions apply.

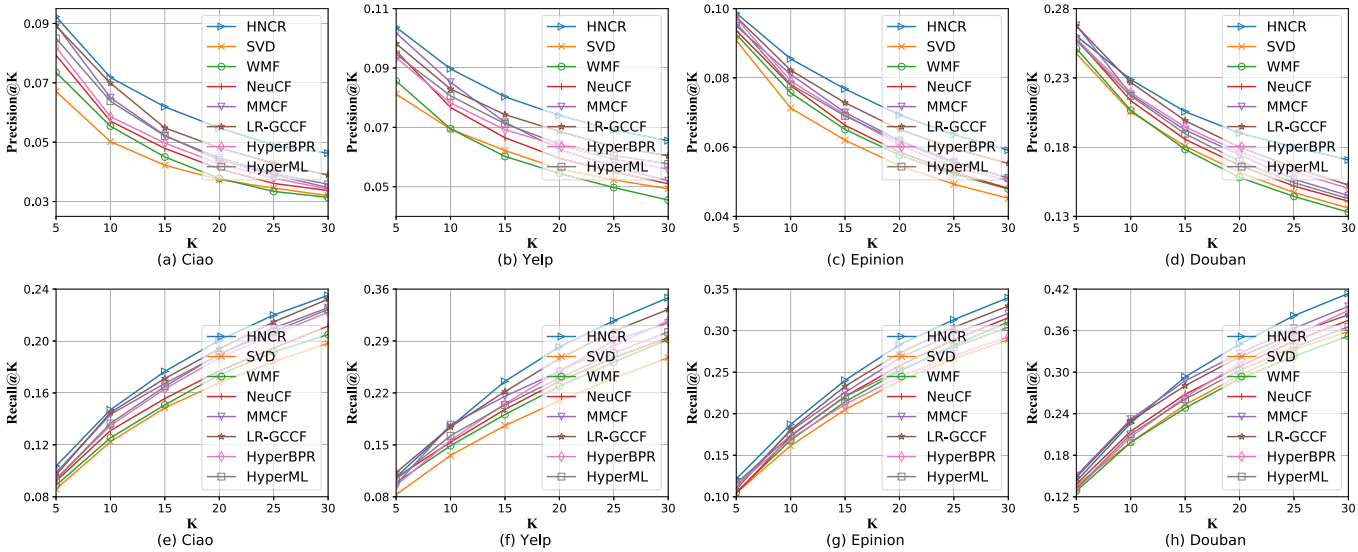


Fig. 6. Top-K recommendation results of Precision@K and Recall@K.

performance, which indicates leveraging hyperbolic geometry can learn high-quality representations, so as to enhance the performance.

iii) HNCR achieves strongly competitive performance on the four datasets in the top-K recommendation. For example, HNCR attains relative improvements of 3.46%, 6.24%, 3.70%, and 5.62% over the strongest baseline w.r.t. Recall@20 on four datasets.

4.4 Performance w.r.t Sparsity Degrees

The sparsity issue often limits performance. In what follows, we study the ability of comparison approaches to handle different sparsity scenarios. Specifically, users from the test set are divided into five groups according to the training interaction numbers. Fig. 7 illustrates the models' recommendation accuracy under different data degrees, where the x-axis indicates the interaction numbers for each user, and the y-axis represents the ACC results. From the figures, we find our HNCR provides better performance than the other strong approaches (e.g., MMCF and LR-GCCF), which indicates HNCR can maintain a decent performance under different data sparsity.

4.5 Ablation Study

Our method HNCR includes two phases: a neighborhood construction method and a deep hyperbolic representation learning model. In what follows, we study the rationality of these two designs.

4.5.1 *Effect of the Neighbor Construction*

To study the effect of our constructed semantic neighbors, we consider two operations for HNCR, ENCR, and MMCF as follows:

- **-C**: Such operation is for HNCR and ENCR, which replaces the semantic neighbors with co-occurrence neighbors.
- **-S**: Such operation is for MMCF, which replaces the co-occurrence neighbors with semantic neighbors.

Table 3 presents AUC results for different variants. HNCR, ENCR, and MMCF are shown as baselines. We draw the conclusions as follows: (i) HNCR-C and ENCR-C perform worse than HNCR and ENCR, respectively. MMCF-S achieves better performance than MMCF. These results verify semantic neighbors can provide more useful information than the co-occurrence neighborhood, so as to enhance the performance; and (ii) HNCR-C consistently outperforms ENCR-C, which indicates the effectiveness of hyperbolic geometry in the CF tasks.

