Geometric Inductive Matrix Completion: A Hyperbolic Approach with Unified Message Passing

ABSTRACT

Collaborative filtering is a central task in a broad range of recommender systems. As traditional methods train latent variables for user/item individuals under a transductive setting, it requires re-training for out-of-sample inferences. Inductive matrix completion (IMC) solves this problem by learning transformation functions upon engineered features, but it sacrifices model expressiveness and highly depends on feature qualities. In this paper, we propose *Geometric Inductive Matrix Completion* (**GIMC**) by introducing hyperbolic geometry and a unified message passing scheme into this generic task. The proposed method is the earliest attempt utilizing capacious hyperbolic space to enhance the capacity of IMC. It is the first work defining *continuous* explicit feedback prediction within non-Euclidean space by introducing hyperbolic regression for vertex interactions. This is also the first to provide comprehensive evidence that edge semantics can significantly improve recommendations, which is ignored by previous works. The proposed method outperforms the state-of-the-art algorithms with less than 1% parameters compared to its transductive counterparts. Extensive analysis and ablation studies are conducted to reveal the design considerations and practicability for a positive impact to the research community.

1 INTRODUCTION

Accurately capturing users' preferences and products' target clients is a key task in designing successful recommender systems [1, 39]. Collaborative filtering approaches utilize the user-item historical interactions to predict possible future interests. Among extensive literature, low-rank matrix factorization [21, 43, 56] demonstrates its simplicity but effectiveness while its deep learning extensions [2, 18] enhance the capacity of factorization models to a new level.

However, as most existing methods **parameterize individuals** directly under the *transductive setting*, (i.e. train a latent variable for each instance), it becomes an open discussion on how to better deal with out-of-sample individuals without the re-training process. To generalize the model to an *inductive setting*, researchers introduce *Inductive Matrix Completion* methods that preserve the generalization ability to predict unseen instances directly. Nevertheless, since inductive methods cannot explicitly parameterize individuals otherwise it requires re-training during out-of-sample inferences, most existing works resort to representation learning on side information for inductive capabilities, such as using profiles or demographics [9, 20, 38, 55]. By transferring the learning task onto user/item profiles is interesting, but these tasks highly depend on the assumption that the estimated low-rank matrix lies in the subspace spanned by the side information latent variables [20]. As this is a strong assumption, these methods are constrained by the quality of side information and they struggle to model complicated relational data. Recently, efforts have been made to conduct more advanced feature engineering directly with the rating matrix under inductive settings [15, 54]. However, since inductive methods have much fewer parameters (do not parameterize instances directly), they may still suffer from a lack of expressiveness for complex network structure and unique preferences. The dilemma still exists regarding the trade-off between inductive ability and model capacity. More advanced techniques are imperatively needed to better capture unique preferences even for out-of-sample instances.

Recently, GNNs are widely utilized for building recommender systems [17, 42, 47, 51] as user-item ratings can be naturally represented as bipartite graphs. However, existing graph-based methods mainly focus on modeling implicit feedback where edges are binary links, while explicit feedback is addressed with multiple sets of GNNs in parallel (i.e. each type of rating with one set of weights) [36, 39, 42, 47]. As vanilla approaches mainly concern propagating node representations along edges, conventional methods endure the same drawback that user-item interactions are sorely utilized as propagation channels between nodes, where edges' semantics are neglected. To solve similar problems, there are attempts to utilize edge embeddings in other fields. For instance, in bioinformatics, chemical bond embeddings are expressly included for molecular analysis since bond features are comparably easier to obtain [3, 50]. However, this type of method hasn't been broadly discussed in the context of recommender systems, especially under the inductive setting where model capacity is crucially needed for improvement. Thus, it opens a direction worth discussing whether edge embeddings can benefit recommendation tasks where explicit feedback is available.

Conventional Euclidean methods suffer from a limitation that model capacity is inherently bounded by dimensionality. However, relational data often contains a complex hierarchical structure where the volume of embedding space requires exponential growth like layers of trees [14, 31]. Recently, representation learning on non-Euclidean space gains substantial attention in neural networks research. For instance, people introduce continuous hyperbolic space embedding to learn the geometric structure of relational data with promising outcomes [13, 31, 32, 37]. In the recommender system field, due to the difficulty of defining the complex relationships between manifold vertices, previous works predominantly target the link prediction tasks with the style of "push and pull" loss functions where the predictions are binary [12, 29, 39, 46]. In terms of explicit feedback with multi-relational data, the gap of how to correctly regress manifold vertices interactions is still open to discussion.

In this paper, we tackle the aforementioned dilemmas under the inductive setting by (1) conducting a cooperative learning task that trains edge embeddings and node embeddings jointly, unifying their collective information for message passing; (2) further enhancing model expressiveness by integrating the hierarchical structure of sparse graphs with hyperbolic space, extending inductive methods into the geometric domain with an end-to-end learning process;

(3) extending existing hyperbolic link prediction practices into a generic matrix completion method by introducing multi-view node embeddings, allowing a robust and transferable hyperbolic regression task.

Specifically, we propose *Geometric Inductive Matrix Completion* (namely **GIMC**) based on advanced graph neural networks (GNNs) and hyperbolic geometry. Firstly, we introduce edge embeddings to model the semantic properties of different types of explicit feedback. Instead of treating edges as propagation channels within a conventional message passing scheme, edges are explicitly modeled as information sources and passed along with node embeddings for their cooperative effects. We empirically validate the effectiveness of this method and regard it as a promising technique for future explicit feedback prediction tasks. Secondly, as inductive methods inevitably deal with brand-new instances to very experienced/popular ones, a long tail and power-law distribution are naturally inherited. To further enhance the model capacity regarding such complex structures, we introduce hyperbolic geometry as the vertex level embedding space. Though the continuous tree-like structure of hyperbolic geometry serves as a more natural choice compared to its Euclidean counterparts, directly utilizing hyperbolic geometry brings up the question that existing methods only target link prediction tasks whereas our context is to predict explicit feedback that are either ordinal or continuous. To this end, in this paper we propose hyperbolic regression for interactions, extending existing hyperbolic recommender systems into a continuous setting where the target can be estimated as a continuous value. We validate the proposed method with extensive experiments including comparisons to the state-of-the-art methods and various ablation studies. We provide sufficient evidence and comprehensive analysis on the practicability of the proposed methods for a greater impact in the research community.

The main contributions of this paper are summarized as follows:

- We introduce **GIMC**, a novel method that improves inductive matrix completion (IMC).
- GIMC introduces new techniques to enhance the expressiveness of IMC for modeling complex network structure, extending IMC into non-Euclidean space and revealing novel insights to improve graph representation learning for explicit feedback prediction.
- To the best of our knowledge, this is the first work that bridges hyperbolic geometry and IMC by proposing hyperbolic regression for vertex interactions, achieving significant inductive and transfer learning ability.
- Extensive experimental results and theoretical analysis demonstrate the effectiveness and practicability of the proposed method.

