Enhancing Recommendation with Automated Tag Taxonomy Construction in Hyperbolic Space

Yanchao Tan†, Carl Yang‡, Xiangyu Wei†, Chaochao Chen†, Longfei Li§, Xiaolin Zheng†*
†*College of Computer Science, Zhejiang University, Hangzhou, China*
‡*Department of Computer Science, Emory University, Atlanta, United States*
§*Ant Financial Services Group, Hangzhou, China*
†*{yctan, weixy, zjuccc, xlzheng}@zju.edu.cn* ‡*j.carlyang@emory.edu* §*longyao.llf@antgroup.com*

Abstract*—The sparse interactions between users and items on the web have aggravated the difficulty of their representations in recommender systems. Existing approaches leverage tags to alleviate the data sparsity problem, so as to enhance the performance and interpretability of recommendation. However, directly using flat item tags fails to fully exploit the hierarchical relations in data, but tag taxonomies are not always available. To this end, we propose TaxoRec to jointly construct a tag taxonomy automatically and perform recommendation accurately in hyperbolic space. Specifically, we first leverage hyperbolic space and enable the optimization of a discrete taxonomy structure via a representation-aware scoring function and an adaptive clustering algorithm, and preserve the hierarchical structure for interpretability. Then, we propose to capture the complex relations among users, items, and tags in a unified hyperbolic metric space, where a novel tag-enhanced aggregation mechanism and tag-enhanced metric learning algorithm for users and items are defined. Extensive experiments on four real-world benchmark datasets show drastic performance gains brought by our proposed TaxoRec framework[1], which constantly achieves an average of 7.76% improvement over the state-of-the-art baselines regarding both Recall and NDCG metrics. Insightful case studies also show that our automatically constructed tag taxonomies are highly accurate and interpretable.

I. Introduction

In the era of information explosion, recommender systems (RSs) play a pivotal role in helping users alleviate the problem of information overload. Traditional collaborative filtering methods that only implicitly model users' preferences are limited by the sparsity of user-item interactions. To address the sparsity challenge in RSs, it is a common practice to combine collaborative filtering with auxiliary data. Among them, tags are one of the most commonly used types of data due to their vast availability and clear semantics, which can help profile users' preferences and item's properties [10], [28], [58], [29]. Existing approaches usually use the information of tags to implicitly group items so as to alleviate the sparsity of user-item interactions.

However, directly using flat item tags may fail to fully exploit the hierarchical relations among tags. To address this problem, a few studies [14], [22] start to take tag taxonomy into consideration. As shown in Fig. 1, a tag taxonomy is a tree-structured hierarchy, where parent nodes represent more

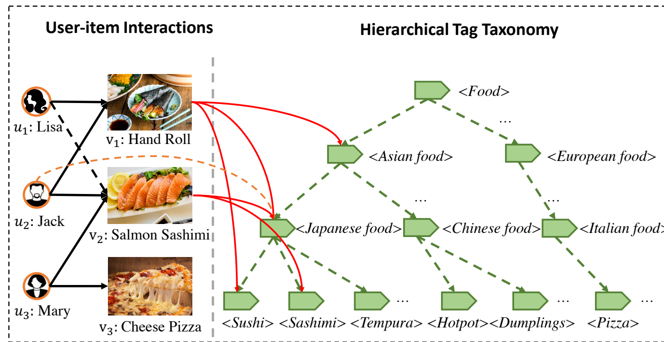


Fig. 1. An illustrative example of recommendation with the tag taxonomy. All solid lines are relations already in the data while the dotted lines refer to the relations that we want to infer. All tags are marked with <>.

abstract concepts than their children (*e.g.*, <*Japanese food*> and <*Chinese food*> are children of <*Asian food*>, whereas <*Sushi*> and <*Sashimi*> are children of <*Japanese food*>).

In this work, we propose to leverage the structured knowledge captured by such tag taxonomies to enhance the recommendation performance while providing valuable interpretability. As shown in Fig. 1, although Hand Roll and Salmon Sashimi are different kinds of food, both of them belong to <*Japanese food*>. By leveraging this extra knowledge, we may recommend Salmon Sashimi to Lisa who has interacted with Hand roll and other types of Japanese food before.

Although several methods [14], [22] directly use existing taxonomies to enhance recommendation, well-constructed taxonomies are not readily available in every situation and manual taxonomy construction is often very costly [57], [42], [55], [53]. Luckily, we find that user-item interactions and item-tag relations can be helpful to construct a tag taxonomy from scratch. As shown in Fig. 1, Jack has interacted with Hand roll (labeled as <*Japanese food*> and <*Sushi*>) and Salmon Sashimi (labeled as <*Japanese food*> and <*Sashimi*>), so we know that Jack might like <*Japanese food*>. Therefore, we can also infer the parent-child relations between <*Japanese food*> $\rightarrow$ <*Sushi*> and <*Japanese food*> $\rightarrow$ <*Sashimi*> for constructing the tag taxonomy.

Since taxonomy can enhance recommendation and user-item interactions can also help to construct taxonomy, a natural question is:

*Corresponding author
[1]https://github.com/Melinda315/TaxoRec

Can we simultaneously learn to construct taxonomy and perform recommendation?

In this work, we first propose a joint tag Taxonomy construction and Recommendation (TaxoRec) framework in hyperbolic space rather than the traditional Euclidean space. Existing studies about hyperbolic space [34], [35], [39] have proven that the representation ability of Euclidean space is not ideal to model complex patterns such as hierarchical structures. Motivated by recent improvements achieved by hyperbolic models [7], [48], [44], [54], we propose to leverage hyperbolic embeddings for our TaxoRec framework, which captures both the hierarchical relations among tags and the complex relations among users, items, and tags without additional supervision beyond the user-item and item-tag matrixes. Due to the different advantages of Poincaré [34] and Lorentz [35] models in clustering and optimizing, respectively, we novelly exploit these two types of hyperbolic models together and propose a TaxoRec framework in hyperbolic space (Section IV-B).

In particular, we first formulate the tag taxonomy construction problem as finding a discrete tag tree structure through optimizing the tag embeddings. Specifically, we propose a representation-aware scoring function to properly rank the tags. Then, we propose an adaptive clustering algorithm to capture the hierarchies among tags and explicitly generate a tag taxonomy in Poincaré model, where the Poincaré model is suitable for hierarchical clustering [8], [33]. Finally, to further leverage the constructed tag taxonomy for the modeling of the tag space, we propose a taxonomy-aware regularization objective (Section IV-C).