4.5.2 *Effect of the aggregator*

The key part of the recommendation model is that we design an aggregator to refine user and item hyperbolic representations. In this subsection, we evaluate the aggregator

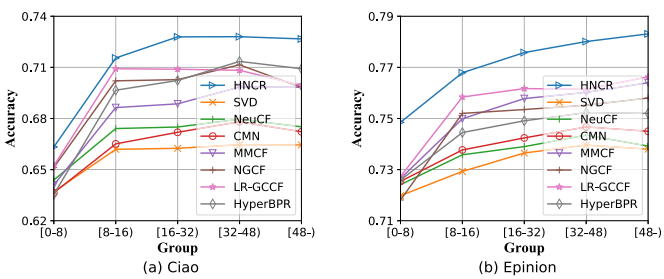


Fig. 7. Performance of our HNCR and other baseline approaches under different sparsity degrees.

TABLE 3
Effect of the Neighbor Construction Strategy

Models	Ciao	Yelp	Epinion	Douban
HNCR	0.8010	0.8599	0.8529	0.8810
HNCR-C	0.7937	0.8505	0.8464	0.8702
ENCR	0.7756	0.8403	0.8330	0.8581
ENCR-C	0.7695	0.8357	0.8242	0.8509
MMCF	0.7569	0.8290	0.8276	0.8488
MMCF-S	0.7628	0.8392	0.8302	0.8554

Authorized licensed use limited to: JILIN UNIVERSITY. Downloaded on April 02,2024 at 07:18:22 UTC from IEEE Xplore. Restrictions apply.

TABLE 4
Effect of Components in the Aggregator

Models	Ciao	Yelp	Epinion	Douban
HNCR	0.8010	0.8599	0.8529	0.8810
HNCR-N	0.7908	0.8481	0.8415	0.8624
HNCR-H	0.7872	0.8454	0.8437	0.8668
HNCR-A	0.7964	0.8513	0.8498	0.8762
ENCR	0.7756	0.8403	0.8330	0.8581
ENCR-N	0.7650	0.8325	0.8217	0.8395
ENCR-H	0.7625	0.8286	0.8203	0.8410
ENCR-A	0.7722	0.8360	0.8248	0.8473

by analyzing the contributions from different components. We consider three operations for the aggregator in HNCR and ENCR as follows:

- **-N**: Removing the semantic neighbor information aggregation.
- **-H**: Removing the historical behavior information aggregation.
- **-A**: Replacing the attention mechanism with the mean pooling.

The AUC results for all model variants are shown in Table 4. HNCR and ENCR are shown as baselines. We draw the conclusions as follows: (i) Removing any parts of the framework decreases the recommendation accuracy. For instance, the complete model HNCR performs better than the variants HNCR-N and HNCR-H, which indicates that both semantic neighbors and historical behaviors benefit the recommendation; (ii) HNCR and ENCR achieve better scores than HNCR-A and ENCR-A, respectively. This validates that considering the importance of different neighbors in the aggregation operation can improve the recommendation performance; and (iii) For different operations, hyperbolic framework HNCR outperforms euclidean framework ENCR, which shows the advantage of hyperbolic geometry over euclidean geometry in modeling user-item interactions.

4.6 Hyper-Parameter Study

In this subsection, two hyper-parameters are studied for our approaches, i.e., semantic neighbor number K_u, K_v and layer size L. Fig. 8 presents the AUC results for HNCR and ENCR. From the figures, we draw the conclusions as follows: (i) For neighbor size K_u, K_v, we find that the AUC results increase first and then start to decrease. It may be probably because a larger semantic neighbor number will be prone to be misled by noises. (ii) For layer size L, we observe that HNCR and ENCR achieve the best performance when $L = 1$ and $L = 2$, respectively. The results also show that leveraging hyperbolic geometry can achieve nice performance without using multi-layer network structures.