2 RELATED WORKS AND PRELIMINARIES

Inductive Matrix Completion. Considering a sparse rating matrix with observations denoted as $\boldsymbol{R}_{u,v}, (u,v) \in \Omega$, a continuous collaborative filtering task aims to estimate a recovered interaction matrix: $\hat{\boldsymbol{R}}^* = \arg\min_{\hat{\boldsymbol{R}}} \sum_{(u,v)\in\Omega} \|r_{u,v} - \hat{r}_{u,v}\|^2$. Conventional matrix factorization method considers user/item interactions as dot products between transductive parameters [24] while neural networks methods consider similar approaches with deeper layers [18, 42, 47]. Though transductive methods are extensively studied in recent literature, *Inductive Matrix Completion* is less approached due to the difficulty of predicting out-of-sample instances. Jain and Dhillon [20] studied this generic problem by incorporating side information, such as users' age or movies' genre. With the assumption that the estimated low-rank matrix lies in the subspace spanned by the side information, they proved theoretical bounds so that the alternating minimization method can recover the underlying matrix. On top of [20], Chiang et al. [9] quantify the quality of side features and introduces an approach under the noisy condition while Si et al. [38] uses a feature transformation method that learns a mapping function in a supervised manner. In recent years, theoretical recovery guarantees for neural networks-based methods are discussed [55], which pushes the relevant research into the neural network domain. Hereby, the latest approaches for inductive matrix completion are based on GNNs. For instance, Zhang and Chen [54] demonstrates the predictive power of shallow-hop subgraph structure of rating matrix without side information. Wu et al. [49] conducts two separate steps as a hybrid method to reinforce personalized recommendations via pre-training ID-specific parameters, and querying pre-trained parameters for relational inference.

Hyperbolic Embeddings in Recommender Systems. Since relational data contains continuous tree-like structures which Euclidean space struggles to capture, researchers propose geometric representation learning within non-Euclidean spaces. Nickel and Kiela [31, 32] first introduce hyperbolic space for shallow network embeddings in the context of Poincaré and Lorentz model. Then, people extend hyperbolic geometry into deep neural networks by introducing hyperbolic neural networks (HNNs) [13, 37]. HNNs are utilized for various tasks, such as knowledge graphs [7, 40], image embeddings [22, 28], and natural language processing [41, 52]. In recent years, the community brings forward HNNs' graph extension by conducting message passing on tangent space for graph learning tasks and achieved promising results [8, 27]. In the field of recommender systems, Feng et al. [12], Vinh Tran et al. [46] study metric learning in hyperbolic space for user-item embeddings, Mirvakhabova et al. [29] conduct single layer auto-encoders while Sun et al. [39] interpret user-item embeddings as hyperbolic graph signals and conducts message passing on the tangent space for downstream tasks. Our proposed method is related to the aforementioned methods as it also utilizes hyperbolic geometry. The key difference is that our approach is particularly designed for inductive settings where out-of-sample instances can be directly predicted. Meanwhile, to best of our knowledge this is also the first work defining continuous regression for hyperbolic vertex interactions. Specifically, as non-Euclidean distances are normally used in the context of a push-and-pull based loss function such as margin ranking loss [39] or binary cross-entropy loss [29], previous methods cast hyperbolic collaborative filtering as a link prediction task where edges are binary. Differently, this paper proposes inductive matrix completion in a continuous setting where the target can be estimated as a continuous value.

Hyperbolic Geometry. Hyperbolic space is a special type of manifold where its curvature is negatively constant. It can be defined within the context of a Riemannian manifold and various equivalent models exist, including Poincaré, Lorentz, and Klein, etc [5]. In this paper, we define the proposed method on the most interpretive model as Poincaré ball. However, it is a trivial task to switch it into

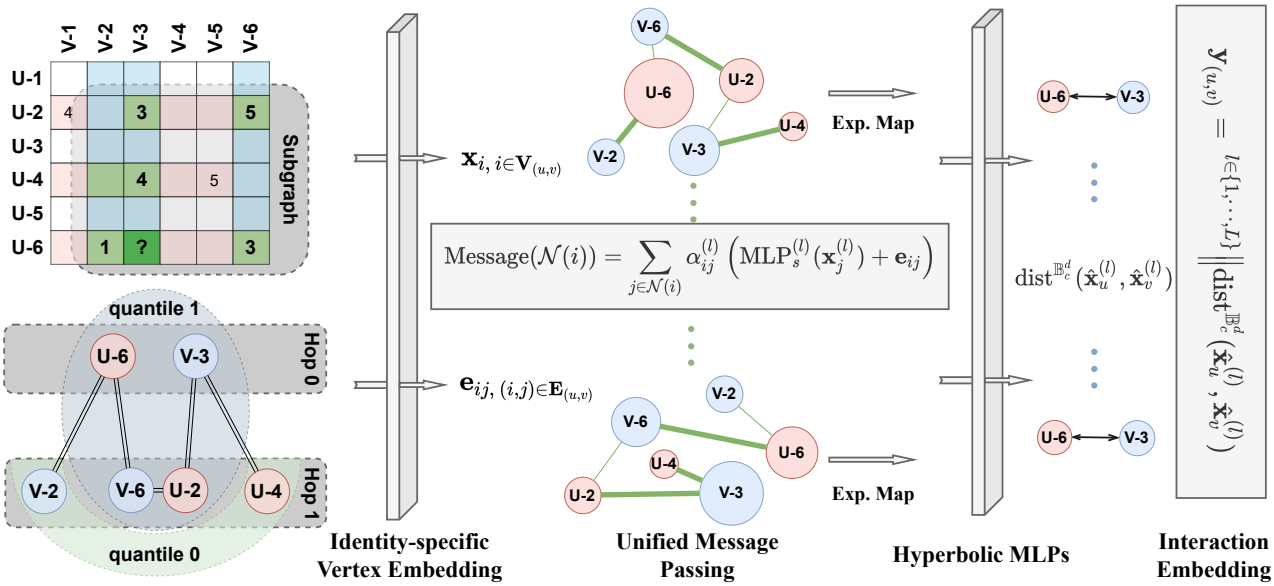


Figure 1: Our proposed Geometric Inductive Matrix Completion consisting of three subsequent stages.

the other models interchangeably. Considering g^E as the Euclidean metric tensor, the Poincaré ball model is defined as a Riemannian manifold with a set $\mathbb{B}_c^d$ and a metric tensor g^P:

$$\mathbb{B}_c^d = \{\mathbf{x} \in \mathbb{R}^d : \|\mathbf{x}\| < 1\}; \quad g^P(\mathbf{x}) = (\frac{2}{1 - \|\mathbf{x}\|^2})^2 g^E,$$

Then, the distance can be derived as:

$$\text{dist}^{\mathbb{B}_c^d}(\mathbf{x}, \mathbf{y}) = \text{arcCosh}(1 + 2\frac{\|\mathbf{x} - \mathbf{y}\|^2}{(1 - \|\mathbf{x}\|^2)(1 - \|\mathbf{y}\|^2)}). \quad (1)$$

3 METHODOLOGY

3.1 Inductive Feature Engineering

As inductive methods aim to conduct recommendation tasks for out-of-sample instances without the re-training process, the method should be able to extract initial features regarding new user/item during the inference stage. To achieve the aforementioned requirement, a natural attempt would be using side information, such as profiles or demographics. However, directly using the collected side information is a trivial task for neural network-based methods and the performance is highly dependent on the quality of available side information. Instead, this paper intends to explore the boundaries of feature engineering only from user-item interactions. In this subsection, we introduce *Subgraph extraction* and *Identity-specific vertex embedding* to generate initial features solely from the sparse rating matrix. The proposed method can be utilized exclusively with only rating matrices, but it preserves the ability to be extended for the scenario where high-quality side information is available.