Based on the learned hyperbolic representations of tags, we propose a novel tag-enhanced hyperbolic representation of users and items by concatenating their corresponding tag-irrelevant and tag-relevant embeddings in Lorentz model, due to its effectiveness in optimization [25]. Since each item can have multiple tags, we develop a hyperbolic aggregation operation (*i.e.*, local aggregation operation) for items based on the Einstein midpoint method [17]. To model higher order similarity across the user-item bipartite graph (*e.g.*, neighbors-of-neighbors), we propose a global aggregation operation based on graph convolution network (GCN) [18], [44], [51]. The local and global aggregation operations form our tag-enhanced aggregation mechanism. Moreover, we propose a tag-enhanced metric learning algorithm to simultaneously learn tag-enhanced representation by considering the structure of the tag taxonomy (Section IV-D).

We evaluate the proposed TaxoRec with extensive experiments on four real-world benchmark datasets for recommendation with implicit feedback. We compare TaxoRec with 14 recommendation methods focusing on the state-of-the-art metric learning based, graph based, and tag based methods. Extensive experimental results show that TaxoRec is able to significantly improve the recommendation overall baselines (*e.g.*, with up to 13.83% relative improvements on Recall@10 over the best baseline). More comprehensive results and discussions as well as ablation studies, hyperparameter studies, and case studies are all presented in Section V.

In summary, we mainly make the following contributions:

- We propose a novel research problem of jointly constructing tag taxonomy and enhancing recommendation based on only user-item interactions and item-tag relations.
- We propose a TaxoRec framework, which can learn hyperbolic tag-enhanced representations of users and items along with automated taxonomy construction, through exploiting the individual strengths of Poincaré and Lorentz models simultaneously.
- We conduct extensive experiments on four real-world datasets, which demonstrate significant improvements of the proposed TaxoRec framework on recommendation together with highly accurate and interpretable taxonomy construction results.

II. Related work

A. Metric Learning for Recommendation

The metric learning methods for recommendation use Euclidean distance to measure the similarity between users and items. Compared with methods based on Matrix Factorization (MF) that assumes linear relationships between users and items and uses inner products to model the similarities of user-item pairs [38], [30], [56], metric learning methods that satisfy the triangle inequality can better model the complex interactions in real-world applications [43], [21], [46], and thus can address the limitations of MF. For example, [21] first proposed a method called collaborative metric learning (CML), which learns a metric space to encode not only users' preferences but also the user-user and item-item similarities. Since CML in the Euclidean space limit the representation of users and items, [36] turned this problem to one-to-one mappings between Euclidean and hyperbolic spaces. Moreover, to capture higher order graph structure for user (item) representation learning such as neighbors-of-neighbors relations among users and items, [44] measured the distance with hyperbolic graph convolutional neural networks model for collaborative filtering.

Although the above metric learning methods achieve promising performance, the learned representations that rely on user-item interactions only are limited by data sparsity. Though CML tried to leverage auxiliary data to alleviate sparsity, there is no metric learning work that has explicitly leveraged the hierarchical structural information like taxonomy for fine-grained user and item modeling.

B. Taxonomy-based Recommendation

Taxonomy has attracted tremendous attention in many application domains, due to its fundamental utility and the tree-structured hierarchy [55], [37], [53]. Previous methods in recommendation usually use taxonomy data for resolving the sparsity and costly computation problem. For example, [59] exploited taxonomic background knowledge to infer users' profiling effectively. [15] proposed to generate appropriate explanations for recommendation results with the aid of the taxonomy data. Besides taxonomies, many studies [49], [31] based on knowledge graphs (KGs) are applied to alleviate the sparsity problem. Although both KGs and tag taxonomies can be regarded as side information for recommender systems, KGs cover various semantic relations while tag taxonomies focus on the hierarchical relations. Compared with KGs, the advantages of incorporating a taxonomy are that (1) KGs cover lots of relations that are irrelevant to the recommendation task, blindly incorporating which can lead to high computational cost and even harm the recommendation performance, and (2) by focusing on the hierarchical relations, we can leverage the tree-structures of taxonomies and hyperbolic spaces to better arrange user/item embeddings to achieve more accurate and interpretable recommendations.

While the advantages of taxonomy seem eminent in recommendation, recent taxonomy-aware methods cannot work once the taxonomy data is unavailable. Inspired by the works in NLP that use machine learning algorithms to construct taxonomies from unstructured data [55], [11], we propose to automatically construct taxonomy from scratch to ensure the availability of structural knowledge. As closest to us, [57] proposed to discover a taxonomy from shopping data automatically based on a latent factor model. However, they only leverage the item-tag relations instead of the rich relations among users, items, and tags.

Note that, automated tag taxonomy construction is different from semantic annotation that mainly focuses on linking entities in the texts to their semantic descriptions [1], [2], [3]. Performing semantic annotation in recommender systems corresponds to linking tags to items (based on item contents such as review texts). However, in our setting, we already have all tags of items given in the dataset which do not really need semantic annotation. Instead, our goal is to organize these tags with the underlying hierarchical structure (taxonomy).

C. Hyperbolic Embedding Learning

The semi-structured and unstructured data (*e.g.*, text and tags) often contain an underlying hierarchical structure that is difficult to capture with representations in Euclidean space [12], [48]. To mitigate this problem, [34] proposed to learn representation in the Poincaré ball formulation of hyperbolic space that naturally accommodates hierarchical structures. Expanding on that work, [35] found that learning representations based on the Lorentz formulation of the hyperbolic space are well-suited for Riemannian optimization.

Recently, with the assumption that real-world data often reside on implicit hierarchical structures, hyperbolic representation learning has been applied to different problems [9], [17], [23], including recommendation [32], [12], [48]. For example, [32] used a single layer autoencoder in hyperbolic space to learn user and item embeddings. [48] studied metric learning in hyperbolic space and its connection to recommnedation. [12] applied hyperbolic learning for point of interest recommendation. Our approach is related to these works in that we also learn user and item representations in hyperbolic space. However, a key difference is that our approach can learn the hierarchical structure explicitly through automated tag taxonomy construction, which can further deliver accurate and interpretable recommendation.