4.7 Case Study

In this subsection, we study whether our approaches can reflect the underlying data structure in CF scenarios. Specifically, we consider the hierarchical structure of frequencies hidden in power-law distribution from the data, i.e., the more frequently users or items appear in interactions, the higher the hierarchy of users or items. We check learned

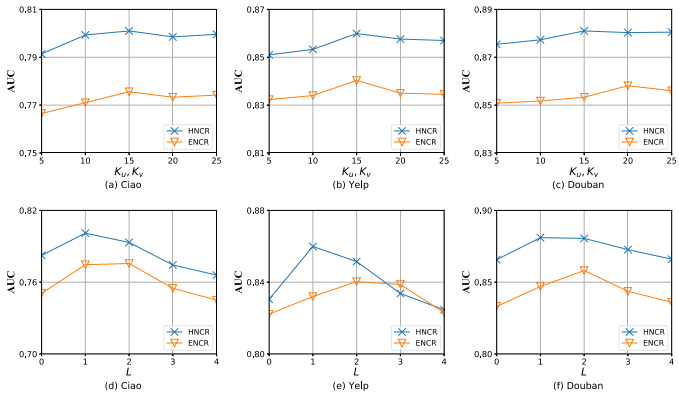


Fig. 8. Hyper-parameter study on the three datasets.

representations from HNCR and ENCR to compare the ability of modeling hierarchy between hyperbolic geometry and euclidean geometry. To be specific, according to the distance toward the origin (i.e., **0**), we divide nodes (i.e., users and items) from the user-item interaction graph into four groups. In this process, we try to ensure that each group's node number is similar. From the first group to the fourth group, the average distance between the nodes in the group and the origin increases gradually. To show the activity frequency of each node group, we calculate the average node interaction numbers in each group. Fig. 9 presents the results for HNCR and ENCR. From the first group to the fourth group, we find average interaction numbers decrease with the distance from the origin, which shows HNCR and ENCR can model the hierarchy of activity in the user-item interactions. Compared with ENCR, we can see that HNCR is capable of reflecting underlying data structure more clearly. These observations show hyperbolic geometry has a stronger ability than euclidean geometry to model the latent hierarchy.

To further show the hierarchical structure learned in the embeddings, we randomly select 200 users and 200 items from the Ciao, Yelp, and Douban datasets, and plot them in Fig. 10 where the x-axis and y-axis indicate the distance from the origin and the average distance from all other nodes in the dataset, respectively. The results show active nodes (near the origin) generally have small average distances and vice versa. Moreover, we find that users are closer to the origin than items, which is consistent with previous studies [44]. Compared with ENCR, we find that the distribution of nodes in HNCR is more regular, which indicates that utilizing hyperbolic geometry for modeling embeddings can better organize the latent data structure in the user-item interactions.

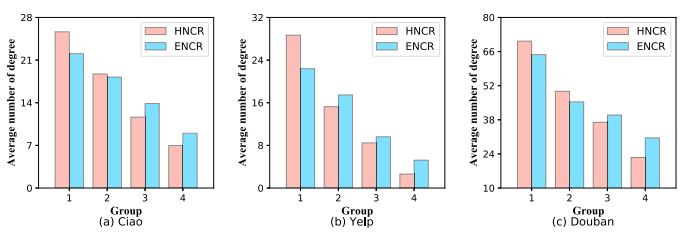
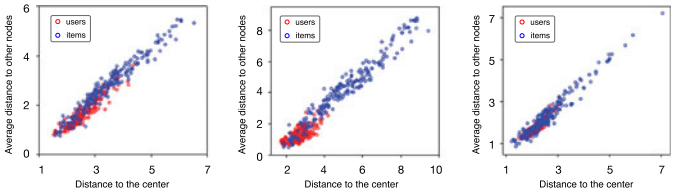
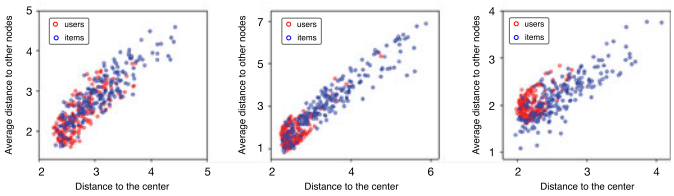


Fig. 9. Hierarchy analysis on the three datasets.

Authorized licensed use limited to: JILIN UNIVERSITY. Downloaded on April 02,2024 at 07:18:22 UTC from IEEE Xplore. Restrictions apply.



(a) Analysis of the embeddings for HNCR on Ciao, Yelp, and Douban datasets



(b) Analysis of the embeddings for ENCR on Ciao, Yelp, and Douban datasets

Fig. 10. Embeddings analysis on the three datasets.

5 Related Work

Three areas that are relevant to the study are reviewed in this section, i.e., deep collaborative filtering methods, relation-aware recommendation, and hyperbolic representation learning.