Subgraph Extraction. Considering a user-item bipartite graph $\mathcal{G}$ with observed ratings denoted as $\boldsymbol{R}_{u,v}$, $(u, v) \in \Omega$, the process of message passing in the graph is similar to a diffusion process where direct/shallow hop neighbors carry more information compared to distant ones. To utilize this unique property for an inductive representation learning task, we first follow previous prosper practices [19, 53, 54] by applying breadth-first search (BFS) to extract an h-hop enclosing subgraph around (u, v) as $\mathcal{S}_{u,v} = \{\boldsymbol{V}_{u,v}, \boldsymbol{E}_{u,v}, \boldsymbol{W}_{u,v}\}$, where $\boldsymbol{V}_{u,v}$ is the vertices with h-hop interactions, $\boldsymbol{E}_{u,v}$ is the edges between $\boldsymbol{V}_{u,v}$, and $\boldsymbol{W}_{u,v}$ is the corresponding edge type. For instance, as shown in the first stage of Fig. 1, to generate the subgraph for the queried entry (U-6, V-3), BFS traverses its 1-hop neighbourhood and extract {U-6, V-3, V-2, V-6, U-2, U-4} as $\boldsymbol{V}_{u,v}$ along with their edges correspondingly. Notably, this feature engineering method is inductive and can be used for completely new users/items. Take the previous example, consider if U-6 has no existing interactions, then the subgraph is generated sorely around the target item V-3 with $\boldsymbol{V}_{u,v}$ = {U-6, V-3, U-2, U-4}. It is worth noting that, in terms of rating matrices with very sparse interactions, the extracted subgraphs normally inherit a hierarchical structure where the queried user-item can be considered as roots while the other vertices as their child nodes. Meanwhile, user-item data in recommender systems are naturally skewed since trending items undeniably gain more exposures and experienced customers indisputably give more ratings. Consequently, the subgraphs also acquire a power-law distribution [4, 6]. To target and fully exploit these properties for an easier upcoming learning task, we propose a fresh vertex embedding method in the following paragraph.

Identity-Specific Vertex Embedding. Common model-based collaborative filtering methods, including matrix factorization and factorization machines [35], take user/item IDs as initial features and use one hot encoder along with an embedding layer to parameterize them. For instance, consider m users, n items and d is the embedding size, typically at least $(m + n) \times d$ parameters shall be trained regarding IDs only. However, since new users or items results in new parameter vectors, this conventional parameterization approach is transductive because re-training is inevitable for new inferences. Instead, we initialize graph signals by using **degree effect**: $\boldsymbol{q}_{\text{deg}} \in \mathbb{R}^d$ and a common parameter group: $\boldsymbol{Q} \in \mathbb{R}^{|\mathcal{Q}| \times d}$, where each row of $\boldsymbol{Q}$ is a representation for a unique identity. In particular, the design of this parameter group shall (1) distinguish

the nature of different instances, (2) reflect the power-law structure of the extracted subgraphs, and (3) most importantly, have to be inductive. Namely, to achieve the aforementioned requirements, we consider each node as an aggregation of the following unique identities, including (a) **entity nature** (*Is the current node a user or item?*); (b) **hop identity** (*On which hop this node is located within the subgraph around the queried entry?*); and (c) **quantile identity** (*How experienced/popular this user/item is and in which quantile this user/item locates?*). In terms of quantile identity, we bin users/items into q quantile groups according to their number of interactions within the subgraph and set an embedding for each group. Specifically, denoting vertex-i with a sparse binary feature vector $\boldsymbol{v}_i \in \{0,1\}^{|\mathcal{Q}|}$ and its number of interactions in subgraph as degree: $\deg_i$, the corresponding graph signals of vertex-i is initialized as the summation of identity embeddings and degree effects:

$$\boldsymbol{x}_i^{(0)} = \sum_{k=1}^{|\mathcal{Q}|} \boldsymbol{v}_{ik} \cdot \boldsymbol{q}_k + \deg_i \boldsymbol{q}_{\deg}, \text{where } \boldsymbol{q}_k \in \mathcal{Q}. \tag{2}$$

Noted that we point out the differences between quantile embeddings and degree effect: (1) Degree effect is a parameter proportional to the experience/popularity of each user/item, i.e. items with more exposure and users with more experience carry more effects; (2) Quantile embedding is a **group-specific** method where each group carries the same amount of unique semantics; (3) Quantile embeddings are learned in parallel in terms of groups but degree effect is a sole parameter. According to the general convergence bound of matrix factorization with group effects as in *Corollary 1* in [4], methods ignoring group parameters lead to a larger loss in general. In terms of inductive matrix completion with sparse subgraphs, we use quantile groups and degree effects to encode the hierarchical and long-tail structure as initial graph signals and empirically verify its best performance compared to the other methods [42, 47, 54].

The aforementioned subgraph extraction and node initialization methods are **inductive** since both subgraphs and subgraph signals can be extracted for new users without the retraining process. Meanwhile, as the initialization only includes parameters for specific identities, the total number of parameters has been greatly reduced compared to its transductive counterparts. To embed such a small amount of unique identities, we choose vanilla Euclidean geometry since there is no extra capacity required for the embedding space at this stage. However, after the initialization step, a more geometric structure is expected in the constructed subgraphs. As the result, to realize the natural long tail distribution [6] of user-item interactions, we choose hyperbolic manifold for the vertex level representations since the constant negative curvature stereo-typically enable the hierarchical power-law distribution to be incorporated for the downstream tasks.

3.2 Unified Message Passing: explicit feedback as messages

As the proposed method targets explicit feedback where interactions are ordinal ratings instead of binary links, we point out that user-item interactions should also be considered as messages instead of mere connections through graph convolutional layers. Specifically, to obtain a predictive representation for the interactions between user-u and item-v from the generated subgraph $\mathcal{S}_{u,v} = \{\boldsymbol{V}_{u,v}, \boldsymbol{E}_{u,v}, \boldsymbol{W}_{u,v}\}$, we introduce unified message passing scheme to convolute node and edge embeddings jointly.

In detail, after initializing $\{\boldsymbol{x}_i^{(0)}\}, i \in \boldsymbol{V}_{u,v}$ as the inceptive graph signals via equation 2, we further define edge embedding parameters regarding different explicit feedback types. For instance, for a heterogeneous subgraph where ratings range from 1 to 5, we define $\boldsymbol{e}_t, t \in \{1,2,3,4,5\}$ as edge embeddings to explicitly model the user-item interactions. Then, for each node-i: $\{\boldsymbol{x}_i^{(l)}\}, i \in \boldsymbol{V}_{u,v}$ at convolution layer-l, we aggregate the information from i's node neighbours: $j \in \mathcal{N}(i)$ along with the semantics of their in-between interactions $\boldsymbol{e}_{ij}$ as:

$$\boldsymbol{x}_i^{(l+1)} = \text{MLP}_t^{(l)}(\boldsymbol{x}_i^{(l)}) + \text{Message}(\mathcal{N}(i)), \tag{3}$$

where the passed messages are defined by:

$$\text{Message}(\mathcal{N}(i)) = \sum_{j \in \mathcal{N}(i)} \alpha_{ij}^{(l)} \left(\text{MLP}_s^{(l)}(\boldsymbol{x}_j^{(l)}) + \boldsymbol{e}_{ij}\right). \tag{4}$$

Notably, $\alpha_{ij}^{(l)}$ is the aggregation weights that can be determined with different methods and two multilayer perceptron (MLP) layers are learned to map the target node and source node respectively. The major difference between unified message passing and the other graph convolutional networks (GCNs) is that: (1) GCNs treat edges as propagation channels to bridge information among vertices, (2) unified message passing not only uses edges as channels but also as information sources for convolution.