III. Preliminaries

A. Recommender Systems

In recommender systems (RSs), we have historical interactions between users and items, where the interaction data can be represented as a bipartite graph $\mathcal{G} = \{(u, v)|u \in \mathcal{U}, v \in \mathcal{V}\}$. $\mathcal{U}$ and $\mathcal{V}$ denote the user and item sets. We consider recommendation based on the implicit feedback matrix $\mathbf{X}$, where $X_{uv} = 1$ denotes a positive sample (u, v_p), where user u interacted with item v_p, and $X_{uv} = 0$ denotes a negative sample (u, v_q), where the interaction between u and v_q is missing. To alleviate the sparsity problem in RSs, we adopt tags $t \in \mathcal{T}$ based on the attribute matrix $\mathbf{A}$, where $A_{vt} = 1$ denotes that item v has tag t.

B. The Models of Hyperbolic Space

Poincaré model. The Poincaré model $\mathcal{P}^d = \{\boldsymbol{x} \in \mathbb{R}^d : \|\boldsymbol{x}\| < 1\}$ is defined as a set of d-dimensional vectors with Euclidean norm smaller than 1. The Poincaré distance metric is defined as: $d_{\mathcal{P}}(\boldsymbol{x}, \boldsymbol{y}) = \cosh^{-1}\left(1 + 2\frac{\|\boldsymbol{x}-\boldsymbol{y}\|_2^2}{(1-\|\boldsymbol{x}\|_2^2)(1-\|\boldsymbol{y}\|_2^2)}\right)$. Another advantage of Poincaré model is the convenience of mapping with Klein model $\mathcal{K}^d = \{\boldsymbol{x}^K \in \mathbb{R}^d : \|\boldsymbol{x}^K\| < 1\}$ [54] via $f(\boldsymbol{x}) = \frac{2\boldsymbol{x}}{1+||\boldsymbol{x}||^2}$, where the hyperbolic embeddings can be aggregated via the Einstein midpoint method [23] as

$$\operatorname{HypAve}\left(\boldsymbol{x}_1^K, \ldots, \boldsymbol{x}_N^K\right) = \sum_{i=1}^{N} \gamma_i \boldsymbol{x}_i^K / \sum_{i=1}^{N} \gamma_i, \tag{1}$$

where $\gamma_i = 1/\sqrt{1 - ||\boldsymbol{x}_i||^2}$ is the Lorentz factor [23].

Lorentz model. The Lorentz model is the only unbounded hyperbolic model [54] and is defined as $\mathcal{L}^d = (\mathcal{H}^d, g_{\mathcal{L}})$ with points constrained by $\mathcal{H}^d = \{\boldsymbol{x} \in \mathbb{R}^{d+1} : \langle \boldsymbol{x}, \boldsymbol{x}\rangle_{\mathcal{L}} = 1, \boldsymbol{x}_0 \geq 0\}$, where $\langle \boldsymbol{x}, \boldsymbol{y}\rangle_{\mathcal{L}}$ is the Lorentzian scale inner product: $\langle \boldsymbol{x}, \boldsymbol{y}\rangle_{\mathcal{L}} = -\boldsymbol{x}_0\boldsymbol{y}_0 + \sum_{i=1}^{d} \boldsymbol{x}_i\boldsymbol{y}_i$, and the metric tensor is: $g_{\mathcal{L}}(\boldsymbol{x}) = \operatorname{diag}(-1, 1, \ldots, 1)$. The associated distance function in the Lorentz model is given as: $d_{\mathcal{H}}(\boldsymbol{x}, \boldsymbol{y}) = \cosh^{-1}(-\langle \boldsymbol{x}, \boldsymbol{y}\rangle_{\mathcal{L}})$.

Strengths and limitations. The metric in Poincaré model satisfies all the properties of a distance metric and is interpretable for visualization [35]. However, it is unstable to update the latent embeddings through this metric directly, especially when $||\boldsymbol{x}|| \sim 1$ or $||\boldsymbol{y}|| \sim 1$ [25]. The Lorentz allows for an efficient closed-form computation of the geodesics on the manifold, and can avoid numerical instabilities that arisen from the Poincaré distance [35], [5], [25]. Therefore, Lorentz model is better-suited for Riemannian optimization than Poincaré model [35].

Luckily, due to the equivalence of Poincaré and Lorentz models [35], we can exploit the models' individual strengths simultaneously. In particular, points in Lorentz model can be mapped into Poincaré model via diffeomorphism p as:

$$p(\boldsymbol{x}_0, \boldsymbol{x}_1, \cdots, \boldsymbol{x}_d) = \frac{(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_d)}{\boldsymbol{x}_0 + 1}. \tag{2}$$

Furthermore, points in Poincaré model can be mapped into Lorentz model via diffeomorphism p^{-1} as:

$$p^{-1}(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_d) = \frac{\left(1 + \|\boldsymbol{x}\|^2, 2\boldsymbol{x}_1, \cdots, 2\boldsymbol{x}_d\right)}{1 - \|\boldsymbol{x}\|^2}. \tag{3}$$

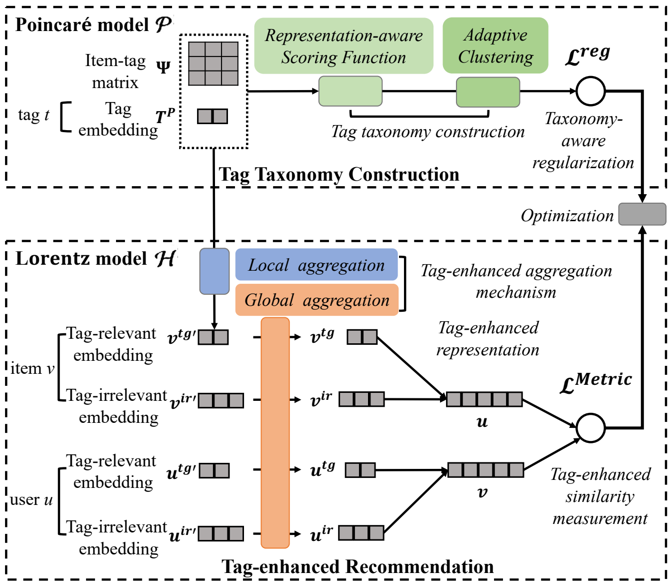


Fig. 2. The overall framework of TaxoRec.