5.1 Deep Collaborative Filtering Methods

Collaborative filtering (CF) is one of the dominant recommendation strategies [18], [26], [27], [46]. Recently, neural networks have yielded immense success on CF tasks. Researchers use a variety of neural architectures to improve recommendation performance. Early works leverage restricted Boltzmann machines [37] and autoencoders [38] for CF with explicit feedback. In addition, some recent studies [15], [16] exploit multi-layer perceptron for elaborate representation learning and complex user-item interaction modeling. To consider the neighborhood information, memory networks have become a popular choice for CF [8], [19]. The idea of memory-based architectures is to use an external memory matrix and a controller to increase the model capacity. As graph learning techniques developed, researchers propose utilizing graph neural networks and their variants for capturing multi-hop relationships in the user-item bipartite graph [7], [14], [47]. Due to the powerful ability of convolutional and recurrent neural networks, some recommenders leverage these neural blocks to incorporate some textual data and image data in CF [2], [13]. Some hybrid CF methods integrate different deep learning techniques to build more powerful models [56]. Notably, most deep CF models operate in the euclidean space. In this paper, a deep hyperbolic framework is developed for pure CF systems. We show that the combination of hyperbolic geometry and deep learning techniques is a promising choice for modeling user-item interactions.

5.2 Relation-Aware Recommendation

There are many recommenders using various relations to enhance the performance. In some recommendation scenarios, researchers use the relations existing in the side information to design the model. Social connections have been widely exploited in recommender systems. Early models devise regularization terms in the MF framework to incorporate social relationships [31], [55]. Recently, deep learning approaches (e.g., multi-layer perceptron [50] and graph neural networks [48]) are utilized for social recommendation. Researchers also leverage knowledge graphs to understand user preferences. There are three typical methods (i.e., path-based approach, embedding-based approach, and graph-based approach) for knowledge-based recommendation [12], and each of them has been shown effective. In addition, user/item attributes [53], session sequences [52], and textual reviews [49] are also used to build relation-aware recommendation.

Another line of studies constructs relationships from the user-item interactions. Considering co-occurrence relationships is a common approach. For instance, CMN [8] and MMCF [19] adopt memory networks to model the co-occurrence contexts. Some studies further define similarity coefficients based on the co-occurrence relationships to build neighborhoods [9], [51]. For example, DICER defines a collaborative similarity metric for identifying user-user and item-item relations [9]. Despite their effectiveness, these co-occurrence relation-based methods may neglect user-user or item-item high-order semantic dependencies. Some work also adopts graph convolution techniques on the interaction graph to capture high-order connections, which provides an implicit way to model user-user and item-item semantic correlation. However, because not all the information from high-order neighbors is positively useful [28], [47], designing multi-layer convolution operations to capture some long-range semantics may introduce irrelevant messages [35]. Different from the above studies, our method leverages the network geometry to build neighborhoods, which explicitly considers semantic information between users/items. Furthermore, since our method selects neighbors according to distance in the latent space, it has the potential to mitigate the above problem in graph-based CF recommenders. Although some studies [24], [35] also propose constructing neighborhoods via the latent space, we believe that their methods cannot be directly applied to our settings. In our settings, the user-item bipartite graph is a typical heterogeneous structure. In addition, to reflect the detailed relationships between nodes, we need to design an edge-weighted method based on implicit feedback for the graph. However, these methods either focus on relation mining in homogeneous graphs or on building relationships using explicit feedback. In this paper, to consider semantic relations among users/items, we build relational graphs for users and items respectively. A delicate method is also designed to weight the edges in constructed relational graphs, which can represent the node detailed relationships.

5.3 Hyperbolic Representation Learning

Recently, hyperbolic geometry has shown great promise in machine learning. In knowledge graph completion, MuRP [1] and HyperKG [20] represent entities in the Poincaré ball for better modeling the hierarchy property of the knowledge graph. In computer vision, ST-GCN devises a neural architecture on a Riemann manifold for action recognition tasks and experimental results proved its superiority [36]. In natural language processing,

Authorized licensed use limited to: JILIN UNIVERSITY. Downloaded on April 02,2024 at 07:18:22 UTC from IEEE Xplore. Restrictions apply.

HyperQA uses Poincaré disk to design a parameter-efficient neural network for question-answer ranking and retrieval [40]. In graph representation learning, various hyperbolic graph neural networks (e.g., HGCN [6] and HAT [58]) are designed for link prediction, node clustering, and node classification.