Edge embeddings have been utilized for graphs with edge features in other fields, such as chemical bonds of molecules [3, 50]. However, such type of methods hasn't gained much attention in the recommender system area. Conventionally, regarding implicit feedback tasks, people use GCNs where edges determine aggregation weights via its Laplacian matrix [39, 47, 54]. On top of this, people introduce graph attention networks (GATs) [45] to learn an importance coefficient for aggregation. However, GATs is still similar to GCNs where edges only indicate connections. In terms of explicit feedback, existing graph methods only consider each rating type as a standalone graph and learn multiple sets of GCNs weights separately, such as R-GCNS [36], GCMC [42] and IGMC [54]. The major drawback of these methods is that different ratings should not only be utilized as channels but also as information sources. We provide empirical evidence and comprehensive analysis to validate the superiority of the unified message passing scheme in Sec. 4.

Lastly, to define the aggregation weights as $\alpha_{ij}^{(l)}$, one could utilize any weighted averaging method while for our case we use the self-attention mechanism to calculate a contextual coefficient for each node-edge-node pair. Specifically, for $j \in \mathcal{N}(i)$ we compute its attention weight α_{ij} via the defined query and key as:

$$q_i^{(l)} = \boldsymbol{W}_q^{(l)} \boldsymbol{x}_i^{(l)} + b_q^{(l)}, \quad k_j^{(l)} = \boldsymbol{W}_k^{(l)} \boldsymbol{x}_j^{(l)} + b_k^{(l)}.$$

Then, the coefficient are calculated by:

$$\alpha_{ij}^{(l)} = \frac{\left\langle q_i^{(l)}, k_j^{(l)} + \boldsymbol{e}_{ij} \right\rangle}{\sum_{u \in \mathcal{N}(i)} \left\langle q_i^{(l)}, k_u^{(l)} + \boldsymbol{e}_{iu} \right\rangle}. \tag{5}$$

Compared to other heuristics weighted averaging methods, we find self-attention consistently performs the best under the inductive setting either from convergence speed or converged result. We conclude the major reason as self-attention mechanism broadens the expressiveness of the model for complex relationships as it

models the aggregation weight with contextual meaning. In detail, since the original graph signals are initialized inductively with identity-specific embeddings, the number of initial parameters of our model is greatly reduced compared to transductive methods (around $1:400$). As a result, to compensate the capacity of the model for unique preferences while still maintaining the inductive property, it is essential to introduce a more comprehensive contextual meaning for various combinations between message sources, targets, and interactions.

3.3 Hyperbolic Regression for Interactions

Similar to conventional 'cold start' problems, the number of interactions for new users/items is relatively low as inductive matrix completion is normally conducted for out-of-sample instances. Meanwhile, implicit feedback also plays a vital role in recommender systems since trending items with more exposures can generally have higher ratings while a curmudgeon user may give comparatively low ratings [4]. Consequently, the observed rating matrices are normally sparse and long-tail distributed [10, 48].

To tackle the aforementioned obstacles, we embed graph signals into the hyperbolic space to capture the power-law distribution. More precisely, consider a 1-hop subgraph where the number of edges is low due to sparsity and skewness, it is of less probability to observe cycles in the subgraph compared to densely connected ones. As the result, the graph is more similar to an **n-ary tree** structure (or forest). For an n-ary tree, the number of nodes with distances: $dist(root, node) = r$ grows exponentially as n^r. Similarly, consider a two-dimensional hyperbolic space with a constant negative curvature: $-c$, the area of the disc of radius r also grows exponentially as: $2\pi(\cosh(\sqrt{c}r) - 1)$ [25, 26], which makes hyperbolic space particularly suitable for the subgraphs as we extracted.

In order to utilize the geometric property of hyperbolic space for node embeddings, we cast Euclidean hidden state $\boldsymbol{x}_i^{(l)}$ computed from equation 3 as a hyperbolic tangent vector around origin in a Poincaré hyperbolic manifold with curvature $-c$, i.e. $\forall \boldsymbol{x}_i^{(l)} \in \mathcal{T}_{\boldsymbol{x}}\mathbb{B}_c^d$.Then, by taking the exponential map around the origin, $\boldsymbol{x}_i^{(l)}$ can be projected to the surface of Poincaré ball as a hyperbolic vertex embedding with its geometric properties reflected in the new hyperbolic coordinates:

$$\tilde{\boldsymbol{x}}_i^{(l)} = \exp_{\mathbf{0}}^c(\boldsymbol{x}_i^{(l)}) = \mathbf{0} \oplus_c \left(\tanh\left(\sqrt{c} \frac{\lambda_{\mathbf{0}}^c \|\boldsymbol{x}_i^{(l)}\|}{2} \right) \frac{\boldsymbol{x}_i^{(l)}}{\sqrt{c}\|\boldsymbol{x}_i^{(l)}\|} \right). \quad (6)$$

In view of the trade-off between simplicity and generalization ability of the model, we embed results of multiple message passing layers as a group of hyperbolic nodes, i.e. $\{\tilde{\boldsymbol{x}}_i^{(l)}\},\ l \in \{1, \cdots, L\}$. This design aims to overcome the large variance problems regarding the densities between different subgraphs **without introducing extra hyperparameters**. Obviously, more complicated pooling methods can be naturally incorporated within our pipeline but it is beyond the main phase of this paper.

Then, to make the final prediction for the queried entry on the curved space, vertex embeddings are further mapped via a hyperbolic linear layers as $\{\tilde{\boldsymbol{x}}_i^{(l)}\} \mapsto \{\hat{\boldsymbol{x}}_i^{(l)}\},\ l \in \{1, \cdots, L\}$ by:

$$\hat{\boldsymbol{x}}_i^{(l)} = \mathrm{MLP}^c(\tilde{\boldsymbol{x}}_i^{(l)}) := \boldsymbol{w} \left(1 + \sqrt{1 + c\|\boldsymbol{w}\|^2}\right)^{-1}, \quad (7)$$

where $\boldsymbol{w} := c^{-\frac{1}{2}} \sinh\left(\sqrt{c} v(\tilde{\boldsymbol{x}}_i^{(l)})\right)$, $v()$ is the linear function to the Poincaré ball model defined in [37], and L is the number of convolutional layers.