IV. The TaxoRec Framework

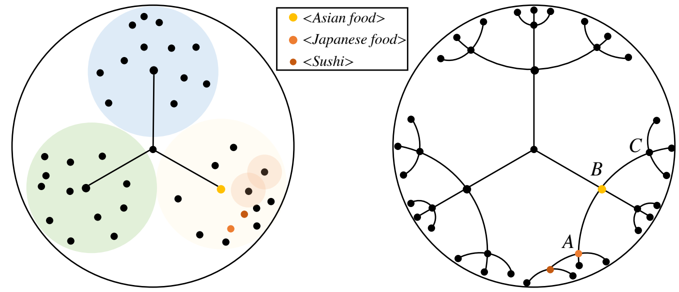
In this section, we present our joint tag Taxonomy construction and Recommendation (TaxoRec) framework in detail. We first give an overview of TaxoRec. Then, we construct tag taxonomy in Poincaré model. Furthermore, we propose a tag-enhanced representation method for users and item. To measure the similarity between users and items, we propose a hyperbolic tag-enhanced metric learning algorithm. Finally, we propose a hyperbolic optimization strategy to effectively train our model.

A. Method Overview

As shown in Fig. 2, the proposed TaxoRec framework includes two parts: (1) tag taxonomy construction in Poincaré model based on the item-tag matrix and tag embeddings; and (2) tag-enhanced recommendation in Lorentz model (Section IV-B).

To build a tree-structured tag taxonomy, we first initialize tag embeddings in Poincaré model and denote them as $\boldsymbol{T}^P = \{\boldsymbol{t}_1^P, \dots, \boldsymbol{t}_S^P\}$, where $\boldsymbol{t}_i^P \in \mathcal{P}^{D_t}$ and S is the number of tags. Starting from the root node that includes all tags, we generate fine-grained tag sets level by level via representation-aware scoring function and adaptive top-down clustering algorithm. Based on the constructed taxonomy, we update the tag embeddings $\boldsymbol{T}^P$ via taxonomy-aware regularization (denoted as $\mathcal{L}_{reg}$) (Section IV-C).

To model users and items from both tag-relevant and tag-irrelevant perspectives, we denote *tag-irrelevant embedding* for a user (*i.e.*, $\boldsymbol{u}^{ir'} \in \mathcal{H}^{D_i}$) and an item (*i.e.*, $\boldsymbol{v}^{ir'} \in \mathcal{H}^{D_i}$), and *tag-relevant embedding* for a user (*i.e.*, $\boldsymbol{u}^{tg'} \in \mathcal{H}^{D_t}$), where D_i and D_t are the embedding dimensions. Considering that an item can be represented by its tags whose embeddings are learned from the tag taxonomy, we denote a *tag-relevant embedding* $\boldsymbol{v}^{tg'}$ for an item based on tag embedding $\boldsymbol{T}^P$ by a local aggregation operation. Then, to capture higher order similarity across user-item bipartite graphs and represent users, items, and tags more precisely, we propose a global aggregation operation and obtain $\boldsymbol{u}^{ir}, \boldsymbol{v}^{ir}, \boldsymbol{u}^{tg}$, and $\boldsymbol{v}^{tg}$. By combining the tag-irrelevant embeddings with tag-relevant ones, we obtain a *tag-enhanced representation*, *i.e.*, $\boldsymbol{u} = [\boldsymbol{u}^{ir}, \boldsymbol{u}^{tg}] \in \mathcal{H}^{D_i} \times \mathcal{H}^{D_t}$ and $\boldsymbol{v} = [\boldsymbol{v}^{ir}, \boldsymbol{v}^{tg}] \in \mathcal{H}^{D_i} \times \mathcal{H}^{D_t}$. Finally, we compute the *tag-enhanced similarity* $g(\boldsymbol{u}, \boldsymbol{v})$ between user u and item v via the objectives of the tag-enhanced metric learning (denoted as $\mathcal{L}_{Metric}$) (Section IV-D).



(a) Euclidean space (b) Hyperbolic space

Fig. 3. The comparison between Euclidean and hyperbolic spaces. The volume of Euclidean space is polynomial, whereas the volume expands exponentially in hyperbolic spaces. Therefore, the sum of the distances between the points and the origin in hyperbolic space is much larger than the Euclidean distance, providing a better arrangement to separate embeddings of data points in more fine-grained levels of the hierarchy.

B. Modelling in Hyperbolic Space

Existing studies [34], [35], [39] have found flaws in Euclidean spaces, where the polynomial expansion has bounded the ability of the model to represent complex patterns by the dimensionality of embedding space. The problem is especially concerning in our joint tag taxonomy construction and recommendation framework due to the consideration of latent hierarchies.

Taking tags modeling for example, the limited sum of the distances between the tags and the origin will make it hard to arrange all hierarchical tags properly [40], [23]. As shown in Fig. 3, when representing a tag taxonomy with a two-dimensional embedding in Euclidean space, the model can only arrange the tag sets that reside near the origin in the taxonomy, where we can observe clear boundaries among tag sets in light blue, light green, and light yellow shallow. However, the Euclidean space fails to model the relations among tags that are embedded near the edge of unit ball (*i.e.*, $||\boldsymbol{t}||^2 \sim 1$), where we can observe some overlaps between two light orange shallows. In this case, it is unclear which tag sets should the tag in the overlap area belong to.

Furthermore, the suboptimal tag embedding optimization in Euclidean space will lead to suboptimal taxonomy construction, where the embeddings of some child tags can reside even closer to the origin compared with the one of their parent. For example, in Fig. 3(a), though <*Japanese food*> acts as <*Sushi*>'s parent for a more general concept, the embedding of <*Sushi*> near the edge of unit ball can be closer to root compared with the one of *<Japanese food>*. Due to the unclear relations among tags near the edge of the unit ball, the constructed tag taxonomy is weak due to the incorrect hierarchical relations.