Hyperbolic representation learning is also becoming a popular research field for recommendation. Some recommenders utilize hyperbolic geometry to model side information with the hierarchical structure [25], [30], [45]. For example, Hyper-Know maps the entities of the knowledge graph as well as users and items to the Poincaré ball model for knowledge-aware recommendation [30]. HyperSoRec [45] leverages the hyperbolic geometry to model social networks for social recommendation. For pure CF approaches in hyperbolic space, HyperBPR and HyperML are two representative works. HyperBPR models user and item embeddings in Poincaré ball and utilizes the Bayesian personalized ranking framework for recommendation [43]. HyperML is a metric learning approach, which utilizes gyrovector space operations for modeling user-item interactions [44]. However, these hyperbolic CF methods are different from our HNCR as they consider every user-item pair as an isolated instance and neglect semantic relations. Also, they are shallow representation learning models, which may lack the capability of modeling user and item features. Our approach HNCR explicitly considers semantic correlations among users/items and develops a deep framework based on hyperbolic geometry for recommendation. We believe that the insights in this work are enlightening for other future studies to explore the hyperbolic space in recommender systems.

6 Conclusion and Future Work

In this work, we introduce a novel solution for deep learning-based CF recommenders. We argue that most deep methods face at least one of the following issues. First, most methods lack techniques for distilling out detailed semantic relationships among users/items explicitly. Second, most euclidean CF methods may suffer from a certain degree of distortion when embedding user-item interaction data in euclidean geometry. To bridge the gap, we develop a new approach called HNCR. We first leverage the network embedding to build semantic neighborhoods. Then we devise a deep hyperbolic framework, which considers both constructed semantic neighborhoods and the interaction history for recommendation. The results show the rationality and effectiveness of HNCR. We believe our approach provides a promising approach to utilizing hyperbolic geometry for enhancing deep learning-based recommender systems. For future work, we will improve HNCR in three promising directions. First, we plan to integrate auxiliary information (e.g., social networks and user reviews) to improve performance. Second, some causal concepts (e.g., causal effect inference and counterfactual reasoning) will be considered, which can discover and amplify popularity bias. Third, we will explore the use of sequential models and hash algorithms to benefit the online real-time recommendation.

Acknowledgments

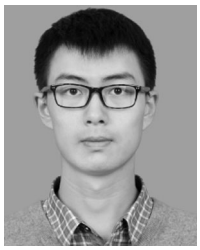
The authors thank all the reviewers for their helpful comments.

References

[1] I. Balazevic, C. Allen, and T. Hospedales, "Multi-relational poincaré graph embeddings," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2019, pp. 4465–4475.
[2] T. Bansal, D. Belanger, and A. McCallum, "Ask the GRU: Multi-task learning for deep text recommendations," in *Proc. 10th ACM Conf. Recommender Syst.*, 2016, pp. 107–114.
[3] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Commun. ACM*, vol. 18, no. 9, pp. 509–517, 1975.
[4] S. Bonnabel, "Stochastic gradient descent on riemannian manifolds," *IEEE Trans. Autom. Control*, vol. 58, no. 9, pp. 2217–2229, Sep. 2013.
[5] J. W. Cannon et al., "Hyperbolic geometry," *Flavors Geometry*, vol. 31, no. 59-115, 1997, Art. no. 2.
[6] I. Chami, Z. Ying, C. Ré, and J. Leskovec, "Hyperbolic graph convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 4869–4880.
[7] L. Chen, L. Wu, R. Hong, K. Zhang, and M. Wang, "Revisiting graph based collaborative filtering: A linear residual graph convolutional network approach," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 27–34.
[8] T. Ebesu, B. Shen, and Y. Fang, "Collaborative memory network for recommendation systems," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2018, pp. 515–524.
[9] B. Fu, W. Zhang, G. Hu, X. Dai, S. Huang, and J. Chen, "Dual side deep context-aware modulation for social recommendation," in *Proc. Web Conf.*, 2021, pp. 2524–2534.
[10] O. Ganea, G. Bécigneul, and T. Hofmann, "Hyperbolic neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 5350–5360.
[11] J.-L. Guillaume and M. Latapy, "Bipartite graphs as models of complex networks," *Physica A: Stat. Mechanics Appl.*, vol. 371, no. 2, pp. 795–813, 2006.
[12] Q. Guo et al., "A survey on knowledge graph-based recommender systems," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 8, pp. 3549–3568, Aug. 2022.
[13] R. He and J. McAuley, "Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering," in *Proc. Web Conf.*, 2016, pp. 507–517.
[14] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "LightGCN: Simplifying and powering graph convolution network for recommendation," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2020, pp. 639–648.
[15] X. He, Z. He, J. Song, Z. Liu, Y.-G. Jiang, and T.-S. Chua, "NAIS: Neural attentive item similarity model for recommendation," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 12, pp. 2354–2366, Dec. 2018.
[16] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proc. Web Conf.*, 2017, pp. 173–182.
[17] P. D. Hoff, A. E. Raftery, and M. S. Handcock, "Latent space approaches to social network analysis," *J. Amer. Statist. Assoc.*, vol. 97, no. 460, pp. 1090–1098, 2002.
[18] Y. Hu, Y. Koren, and C. Volinsky, "Collaborative filtering for implicit feedback datasets," in *Proc. IEEE 8th Int. Conf. Data Mining*, 2008, pp. 263–272.
[19] X. Jiang, B. Hu, Y. Fang, and C. Shi, "Multiplex memory network for collaborative filtering," in *Proc. SIAM Int. Conf. Data Mining*, 2020, pp. 91–99.
[20] P. Kolyvakis, A. Kalousis, and D. Kiritsis, "Hyperkg: Hyperbolic knowledge graph embeddings for knowledge base completion," 2019, *arXiv:1908.04895*.
[21] Y. Koren, "Factorization meets the neighborhood: A multifaceted collaborative filtering model," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2008, pp. 426–434.
[22] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.
[23] D. Krioukov, F. Papadopoulos, M. Kitsak, A. Vahdat, and M. Boguná, "Hyperbolic geometry of complex networks," *Phys. Rev. E*, vol. 82, no. 3, 2010, Art. no. 036106.