To regress the magnitude of the paired interaction between user-u and item-v, we obtain L pairs hyperbolic vertex embeddings $\left(\hat{\boldsymbol{x}}_u^{(l)}, \hat{\boldsymbol{x}}_v^{(l)}\right), l \in \{1, \cdots, L\}$ for a multi-view interaction learning task. We utilize hyperbolic distance function on Poincaré manifold $\mathbb{B}_c^d$ (equation 1) to compute L pairwise distances and concatenate the results as the final representation for the queried entry:

$$\boldsymbol{y}_{(u,v)} = \Big\|_{l \in \{1, \cdots, L\}} \mathrm{dist}^{\mathbb{B}_c^d}(\hat{\boldsymbol{x}}_u^{(l)}, \hat{\boldsymbol{x}}_v^{(l)}). \quad (8)$$

As the information is gathered from L-hops respectively, the above equation can be interpreted as a **multi-view embedding** for the interaction between user-u and item-v. Lastly, we use $\boldsymbol{y}_{(u,i)}$ for the final regression task:

$$loss = \frac{1}{|\Omega|} \sum_{(u,v) \in \Omega} \left(\boldsymbol{R}_{(u,v)} - \mathrm{Linear}(\boldsymbol{y}_{(u,i)}) \right), \quad (9)$$

where $\mathrm{Linear}(\boldsymbol{y}_{(u,i)})$ is a vanilla Euclidean linear layer.

Transferable Graph Reasoning Layers. Remarkably, the proposed multi-view embeddings for hyperbolic regression not only boost the robustness by tackling subgraphs with different densities, but also reinforce the transferability of the proposed method. As illustrated in Fig. 1, the first two stages of the proposed methods aim to extract information from subgraphs while the last stage selects appropriate node embeddings from multiple message passing layers for regression tasks. Since the first two stages of the model, i.e. the graph reasoning layers, only target information on different hops without a dedicated selection, their effectiveness is certainly transferable for datasets with completely different observation densities. We demonstrate this idea in Sec. 4.4 by transferring the trained model weights to different datasets but only re-train the final stage with astonishing results.

4 EXPERIMENTS

In this section, the proposed method is applied to several recommendation datasets to evaluate its effectiveness via (1) benchmark comparison with the latest state-of-the-art, and (2) ablation studies, including Sparsity Test, Transfer Learning, and visualizations.

4.1 Experiment Setup

We evaluate the proposed methods on five common recommender system datasets, including MovieLens-100K (ML-100K), MovieLens-1M (ML-1M), Douban, Flixster, and YahooMusic. A Data statistics summary has been included in Table 2. We follow previous works [49, 54] by splitting ML-1M, Douban, Flixster, and YahooMusic into 90% and 10%, ML-100K into 80% and 20% train/test sets. Notably, due to the low observation density, the test sets include **out-of-sample instances**. Hyperparameters are tuned based on Douban dataset and the best ones are used across all experiments, including *weight decay: 0, edge dropout: 0, hidden state dimension: 32, and learning rate: 0.001.* The learning rate is scheduled to decrease to $1e-5$ till training ends. In terms of batch size, we use 50 for all datasets except for YahooMusic, for which we set batch size as 16 otherwise we find it hard to converge. To bring the advantages of hyperbolic space for modeling tree structure, we use 1-hop subgraphs for all

Table 1: RMSE test results on Flixster, Douban, YahooMusic, MovieLens-100K, and MovieLens-1M.

		Model	Content	Flixster	Douban	YahooMusic	ML-100K	ML-1M
Tranductive		GRALS	Yes	1.245	0.833	38.0	0.945	0.831
		sRGCNN	Yes	0.926	0.801	22.4	0.929	**0.829**
		GC-MC	Yes	**0.917**	0.734	**20.5**	**0.905**	0.832
Hybrid		IDCF-NN	No	-	0.738	-	0.931	0.844
		IDCF-GC	No	-	**0.733**	-	**0.905**	0.839
Pure Inductive		IGC-MC	Yes	0.999 ± 0.062	0.990 ± 0.082	21.3 ± 0.989	1.142	1.259
		F-EAE	No	0.908	0.738	20.0	0.920	0.860
		PinSage	Yes	0.954 ± 0.005	0.739 ± 0.002	22.91 ± 0.629	0.951	0.906
		IGMC	No	0.872 ± 0.001	0.721 ± 0.001	19.1 ± 0.138	0.905	0.857
	gcn	**GIMC**$_{c=0}$	**No**	0.861 ± 0.016	0.730 ± 0.007	20.0 ± 1.128	0.911 ± 0.001	0.860
		GIMC$_{c=-1}$	**No**	0.853 ± 0.001	0.723 ± 0.001	20.0 ± 0.510	0.911 ± 0.001	0.860
	unified	**GIMC**$_{c=0}$	**No**	0.830 ± 0.001	**0.710 ± 0.002**	19.0 ± 0.500	**0.895 ± 0.001**	**0.853**
		GIMC$_{c=-1}$	**No**	**0.824 ± 0.001**	**0.710 ± 0.001**	**18.4 ± 0.200**	**0.895 ± 0.001**	0.855

* The best results of transductive and hybrid methods are highlighted with underlined bold blue font as: **1.0**. The best results of pure inductive methods are highlighted with bold black font as: **1.0**.
* Estimated parameter size on ML-1M: (1) transductive, GC-MC: 312k; (2) hybrid, IDCF-GC: 250k; (3) pure inductive, GIMC: 1.6k.
* The transferability of inductive methods are compared in Tab. 3.

datasets with at most 100 nodes for each graph (the redundant nodes have been excluded randomly). The model is trained with Adam optimizer [23] where the validity of Riemannian points is ensured among the forward path. All implementations are conducted with PyTorch framework [33] with 32GB RAM and one NVIDIA V100 graphics card. [1]

Table 2: Dataset statistics summary.

Dataset	#Users/#Items	Ratings	RatingTypes	Density	%Training/Test
ML-1M	6040/3706	1,000,000	$1, 2, \cdots, 5$	0.0447	90%/10%
Douban	3000/3000	136,891	$1, 2, \cdots, 5$	0.0152	90%/10%
ML-100K	943/1682	100,000	$1, 2, \cdots, 5$	0.0630	80%/20%
Flixster	3000/3000	26,173	$0.5, 1, \cdots, 5$	0.0029	90%/10%
YahooMusic	3000/3000	5,335	$1, 2, \cdots, 100$	0.0006	90%/10%

4.2 Benchmark Evaluation

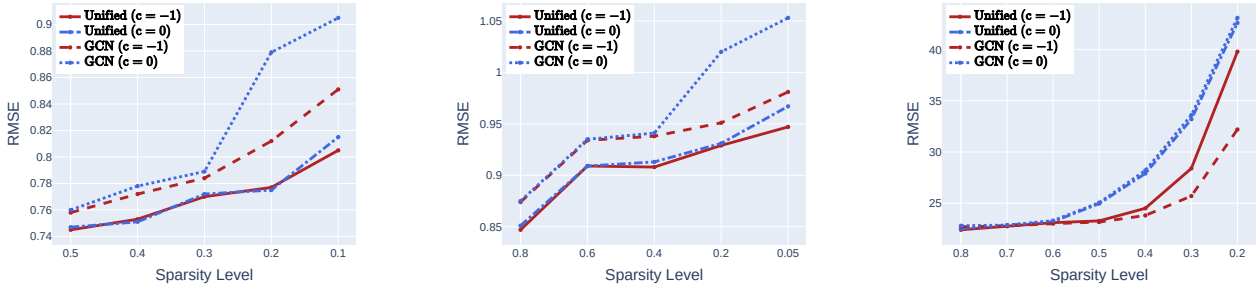
Since we target recommendation tasks with explicit feedback where the goal is to predict missing values in user-item rating matrix, we evaluate our methods by calculating **RMSE** to count the overall $l2$ distances between predicted ratings and testing observations. We follow [54] to store the model parameters and use the last 4 models for every 5 epochs (e.g. models at epoch 45, 50, 55, and 60 if train 60 epochs) to predict an ensemble results. The proposed method are compared to the lastest state-of-the-art (1) transductive models, including GRALS [34], sRGCNN [30], and GC-MC [42], (2) inductive models: IGC-MC [42], F-EAE [16], PinSage [51], IGMC [54], and (3) hybrid methods[2]: IDCF-NN and IDCF-GC [49].