To address these problems, we propose to leverage hyperbolic space for our TaxoRec, which is inspired by the recent studies about hyperbolic space [54], [48], [7], [44]. Recall that hyperbolic space has five models, which are isometric to each other [6], [13]. Among them, Poincaré model provides an intuitive way to layout the tags and thus is suitable for hierarchical clustering [35], [44]. Lorentz model is found to be more stable for numeric optimization [35], [44]. In light of these, we propose to construct a tag taxonomy in Poincaré model, whose goal is to mine hierarchical relations among tags. Then, we leverage the stable property of Lorentz model to optimize the whole TaxoRec framework.

In particular, we leverage Poincaré model for a better arrangement of tags with hierarchical structure inspired by [40], [23]. Since the volume in Poincaré model expands exponentially, the sum of the distances between the points and the origin is larger than that in Euclidean space. Therefore, we can represent a tag taxonomy in Poincaré model such that its structure is clearly reflected in tag embeddings even when $||\boldsymbol{t}||^2 \sim 1$. For example, we can observe clear hierarchies of *<Asian food>* $\rightarrow$ *<Japanese food>* $\rightarrow$ *<Sushi>* in Fig. 3(b) but not in Fig. 3(a). More importantly, we can observe that tag A can be closer to its immediate parent B while distant from its sibling C. Such clear hierarchical structures meet our goal of properly arranging tags in a taxonomy.

C. Tag taxonomy Construction

As motivated in Section I and IV-B, it is important to construct tag taxonomy in Poincaré model. In this way, we can leverage the hierarchies among tags to enhance the performance of recommendation while providing valuable interpretability. Though the idea of combining tag embedding and hierarchical clustering is intuitive by itself, two key challenges as follows need to be addressed for building high-quality tag taxonomies.

Fisrtly, the tag data lack pre-defined lexico-syntactic patterns (*e.g.*, *<Asian food>* contains *<Japanese food>*) to extract hypernym-hyponym tag pairs. For example, in Fig. 1, we only know Hand Roll is tagged with *<Asian food>*, *<Japanese food>*, and *<Sushi>*, where we do not know the relations among these three tags. In this case, it is challenging to properly represent tags and the relations among them for taxonomy construction.

Secondly, it is nontrivial to determine the proper granularity levels for different tags. When splitting a general tag set node into fine-grained ones, not all the tags should be pushed down to the child level. For example, when splitting a tag set {*<Asian food>*, *<Chinese food>*, and *<Japanese food>*...} in Fig. 1, the general tag *<Asian food>* should remain in the parent instead of being allocated into any child sets. Therefore, it is problematic to directly split parent tags to form child tag sets by general clustering.

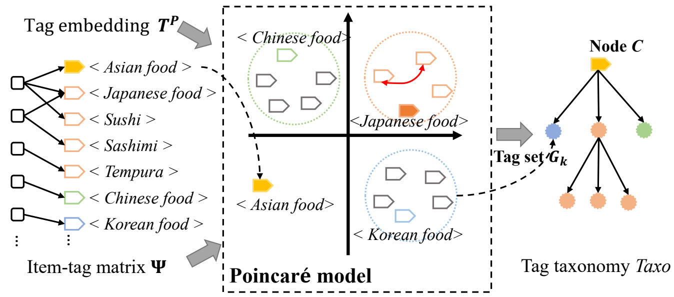


Fig. 4. The illustration of constructing tag taxonomy.

To address the above two challenges, we propose a representation-aware scoring function and an adaptive clustering algorithm. In this way, we can iteratively detect the hierarchies among tags without additional supervision beyond user-item interactions and item-tag matrix. Fig. 4 shows how these two components work together and we describe each component in detail below.

1) Representation-aware scoring function: To select a representative object in one node, we should ensure that this object's frequency and relevance are higher in the current node than that in its sibling [55]. Inspire by the above argument, we propose a *representation-aware scoring function*. Suppose a node C has a set of children $\mathcal{G}_C = \{G_1, G_2, \ldots, G_K\}$, then each $G_k (1 \leq k \leq K)$ should be a tag subset of C, and have the same semantic granularity with its siblings in $\mathcal{G}_C$. Through item-tag matrix $\boldsymbol{\Psi}$, we can obtain $\mathcal{E}_C = \{E_1, E_2, \ldots, E_K\}$, where each E_k is a set of item corresponds to the tag set G_k.

To properly represent tags, we first define two factors, namely *Context* and *Structure* as follows.

Context: Since a representative tag for G_k should appear frequently in corresponding item set E_k, we define $con(t, G_k)$ as the normalized frequency of tag t in G_k:

$$con(t, G_k) = \frac{\log(tf(t, E_k) + 1)}{\log(tf(E_k))}, \tag{4}$$

where $tf(t, E_k)$ is number of occurrences of tag t in E_k, and $tf(E_k)$ is the total number of tags in item set E_k.

Structure: Since one node's representative tag should be more relevant to the current node than that of its sibling, we define the concentration of tag t on G_k based on its relevance to the item set E_k:

$$stru\,(t, G_k) = \frac{\exp\left(\operatorname{rank}\left(t, E_k\right)\right)}{1 + \sum_{1 \leq j \leq K} \exp\left(\operatorname{rank}\left(t, E_j\right)\right)}, \tag{5}$$

where $\operatorname{rank}(t, E_k)$ is a retrieval function that ranks a set of items based on the query tags appearing in each item, which is defined as follow:

$$\operatorname{rank}(t, E_k) = \frac{idf(t) \cdot tf\left(t, E_k\right) \cdot (k_1 + 1)}{tf\left(t, E_k\right) + k_1 \cdot \left(1 - b + b \cdot \frac{tf(E_k)}{\text{avgdl}}\right)}, \tag{6}$$

where avgdl is the average tag number of each item in E_k. k_1 and b are parameters and are empirically set as 1.2 and 0.5,

Algorithm 1: Tag taxonomy construction.

Data: The item-tag matrix Ψ, the number of children K, the tag score threshold δ.
Result: K sets of tags.