Authorized licensed use limited to: JILIN UNIVERSITY. Downloaded on April 02,2024 at 07:18:22 UTC from IEEE Xplore. Restrictions apply.

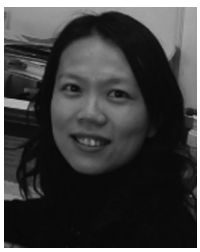
[24] A. Li, B. Yang, H. Huo, and F. Hussain, "Leveraging implicit relations for recommender systems," *Inf. Sci.*, vol. 579, pp. 55–71, 2021.
[25] A. Li, B. Yang, F. K. Hussain, and H. Huo, "HSR: Hyperbolic social recommender," *Inf. Sci.*, vol. 585, pp. 275–288, 2022.
[26] D. Lian et al., "Scalable content-aware collaborative filtering for location recommendation," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 6, pp. 1122–1135, Jun. 2018.
[27] D. Lian, X. Xie, and E. Chen, "Discrete matrix factorization and extension for fast item recommendation," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 5, pp. 1919–1933, May 2021.
[28] F. Liu, Z. Cheng, L. Zhu, Z. Gao, and L. Nie, "Interest-aware message-passing GCn for recommendation," in *Proc. Web Conf.*, 2021, pp. 1296–1305.
[29] T. Liu, A. W. Moore, A. Gray, and C. Cardie, "New algorithms for efficient high-dimensional nonparametric classification," *J. Mach. Learn. Res.*, vol. 7, no. 6, pp. 1135–1158, 2006.
[30] C. Ma, L. Ma, Y. Zhang, H. Wu, X. Liu, and M. Coates, "Knowledge-enhanced top-k recommendation in poincaré ball," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 4285–4293.
[31] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King, "Recommender systems with social regularization," in *Proc. ACM Int. Conf. Web Search Data Mining*, 2011, pp. 287–296.
[32] A. L. Maas et al., "Rectifier nonlinearities improve neural network acoustic models," *Proc. Int. Conf. Mach. Learn.*, vol. 30, no. 1, pp. 3–9, 2013.
[33] A. Muscoloni, J. M. Thomas, S. Ciucci, G. Bianconi, and C. V. Cannistraci, "Machine learning meets complex networks via coalescent embedding in the hyperbolic space," *Nature Commun.*, vol. 8, no. 1, pp. 1–19, 2017.
[34] M. Nickel and D. Kiela, "Poincaré embeddings for learning hierarchical representations," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017.
[35] H. Pei, B. Wei, K. C.-C. Chang, Y. Lei, and B. Yang, "Geom-GCN: Geometric graph convolutional networks," in *Proc. Int. Conf. Learn. Representations*, 2020, pp. 1–12.
[36] W. Peng, J. Shi, Z. Xia, and G. Zhao, "Mix dimension in poincaré geometry for 3D skeleton-based action recognition," in *Proc. ACM Int. Conf. Multimedia*, 2020, pp. 1432–1440.
[37] R. Salakhutdinov, A. Mnih, and G. Hinton, "Restricted boltzmann machines for collaborative filtering," in *Proc. Int. Conf. Mach. Learn.*, 2007, pp. 791–798.
[38] S. Sedhain, A. K. Menon, S. Sanner, and L. Xie, "Autorec: Autoencoders meet collaborative filtering," in *Proc. Web Conf.*, 2015, pp. 111–112.
[39] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line: Large-scale information network embedding," in *Proc. Web Conf.*, 2015, pp. 1067–1077.
[40] Y. Tay, L. A. Tuan, and S. C. Hui, "Hyperbolic representation learning for fast and efficient neural question answering," in *Proc. ACM Int. Conf. Web Search Data Mining*, 2018, pp. 583–591.
[41] A. A. Ungar, "Hyperbolic trigonometry and its application in the poincaré ball model of hyperbolic geometry," *Comput. Math. Appl.*, vol. 41, no. 1-2, pp. 135–147, 2001.