We split the proposed methods as **4 subsequent variants** to validate the effectiveness of different considerations. Firstly, we use vanilla GCNs within a Euclidean context. The message passing uses normalized edge weights via Laplacian for aggregation and the layer-wise node embeddings are averaged for the final user-item dot product similarity measure (**gcn GIMC**$_{(c=0)}$). Secondly, we embed the GCNs approach within the hyperbolic geometry by projecting user-item embeddings into Poincaré ball model for hyperbolic MLPs. The final predictions are made with the pair-wise distances for regression (**gcn GIMC**$_{(c=-1)}$). Then, instead of conducting message passing with vanilla GCNs, we bin edges into 5 types and cast each type as one embedding parameter. We conduct unified message passing according to equation 3 on both Euclidean space (**unified GIMC**$_{(c=0)}$) and hyperbolic space (**unified GIMC**$_{(c=-1)}$). Comparisons of 4 variants with baselines are reported in Table 1.

As shown in Table 1, the proposed method achieves the best results compared to the latest pure inductive benchmarks. On the other hand, our model also outperforms all transductive and hybrid methods on four datasets but does not beat transductive and hybrid methods on ML-1M. We conclude this is due to the huge difference regarding parameter sizes on a densely connected graph. For instance, for ML-1M with the same hidden size, inductive methods have only around **0.4% parameters** compared to transductive ones and around **0.5%** compared to hybrid ones. Meanwhile, the proposed inductive method obtains the **transferability** since it focuses on the relational structure while hybrid or transductive methods can only be used for the same dataset. Furthermore, as we demonstrate the superiority of the proposed pure inductive approach, our model can also be hybridized with its transductive counterparts similar to IDCF-NN and IDCF-GC. In addition, the capacity of the proposed method can also be further extended by considering multi-head attention regarding equation 5 [44]. As this is not the main phase of the paper we leave this discussion for future work.

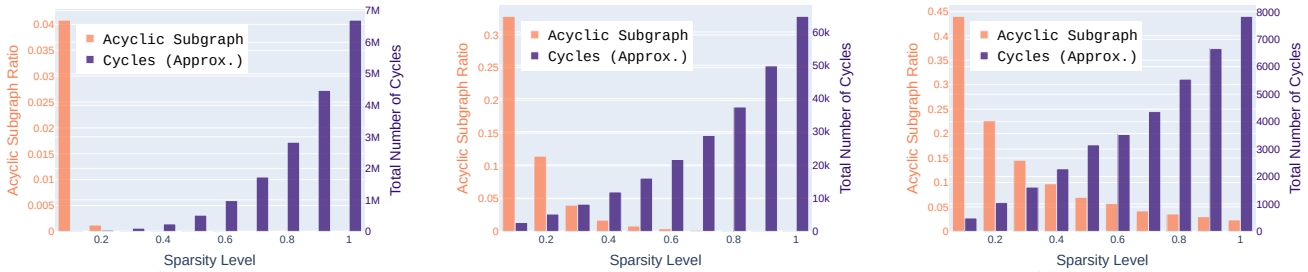
In terms of 4 variants, we observe that methods defined in hyperbolic geometry perform better for most circumstances, especially on datasets with low densities. For denser datasets such as ML-100K and ML-1M, hyperbolic geometry does not reveal performance gains. This validates our hypothesis that **hyperbolic geometry has its superiority on modeling sparse graphs with hierarchical structure**. In detail, regarding a low-density interaction

[1] Our code link will be available after the review process complying with the double-blind policy.
[2] Hybrid methods (IDCF-NN and IDCF-GC) conduct two separate steps as (1) pretrain transductive ID-specific embeddings via conventional matrix factorization method; (2) relational inference by querying embeddings from step (1).



(a) Douban (b) Flixster (c) YahooMusic

Figure 2: Sparsity test on Flixster, YahooMusic and Douban dataset.



(a) Douban (b) Flixster (c) YahooMusic

Figure 3: Ratio of acyclic subgraphs: orange. Approximated total number of cycles within four hops (purple).

matrix, the extracted 1-hop sparse subgraphs have a high chance of degenerating into an N-ary tree which implies a power-law distribution. We conduct further ablation studies on this in Sec. 4.3.

Secondly, it is obvious to notice that methods with the unified message passing scheme comprehensively outperform conventional graph neural networks. It demonstrates the fact that, for recommendation tasks with explicit feedback, **ratings carry more information compared to its implicit counterparts**. Meanwhile, conventional GCNs can not fully realize the benefits of explicit feedback as these methods only incorporate ratings as aggregation weights. Instead, the unified message passing scheme that explicitly defines edge embeddings is effective to expand GCNs' capacity for modeling complex interactions.

4.3 Sparsity Test

To validate the hypothesis that hyperbolic geometries can handle data sparsity and new users with few interactions. We conduct sparsity tests on low observation density datasets: Douban, Flixster, and YahooMusic with 4 variants of the proposed method. We construct train-test sets by defining random mask matrices with different ratios. For efficiency, models are trained with 30 epochs with a higher starting learning rate for ablation study ($lr = 0.01$, compared to $lr = 0.001$ in Table 1, both scheduled to decrease till 1e-5), and edge parameters are defined as direct embeddings of rating types. Line plots of testing RMSEs have been reported in Fig. 2. As shown, when datasets become sparser, RMSEs of all models increase due to a performance drop. However, there are differences regarding the three plots. We summarize them as below:

1. Methods with unified message passing outperform all their vanilla GCNs counterparts if the number of edge types is small. As can be noticed in Fig. 2a and Fig. 2b, trends of 4 variants on these two datasets are very similar where methods with unified message passing achieve better performance consistently. However, in Fig. 2c for YahooMusic, the method with unified message passing has very similar results with GCNs under Euclidean geometry ($c = 0$) while it suffers under the hyperbolic geometry. We conclude the reasons as follows: (a) As shown in Table 2, Douban and Flixster have only 5 and 10 different types of ratings, which means we define 5 and 10 different edge embeddings. However, YahooMusic has 100 different types of ratings with the smallest observation density. As the result, it becomes a hard task to train them given the limited samples, especially under hyperbolic geometry where more exponential functions are included in the computation path (e.g. cosh and sinh). Clearly, since the current approach does not reflect the magnitude of the original edge weights, e.g. embeddings between ratings 1 and 2 should be closer compared to embeddings between 1 and 10, one could experiment with more complicated parameterization or regularization methods in the future. We leave this discussion for future work.

2. Hyperbolic Geometry works at least on par with Euclidean Geometry but it performs better under the sparse scenario. As shown in Fig. 2, methods defined in hyperbolic geometry have very similar performance with their Euclidean counterparts when the data density is high, i.e. "> 0.3" for Douban, "> 0.4" for Flixster and "> 0.6" for YahooMusic. However, while the data becomes sparser, the long tail distribution can be better incorporated in hyperbolic geometry. We conclude this as: when a graph is very sparse with no cycles in it, it degenerates into an acyclic graph or N-*ary* tree structure where the hierarchical structure is clearer. Since hyperbolic geometry preserves special relativity between coordinates and the distance function, it captures such tree-like structure and embeds it as geometric information.