$\mathcal{T}_{sub} \leftarrow \mathcal{T}$;
while *True* **do**
  $G_1, G_2, \ldots, G_K \leftarrow$ Poincaré-KMEANS($\mathcal{T}_{sub}, K$);
  for *k from 1 to K* **do**
    for $t \in \mathcal{T}_k$ **do**
      $s(t, G_k) \leftarrow$ socre of tag t for G_k in Eq. 7;
      if $s(t, G_k) < \delta$ **then**
        $G_k \leftarrow G_k - t$;
  $\mathcal{T}_{sub}^{'} \leftarrow G_1 \cup G_2 \cup \cdots \cup G_K$;
  if $\mathcal{T}_{sub}^{'} = \mathcal{T}_{sub}$ **then**
    Break;
  $\mathcal{T}_{sub} \leftarrow \mathcal{T}_{sub}^{'}$

respectively. Similar to [55], we calculate the inverse document frequency weight as $idf\,(t) = \ln\left(\frac{tf(E_k)-tf(t,E_k)+0.5}{tf(t,E_k)+0.5}+1\right)$.

Based on the above context and structure factors, we propose a representation-aware scoring function for a tag t in a tag set G_k. To ensure the selected representative tags to satisfy both factors, we have the following design similar to [41]:

$$s(t, G_k) = \sqrt{con(t, G_k) \cdot stru(t, G_k)}. \tag{7}$$

2) The adaptive clustering algorithm: Though we can measure the relations among tags via $s(t, G_k)$, the challenge about how to determine the level of tags still remains. If general tags co-occur with some fine-grained tags in the taxonomy, their embeddings tend to fall on the boundaries of different subsets, making it harder to discover clear subset of tags.

To address the above challenge, we propose an adaptive clustering algorithm in Poincaré model (shown in Fig. 4). Since the Poincaré model allows efficiently learning latent hierarchies among tags [33], we leverage the learned embeddings for iteratively clustering. To make the boundaries of clusters clearer, we first identify general tags and then refine the tag subset after pushing general tags back to the parent.

Algorithm 1 shows the way how to construct tag taxonomy via representation-aware scoring and adaptive clustering in hyperbolic space. Given a parent tag set $\mathcal{T}$, it first puts all the tags of $\mathcal{T}$ into the subset of tags $\mathcal{T}_{sub}$. Then it iteratively identifies general tags and refines the subset of tags. In each iteration, it computes the representativeness score of a tag t for the subset of tags G_k, and excludes t if its representativeness is smaller than a threshold δ. After pushing up general tags, it reforms the subset of tags $\mathcal{T}_{sub}$ and prepares for the next Poincaré-KMEANS operation [34]. The iterative process terminates when no more general tags can be detected, and the final subset of tags $G_1, G_2, \ldots, G_K$ are returned.

Finally, to leverage our constructed tag taxonomy and better model the reality of tags, we propose a taxonomy-aware regularization objective. Recall that the tag taxonomy construction is conducted in Poincaré model, where we denote tag embeddings as $\boldsymbol{T}^P$. By calculating scores for tags, we can split tags into different levels and different tag sets. Based on the constructed taxonomy, we can regularize the tag embeddings to be closer to the weighted center of the nodes that they belong to. In this way, there exist the positive correlation between the level of tags and degree of regularization, where the general tags that show up only in top levels will be less regularized than the fine-grained tags that appear in many tag sets in different levels. Based on the above intuition, we propose a taxonomy-aware regularization loss by traversing all nodes in the tag taxonomy as follows

$$\mathcal{L}^{reg} = \sum_{G_k \in Taxo} \sum_{t_i \in G_k} d_{\mathcal{P}}\Big(\boldsymbol{T}_i^P, \sum_{t_j \in G_k} \frac{s(t_j, G_k)\boldsymbol{T}_j^P}{\sum_{t_l \in G_k} s(t_l, G_k)}\Big), \tag{8}$$

where $d_{\mathcal{P}}(\boldsymbol{x}, \boldsymbol{y}) = \cosh^{-1}\left(1 + 2\frac{\|\boldsymbol{x}-\boldsymbol{y}\|_2^2}{(1-\|\boldsymbol{x}\|_2^2)(1-\|\boldsymbol{y}\|_2^2)}\right)$ measure the distance in Poincaré model. S is the number of tags. $G_k \in Taxo$ represent a node in the tag taxonomy and is formed by a set of tags.

D. Tag-enhanced Recommendation

Since we cannot identify whether a user interacts with an item because of its tags, it is important to model users and items from both tag-irrelevant and tag-relevant perspectives [52], [46]. For example, in Fig. 1, Jack may be easily attracted by items' tags, and his interacted Hand roll and Salmon Sashimi are both labeled as <*Japanese food*>. In this case, the interacted tags can reflect Jack's preference. However, the reason that Mary interacted with Salmon Sashimi and Cheese Pizza may be recommended from her friends. In this case, it is not suitable to model Mary as she like <*Japanese food*> and <*Italian food*>. To comprehensively model users and items, a straightforward idea is to directly combine the output of the tag taxonomy with learnable user and item embeddings in Euclidean space. However, this method not only separates the taxonomy construction and recommendation, but also ignores the latent hierarchies that exist in users and items [22]. Therefore, it fails to leverage user-item interactions for refining tag embeddings and further enhance recommendation.

To jointly achieve taxonomy construction and recommendation in a unified tag-enhanced framework, we propose to leverage tags as connections in hyperbolic space. The insight is that, an item can be represented by their tags that are learned from the hierarchical $Taxo$. By representing items via their tags, we can update the representation of users, items, and tags simultaneously. Specifically, we first denote learnable tag-irrelevant embeddings (*i.e.*, $\boldsymbol{u}^{ir'}$ and $\boldsymbol{v}^{ir'}$) to capture the collaborative latent representations of users and items. Since the tag information of users is missing, we propose to learn tag-relevant embeddings for users (*i.e.*, $\boldsymbol{u}^{tg'}$) together with the tag-irrelevant embeddings.

Note that items can have multiple tags, only keep one of them may harm the recommendation performance. As shown in Fig. 1, if we profile Jack with the only <*Sushi*> instead of using the combination of <*Asian food*>, <*Japaneses food*>, and <*Sushi*>, he may fail to share preference with the one who enjoys <*Asian food*> but have only interacted with some <*Chinese food*> before.

To accurately represent items' tag-relevant embeddings and obtain tag-enhanced representation for both users and item, we propose a tag-enhanced aggregation mechanism to properly aggregate tag embeddings $\boldsymbol{T}^P$ via *local* aggregation operation and aggregate the higher order similarities among users and items via *global* aggregation operation.