[42] A. A. Ungar, "A gyrovector space approach to hyperbolic geometry," *Synth. Lectures Math. Statist.*, vol. 1, no. 1, pp. 1–194, 2008.
[43] T. D. Q. Vinh, Y. Tay, S. Zhang, G. Cong, and X.-L. Li, "Hyperbolic recommender systems," 2018, *arXiv:1809.01703*.
[44] L. Vinh Tran, Y. Tay, S. Zhang, G. Cong, and X. Li, "HyperML: A boosting metric learning approach in hyperbolic space for recommender systems," in *Proc. ACM Int. Conf. Web Search Data Mining*, 2020, pp. 609–617.
[45] H. Wang, D. Lian, H. Tong, Q. Liu, Z. Huang, and E. Chen, "HyperSoRec: Exploiting hyperbolic user and item representations with multiple aspects for social-aware recommendation," *ACM Trans. Inf. Syst.*, vol. 40, no. 2, pp. 1–28, 2021.
[46] S. Wang, J. Tang, Y. Wang, and H. Liu, "Exploring hierarchical structures for recommender systems," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 6, pp. 1022–1035, Jun. 2018.
[47] X. Wang, X. He, M. Wang, F. Feng, and T.-S. Chua, "Neural graph collaborative filtering," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2019, pp. 165–174.
[48] L. Wu, J. Li, P. Sun, R. Hong, Y. Ge, and M. Wang, "DiffNet++: A neural influence and interest diffusion network for social recommendation," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 10, pp. 4753–4766, Oct. 2022.
[49] L. Wu, C. Quan, C. Li, and D. Ji, "PARL: Let strangers speak out what you like," in *Proc. ACM Int. Conf. Inf. Knowl. Manage.*, 2018, pp. 677–686.
[50] L. Wu, P. Sun, R. Hong, Y. Ge, and M. Wang, "Collaborative neural social recommendation," *IEEE Trans. Syst., Man, Cybern.: Syst.*, vol. 51, no. 1, pp. 464–476, Jan. 2021.
[51] Q. Wu et al., "Dual graph attention networks for deep latent representation of multifaceted social effects in recommender systems," in *Proc. Web Conf.*, 2019, pp. 2091–2102.
[52] S. Wu, Y. Tang, Y. Zhu, L. Wang, X. Xie, and T. Tan, "Session-based recommendation with graph neural networks," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 346–353.
[53] X. Xin, X. He, Y. Zhang, Y. Zhang, and J. Jose, "Relational collaborative filtering: Modeling multiple item relations for recommendation," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2019, pp. 125–134.
[54] H.-J. Xue, X. Dai, J. Zhang, S. Huang, and J. Chen, "Deep matrix factorization models for recommender systems," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, 2017, pp. 3203–3209.
[55] C. Zhang, L. Yu, Y. Wang, C. Shah, and X. Zhang, "Collaborative user network embedding for social recommender systems," in *Proc. SIAM Int. Conf. Data Mining*, 2017, pp. 381–389.
[56] F. Zhang, N. J. Yuan, D. Lian, X. Xie, and W.-Y. Ma, "Collaborative knowledge base embedding for recommender systems," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2016, pp. 353–362.
[57] S. Zhang, H. Chen, X. Ming, L. Cui, H. Yin, and G. Xu, "Where are we in embedding spaces?," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2021, pp. 2223–2231.
[58] Y. Zhang, X. Wang, C. Shi, X. Jiang, and Y. F. Ye, "Hyperbolic graph attention network," *IEEE Trans. Big Data*, vol. 8, no. 6, pp. 1690–1701, Dec. 2022.