To further validate this hypothesis, we conduct an ablation study to demonstrate the relationship between sparsity and the hierarchical structure of datasets. We define two metrics to measure the graph structure as (a) the ratio of acyclic subgraphs, and (b) the total number of cycles. Firstly, we apply random mask matrices on three datasets with different sparsity levels. Then we conduct BFS search to extract h-hop subgraphs. For each subgraph, we determine if it is acyclic by:

Proposition 4.1. *Consider subgraph $\mathcal{S}_{u,v} = \{V_{u,v}, E_{u,v}\}$ with binarized adjacency matrix $\boldsymbol{A}$ and Laplacian $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}$, where $\boldsymbol{D}$ is the degree matrix, $\mathcal{S}_{u,v}$ is acyclic iff* $\text{trace}(\boldsymbol{L}) = 2 * \text{rank}(\boldsymbol{L})$.

Proof. Assume $S_{u,v}$ has ζ components, if it is an acyclic graph then each component is a tree. Define V_n as vertices within n-th components with $|V_n|$ edges in it. Hence the total number of edges in $S_{u,v}$ is: $|E_{u,v}| = \sum_{n=1}^{\zeta}(|V_n| - 1) = |V_{u,v}| - \zeta$. On the other hand, since the total number of zero eigenvalue of $\boldsymbol{L}$ is the number of connected components of $S_{u,v}$, i.e. $|V_{u,v}| - \text{rank}(\boldsymbol{L}) = \zeta$, we have $|E_{u,v}| = \text{rank}(\boldsymbol{L})$. Lastly, with Handshaking Lemma [11], we have $\sum_{i \in V_{u,v}} \deg(i) = 2|E_{u,v}|$, i.e. $|E_{u,v}| = \frac{1}{2}\text{trace}(\boldsymbol{L})$. As the result, $\text{trace}(\boldsymbol{L}) = 2 * \text{rank}(\boldsymbol{L})$, and $S_{u,v}$ has at least a cycle iff $\text{trace}(\boldsymbol{L}) >= 2 * (\text{rank}(\boldsymbol{L}) + 1)$. □

Secondly, to approximate the number of cycles in the large graph, we count the total number of paths where one can reach its root within 4 steps as $\sum_{s=1}^{4} \text{diag}(\boldsymbol{A}^s)$. We choose the number of steps as 4 since the unified message passing is generally conducted within 4 hops. Though this approach will count paths with repeated nodes, it is much more efficient compared to tree traversals and we only need an approximation for comparison.

As one can see from Fig. 3, the ratio of acyclic subgraphs decreases while the total number of cycles increases when the graph becomes denser, which validates our hypothesis. Another interesting observation is that the performance of hyperbolic and Euclidean geometry (in Fig. 2) deviates roughly from (0.2, 0.3) in Douban, (0.2, 0.4) in Flixster and (0.4, 0.6) in YahooMusic. These positions align with the "elbows" of the acyclic subgraph ratio in Fig. 3. Thus, the elbow method in clustering analysis may also be used as a data-driven heuristic to determine whether one needs to embed the graph into hyperbolic geometry in future tasks.

Table 3: RMSE: Transfer the model trained on ML-100K.

Model	Flixster	Douban	YahooMusic
	Transfer / Re-train		
IGC-MC	1.290 / 0.999	1.144 / 0.990	25.7 / 21.3
F-EAE	0.987 / 0.908	0.766 / 0.738	23.3 / 20.0
IGMC	0.906 / 0.872	0.759 / 0.721	20.1 / 19.1
GIMC$_{c=-1}$	**0.859**	**0.722**	**19.0**

* We list two results from [54] as baselines: (1) transferring the model of ML-100K; and (2) Re-train target data from scratch.

4.4 Multi-view Node Embeddings

To demonstrate the intuition of multi-view node embeddings for hyperbolic regression, we conduct two ablation studies.

Embedding visualization. We train the model on ML-100K dataset with batch size 128. By setting the number of unified message passing layers as 4 and the final hidden state dimension as 2, we record the user-item embeddings of one testing batch when the model converges. The user-item embeddings after different unified message passing layers are visualized in Fig. 4. As shown in Fig. 4, the graph signals become smoother after passing through more unified message passing layers. As the result, the relative locations of user-item pairs also change. Notably, since the embeddings are obtained only via the subgraph structure, each specific user/item has one group of unique embeddings learned from each subgraph. For instance, if user-u has μ interactions and we use L message passing layers, so that we generate $L * \mu$ embeddings for user-u in total. As the result, though the proposed method has a much smaller number of parameters compared to transductive models, its representation ability can also handle unique preferences.

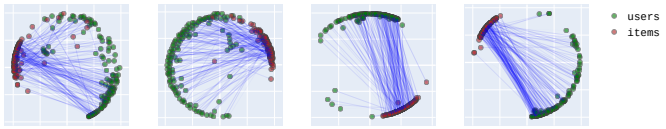


Figure 4: Multi-view Embeddings of user-item pairs after message passing layer 1 - 4 (left to right).

Transferable Graph Reasoning Layers. To validate the transferability of the proposed method, we conduct a transfer learning experiment. Firstly, we train the model with ML-100K dataset until the model converges. Then, we apply the trained model to Douban, Flixster, and YahooMusic datasets with the weights of the first two stages fixed but leave the final embedding layer trainable. Since Flixster and YahooMusic have different rating types, we bin their ratings into 5 quantiles to match the edge embeddings from ML-100K. Notably, our model achieves astonishing performance and converges at a much faster rate (**1-2 epochs**) as it only needs to update the weights of the final prediction layer. We list the results from [54] in Table 3 for comparison with the other inductive models. As shown, the proposed methods achieve excellent performance compared to baselines and even on par with baselines retraining on the target dataset over begin. We conclude this as the benefits of multi-view node embeddings since the proposed unified message passing layers accumulate information regarding different hops and the final prediction layer merges information from different hops selectively. Though unrelated datasets have completely different densities, the unified message passing layers still preserve the ability to summarize its geometrical meaning and pass it to the final prediction layer for a dedicated selection.

5 CONCLUSION

In this paper, we introduce Geometric Inductive Matrix Completion (GIMC). GIMC bridges the generic sparse graph structure with hyperbolic space by introducing identity-specific vertex embeddings and hyperbolic regression for interactions. It also empirically validated the effectiveness of edge embeddings for explicit feedback, which was predominantly ignored in the recommender system community. We corroborate our findings with comprehensive experiments and theoretical analysis, revealing intuitive and novel insights for future research.