Different from traditional Euclidean space, hyperbolic space with a negative curvature cannot simply apply the Euclidean aggregation to obtain the centroid properties. To this end, we propose a local aggregation operation in hyperbolic space for items based on Einstein midpoint aggregation [17]. Specifically, we first map the tag embeddings $\boldsymbol{T}^P$ in Poincaré model to the Klein model, and obtain the tag embeddings in Klein model (denoted as $\boldsymbol{T}^K$) via:

$$\boldsymbol{T}^K = \frac{2\boldsymbol{T}^P}{1 + ||\boldsymbol{T}^P||^2}. \tag{9}$$

A hyperbolic aggregation of tag embeddings in Klein coordinates is computed as:

$$\boldsymbol{\mu} = local(\boldsymbol{\Psi}, \boldsymbol{T}^K) = \frac{\sum_i \gamma_i \boldsymbol{\psi}_i \boldsymbol{T}_i^K}{\sum_i \gamma_i \boldsymbol{\psi}_i} \tag{10}$$

where $\gamma_i = 1/\sqrt{1 - ||\boldsymbol{T}_i^K||^2}$ is the Lorentz factor and $\boldsymbol{\psi}_i \in \boldsymbol{\Psi}$ is from item-tag matrix. After that, we map the aggregated tag embeddings from Klein model to Lorentz model via:

$$\boldsymbol{v}^{tg'} = p^{-1}(\frac{\boldsymbol{\mu}}{1 + \sqrt{1 - \|\boldsymbol{\mu}\|^2}}), \tag{11}$$

where $p^{-1}(x_1, \cdots, x_d)$ can be referred in Eq. 3 in Section III).

To capture similarity across user-item bipartite graphs and encode such information into the final representation, we propose to leverage higher order information via a global aggregation operation $global$, which is under the graph convolutional setting [18], [44], [51] to produce tag-enhanced representations. Since we cannot apply Euclidean mean aggregation in hyperbolic space, we first project the embeddings to the corresponding tangent space as [44]. Taking user's tag-relevant embedding for example, we project $\boldsymbol{u}^{tg'}$ to $\mathcal{T}_o\mathcal{H}^d$ via the logarithmic map $\log_o : \mathcal{H}^D \to \mathcal{T}_o\mathcal{H}^D$ as follows:

$$\log_{\mathbf{o}}(\boldsymbol{u}^{tg'}) = \operatorname{arcosh}\left(-\langle \mathbf{o}, \boldsymbol{u}^{tg'}\rangle_{\mathcal{L}}\right) \frac{\boldsymbol{u}^{tg'} + \langle \mathbf{o}, \boldsymbol{u}^{tg'}\rangle_{\mathcal{L}}\mathbf{o}}{\|\boldsymbol{u}^{tg'} + \langle \mathbf{o}, \boldsymbol{u}^{tg'}\rangle_{\mathcal{L}}\mathbf{o}\|_{\mathcal{L}}}, \tag{12}$$

where $\mathbf{o} = (1, 0, \ldots, 0) \in \mathcal{H}^D$ is the referred origin and $\|\boldsymbol{u}^{tg'}\|_{\mathcal{L}} = \sqrt{< \boldsymbol{u}^{tg'}, \boldsymbol{u}^{tg'} >_{\mathcal{L}}}$.

Then, by taking $\boldsymbol{z}_u^{tg,0} = \log_{\mathbf{o}}(\boldsymbol{u}^{tg'})$ as the input to the first GCN layer in Euclidean space, we can aggregate neighborhood representation from the previous layer as:

$$\begin{aligned} \boldsymbol{z}_u^{tg,l+1} &= \boldsymbol{z}_u^{tg,l} + \sum_{v \in \mathcal{N}_u} \frac{1}{|\mathcal{N}_u|} \boldsymbol{z}_v^{tg,l}, \\ \boldsymbol{z}_v^{tg,l+1} &= \boldsymbol{z}_v^{tg,l} + \sum_{u \in \mathcal{N}_v} \frac{1}{|\mathcal{N}_v|} \boldsymbol{z}_u^{tg,l}, \end{aligned} \tag{13}$$

where $\mathcal{N}_u = \{v | R_{uv} = 1\} \in \mathcal{V}$ is the item set that user u interacts with. Similarly, $\mathcal{N}_v = \{u | R_{uv} = 1\} \in \mathcal{U}$ is the user set who interact with item v. Finally, we aggregate representation from all intermediate layers via global aggregation as:

$$\begin{aligned} \boldsymbol{z}_u^{tg} &= global(\boldsymbol{u}^{tg'}) = \sum_{l=1}^{L} \boldsymbol{z}_u^{tg,l}, \\ \boldsymbol{z}_v^{tg} &= global(\boldsymbol{v}^{tg'}) = \sum_{l=1}^{L} \boldsymbol{z}_v^{tg,l}, \end{aligned} \tag{14}$$

where L is the total number of layers. To project the final embedding back into Lorentz model, we apply an exponential map as follow:

$$\begin{aligned} \boldsymbol{u}^{tg} &= \exp_{\mathbf{o}}(\boldsymbol{z}_u^{tg}) \\ &= \cosh\left(\|\boldsymbol{z}_u^{tg}\|_{\mathcal{L}}\right) \mathbf{o} + \sinh\left(\|\boldsymbol{z}_u^{tg}\|_{\mathcal{L}}\right) \frac{\boldsymbol{z}_u^{tg}}{\|\boldsymbol{z}_u^{tg}\|_{\mathcal{L}}}. \end{aligned} \tag{15}$$

Similarly, we can obtain the represetation of $\boldsymbol{v}^{tg}$, $\boldsymbol{u}^{ir}$ and $\boldsymbol{v}^{ir}$ by replacing the inputs of Eq. 14 with $\boldsymbol{u}^{ir'}$ and $\boldsymbol{v}^{ir'}$, and then apply the exponential as Eq. 15.

By combining the tag-irrelevant embeddings with tag-relevant ones, *i.e.*, $\boldsymbol{u} = [\boldsymbol{u}^{ir},\ \boldsymbol{u}^{tg}] \in \mathcal{H}^{D_i} \times \mathcal{H}^{D_t}$ and $\boldsymbol{v} = [\boldsymbol{v}^{ir},\ \boldsymbol{v}^{tg}] \in \mathcal{H}^{D_i} \times \mathcal{H}^{D_t}$, we represent users and items via tag-enhanced representation $\boldsymbol{u}$ and $\boldsymbol{v}$.