Anchen Li received the MS and BS degrees from the Shool of Computer Science and Technology, Jilin University, in 2020 and 2018, respectively. He is currently working toward the PhD degree with the College of Computer Science and Technology, Jilin University. His main research interests include recommender systems, collaborative filtering/ranking, data mining, machine learning and social network analysis.

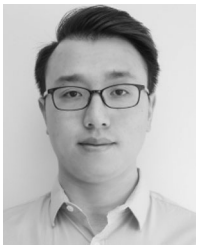


Bo Yang is currently the director of the Key Laboratory of Symbolic Computation and Knowledge Engineering of the Ministry of Education, Jilin University, Changchun, China. His research interests include data mining, machine learning, knowledge engineering, and complex/social network modeling and analysis. He has published more than 120 articles on international journals, including *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *IEEE Transactions on Knowledge and Data Engineering*, *IEEE Transactions on Neural Networks and Learning Systems*, *IEEE Transactions on Cybernetics*, *ACM Transactions on Knowledge Discovery from Data*, *ACM Transactions on the Web*, DKE, JAAMAS, and KBS, and international conferences, including ICLR, NeurIPS, IJCAI, AAAI, WWW, ICDM, CVPR, and COLING.



Huan Huo received the BEng and PhD degrees in computer science and technology from Northeastern University, China, in 2002 and 2007, respectively. She taught with the Department of Computer Information System, University of the Fraser Valley in Canada, and did collaborative research with the University of Waterloo as a visiting scholar for one year. Since 2018, she has been a senior lecturer with the School of Computer Science, University of Technology Sydney, Australia. Her research interests include data stream management technology, advanced data analysis, and data-driven cybersecurity.

Authorized licensed use limited to: JILIN UNIVERSITY. Downloaded on April 02,2024 at 07:18:22 UTC from IEEE Xplore. Restrictions apply.



Hongxu Chen received the PhD degree in computer science from the University of Queensland, in 2020. He is a data scientist, now working as a postdoctoral research fellow with the School of Computer Science, University of Technology Sydney, Australia. His research interests mainly focus on data science in general and expend across multiple practical application scenarios, such as network science, data mining, recommendation systems and social network analytics. Hongxu has published many peer-reviewed papers in top-tier high-quality international conferences and journals, such as SIGKDD, ICDE, ICDM, AAAI, IJCAI, *IEEE Transactions on Knowledge and Data Engineering*. He also serves as program committee member and reviewers in multiple international conference, such as CIKM, ICDM, KDD, SIGIR, AAAI, and he also acts as invited reviewer for multiple journals in his research fields, including *Transactions on Knowledge and Data Engineering (TKDE)*, *WWW Journal*, *IEEE Transactions on Systems, Man and Cybernetics: Systems*, *Journal of Complexity*, *ACM Transactions on Data Science*.



Guandong Xu (Member, IEEE) is currently a full professor with the School of Computer Science, University of Technology Sydney, Ultimo, NSW, Australia. He has published 3 monographs in Springer and CRC press and more than 250 journal articles and conference papers in data science, recommender systems, text mining, and social network analysis. He has served as a guest editor for *Pattern Recognition*, *IEEE Transactions on Computational Social Systems*, *Journal of Software and Systems*, and *World Wide Web Journal* etc. He is also the assistant editor-in-chief of the *World Wide Web Journal*.



Zhen Wang is a full professor with Northwestern Polytechnical University, Xi'an, China. He is a member of Academia Europaea and the European Academy of Sciences and Arts. He has authored or coauthored more than 100 scientific papers and obtained around 12,000 citations. His current research interests include artificial intelligence, network science, data mining and multi-agent learning games.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/csdl.

Authorized licensed use limited to: JILIN UNIVERSITY. Downloaded on April 02,2024 at 07:18:22 UTC from IEEE Xplore. Restrictions apply.