REFERENCES

[1] Charu C Aggarwal et al. 2016. *Recommender Systems.* Springer.
[2] Ting Bai, Ji-Rong Wen, Jun Zhang, and Wayne Xin Zhao. 2017. A neural collaborative filtering model with interaction-based neighborhood. In *CIKM*.
[3] Gary Bécigneul, Octavian-Eugen Ganea, Benson Chen, Regina Barzilay, and Tommi Jaakkola. 2020. Optimal transport graph neural networks. In *ICLR*.
[4] Xuan Bi, Annie Qu, Junhui Wang, and Xiaotong Shen. 2017. A group-specific recommender system. *J. Amer. Statist. Assoc.* (2017).
[5] James W Cannon, William J Floyd, Richard Kenyon, Walter R Parry, et al. 1997. Hyperbolic geometry. *Flavors of geometry* 31 (1997).
[6] Oscar Celma and Paul Lamere. 2011. Music recommendation and discovery revisited. In *RecSys*.
[7] Ines Chami, Adva Wolf, Da-Cheng Juan, Frederic Sala, Sujith Ravi, and Christopher Ré. 2020. Low-Dimensional Hyperbolic Knowledge Graph Embeddings. In *ACL*.
[8] Ines Chami, Rex Ying, Christopher Ré, and Jure Leskovec. 2019. Hyperbolic graph convolutional neural networks. In *NeurIPS*.
[9] Kai-Yang Chiang, Cho-Jui Hsieh, and Inderjit S Dhillon. 2015. Matrix Completion with Noisy Side Information.. In *NeurIPS*.
[10] Aaron Clauset, Cosma Rohilla Shalizi, and Mark EJ Newman. 2009. Power-law distributions in empirical data. *SIAM Rev.* (2009).
[11] Leonhard Euler. 1741. Solutio problematis ad geometriam situs pertinentis. *Commentarii academiae scientiarum Petropolitanae* (1741).
[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 *SIGIR*.
[13] Octavian-Eugen Ganea, Gary Bécigneul, and Thomas Hofmann. 2018. Hyperbolic neural networks. In *NeurIPS*.
[14] 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. In *ICLR*.
[15] William L Hamilton, Rex Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *NeurIPS*.
[16] Jason Hartford, Devon Graham, Kevin Leyton-Brown, and Siamak Ravanbakhsh. 2018. Deep models of interactions across sets. In *ICML*.
[17] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *SIGIR*.
[18] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *WWW*.
[19] Cheng Hsu and Cheng-Te Li. 2021. RetaGNN: Relational Temporal Attentive Graph Neural Networks for Holistic Sequential Recommendation. In *WWW*.
[20] Prateek Jain and Inderjit S Dhillon. 2013. Provable inductive matrix completion. *arXiv preprint arXiv:1306.0626* (2013).
[21] Prateek Jain, Praneeth Netrapalli, and Sujay Sanghavi. 2013. Low-rank matrix completion using alternating minimization. In *STOC*.
[22] Valentin Khrulkov, Leyla Mirvakhabova, Evgeniya Ustinova, Ivan Oseledets, and Victor Lempitsky. 2020. Hyperbolic image embeddings. In *CVPR*.
[23] Diederik P Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *ICLR*.
[24] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009), 30–37.
[25] Dmitri Krioukov, Fragkiskos Papadopoulos, Maksim Kitsak, Amin Vahdat, and Marián Boguná. 2010. Hyperbolic geometry of complex networks. *Physical Review E* 82, 3 (2010), 036106.
[26] Dmitri Krioukov, Fragkiskos Papadopoulos, Amin Vahdat, and Marián Boguná. 2009. Curvature and temperature of complex networks. *Physical Review E* 80, 3 (2009), 035101.
[27] Qi Liu, Maximilian Nickel, and Douwe Kiela. 2019. Hyperbolic graph neural networks. In *NeurIPS*.
[28] Shaoteng Liu, Jingjing Chen, Liangming Pan, Chong-Wah Ngo, Tat-Seng Chua, and Yu-Gang Jiang. 2020. Hyperbolic visual embedding learning for zero-shot recognition. In *CVPR*.
[29] Leyla Mirvakhabova, Evgeny Frolov, Valentin Khrulkov, Ivan Oseledets, and Alexander Tuzhilin. 2020. Performance of hyperbolic geometry models on top-N recommendation tasks. In *RecSys*.
[30] Federico Monti, Michael M Bronstein, and Xavier Bresson. 2017. Geometric matrix completion with recurrent multi-graph neural networks. In *NeurIPS*.
[31] Maximillian Nickel and Douwe Kiela. 2017. Poincaré embeddings for learning hierarchical representations. In *NeurIPS*.
[32] Maximillian Nickel and Douwe Kiela. 2018. Learning continuous hierarchies in the lorentz model of hyperbolic geometry. In *ICML*.
[33] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*.
[34] Nikhil Rao, Hsiang-Fu Yu, Pradeep Ravikumar, and Inderjit S Dhillon. 2015. Collaborative Filtering with Graph Information: Consistency and Scalable Methods.. In *NeurIPS*.
[35] Steffen Rendle. 2010. Factorization machines. In *ICDM*. IEEE.
[36] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *ESWC*.
[37] Ryohei Shimizu, Yusuke Mukuta, and Tatsuya Harada. 2021. Hyperbolic neural networks++. In *ICLR*.
[38] Si Si, Kai-Yang Chiang, Cho-Jui Hsieh, Nikhil Rao, and Inderjit S Dhillon. 2016. Goal-directed inductive matrix completion. In *SIGKDD*.
[39] Jianing Sun, Zhaoyue Cheng, Saba Zuberi, Felipe Pérez, and Maksims Volkovs. 2021. HGCF: Hyperbolic Graph Convolution Networks for Collaborative Filtering. In *WWW*.
[40] Zequn Sun, Muhao Chen, Wei Hu, Chengming Wang, Jian Dai, and Wei Zhang. 2020. Knowledge Association with Hyperbolic Knowledge Graph Embeddings. In *EMNLP*.
[41] Alexandru Tifrea, Gary Bécigneul, and Octavian-Eugen Ganea. 2018. Poincar\'e glove: Hyperbolic word embeddings. In *ICLR*.
[42] Rianne van den Berg, Thomas N Kipf, and Max Welling. 2018. Graph Convolutional Matrix Completion. In *SIGKDD*.
[43] Bart Vandereycken. 2013. Low-rank matrix completion by Riemannian optimization. *SIAM Journal on Optimization* (2013).
[44] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NeurIPS*.
[45] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2018. Graph attention networks. In *ICLR*.
[46] 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 *WSDM*.
[47] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In *SIGIR*.
[48] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. 2021. Self-supervised graph learning for recommendation. In *SIGIR*.
[49] Qitian Wu, Hengrui Zhang, Xiaofeng Gao, Junchi Yan, and Hongyuan Zha. 2021. Towards Open-World Recommendation: An Inductive Model-based Collaborative Filtering Approach. In *ICML*.
[50] Yulei Yang and Dongsheng Li. 2020. Nenn: Incorporate node and edge features in graph neural networks. In *ACML*.
[51] 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 *SIGKDD*.
[52] Chengkun Zhang and Junbin Gao. 2020. Hype-HAN: Hyperbolic Hierarchical Attention Network for Semantic Embedding.. In *IJCAI*.
[53] Muhan Zhang and Yixin Chen. 2018. Link prediction based on graph neural networks. In *NeurIPS*, Vol. 31. 5165–5175.
[54] Muhan Zhang and Yixin Chen. 2020. Inductive Matrix Completion Based on Graph Neural Networks. In *ICLR*.
[55] Kai Zhong, Zhao Song, Prateek Jain, and Inderjit S Dhillon. 2019. Provable non-linear inductive matrix completion. In *NeurIPS*.
[56] Zhihui Zhu, Qiuwei Li, Gongguo Tang, and Michael B Wakin. 2018. Global optimality in low-rank matrix optimization. *IEEE Trans. Signal Process* (2018).