Note that users have different levels of being attracted by tags, modeling all users with equal weights of tag-relevant and tag-irrelevant representation may lead to suboptimal performance of recommendation. However, existing similarity measurements are not designed for such tag-enhanced representation. To model users' personalized preferences towards tags, we propose to adaptively set users' personalized weights for tag-relevant embeddings as follows:

$$\alpha_u = \frac{\sum_{v \in \mathcal{V}_u} |\mathcal{T}_v|}{|\mathcal{V}_u| |\cup_{v \in \mathcal{V}_u} \mathcal{T}_v|}, \tag{16}$$

where $\mathcal{V}_u$ denotes the set of items that user u interacts with and $|\mathcal{V}_u|$ denotes the number of items. $\mathcal{T}_v$ denotes the set of tags that item v interacts with. $\cup$ represents an union operation for sets. We have $\alpha_u \in [0, 1]$. The idea behind Eq. 16 is to leverage the tripartite user-item-tag graph to calculate the ratio of the repeated tags in the whole interacted tag sets, *i.e.*, the more repeated tags of user u have, the more consistent preference of u towards these tags, and thus the more weights of tag-relevant should be considered for user u.

With such personalized weights α_u, we formulate the tag-enhanced user-item similarity by g as follows:

$$g(\boldsymbol{u}, \boldsymbol{v}) = d_{\mathcal{H}}^2(\boldsymbol{u}^{ir}, \boldsymbol{v}^{ir}) + \boldsymbol{\alpha}_u d_{\mathcal{H}}^2(\boldsymbol{u}^{tg}, \boldsymbol{v}^{tg}), \tag{17}$$

where $d_{\mathcal{H}}(\boldsymbol{x},\boldsymbol{y}) = \cosh^{-1}(-\langle \boldsymbol{x},\boldsymbol{y}\rangle_{\mathcal{L}})$ is a distance measurement in Lorentz model.

To learn a tag-enhanced similarity, we utilize the largest margin nearest neighbour algorithm (LMNN) for optimizing:

$$\mathcal{L}^{Metric} = \sum_{(\boldsymbol{u},\boldsymbol{v}_p)\in\mathcal{I}} \sum_{(\boldsymbol{u},\boldsymbol{v}_q)\notin\mathcal{I}} [m + g(\boldsymbol{u},\boldsymbol{v}_p) - g(\boldsymbol{u},\boldsymbol{v}_q)]_+ , \tag{18}$$

where $\mathcal{I}$ is the set of positive user-item pairs derived from the implicit feedback data $\mathbf{X}$. m is the margin to enforce the difference between triplets. $[(x)]_+ = \max(x, 0)$ denotes the standard hinge loss.

The final objective function of the proposed TaxoRec is given by considering the taxonomy-aware regularization (cf., Section IV-C) as follows:

$$\min_{\boldsymbol{u}^{ir},\boldsymbol{v}^{ir},\boldsymbol{u}^{tg},\boldsymbol{T}^P} \mathcal{L}^{Metric} + \lambda\mathcal{L}^{reg}, \tag{19}$$

where λ is a weight hyperparameter to control the regularization for tags.

E. Riemannian Optimization

Different to traditional Euclidean gradient descend optimization, we apply the Riemannian RSGD [4] for optimization. Denoting the $\mathcal{X}$ as the variable set, the parameters are updated by $\mathcal{X}_{t+1} = \exp_{\mathcal{X}_t}(-\alpha_t \operatorname{grad}(\mathcal{L}(\mathcal{X}_t)))$, where the $\exp$ operations in Poincaré model and Lorentz model are different and will be introduced later. The Riemannian gradient $\operatorname{grad}(\mathcal{L}(\mathcal{X}_t))$ can be obtained by

$$\operatorname{grad}(\mathcal{L}(\mathcal{X}_t)) = (\boldsymbol{I} - \mathcal{X}_t\mathcal{X}_t^T)\nabla(\mathcal{L}(\mathcal{X}_t)). \tag{20}$$

Optimizating $\mathcal{L}^{reg}$. In this scenario, tags are embedded in the Poincaré model, therefore we use Möbius exponential map:

$$\exp_{\boldsymbol{T}^P}(\eta) = \boldsymbol{T}^P \oplus \left(\tanh\left(\frac{\|\eta\|}{2}\right)\frac{\eta}{\|\eta\|}\right) = \boldsymbol{T}^P \oplus \boldsymbol{y}, \tag{21}$$

where $\boldsymbol{T}^P \oplus y$ denotes the Möbius addition:

$$\boldsymbol{T}^P \oplus \boldsymbol{y} = \frac{\left(1 + 2\langle \boldsymbol{T}^P, \boldsymbol{y}\rangle + \|\boldsymbol{y}\|^2\right)\boldsymbol{T}^P + \left(1 - \|\boldsymbol{T}^P\|^2\right)\boldsymbol{y}}{1 + 2\langle \boldsymbol{T}^P, \boldsymbol{y}\rangle + \|\boldsymbol{T}^P\|^2\|\boldsymbol{y}\|^2}. \tag{22}$$

Optimizating $\mathcal{L}^{Metric}$. In this scenario, the embeddings are computed by Lorentz model, where $\mathcal{X} = \{\boldsymbol{u}^{ir}, \boldsymbol{v}^{ir}, \boldsymbol{u}^{tg}, \boldsymbol{T}^P\}$. We take $\boldsymbol{v}^{ir}$ for example and show how to optimize in Lorentz model. The exponential map is defined as:

$$\exp_{\boldsymbol{v}^{ir}}(\eta) = \cosh(\|\eta\|_{\mathcal{L}})\boldsymbol{v}^{ir} + \sinh(\|\eta\|_{\mathcal{L}})\frac{\eta}{\|\eta\|_{\mathcal{L}}}. \tag{23}$$

V. Experiments

In this section, we evaluate our proposed TaxoRec framework focusing on the following four research questions:

- **RQ1:** How does TaxoRec framework perform compared to state-of-the-art recommendation methods?
- **RQ2:** What are the effects of the model components?
- **RQ3:** How do the hyperparameters affect the recommendation performance and how to choose optimal values?
- **RQ4:** Can TaxoRec construct hierarchical tag taxonomy?
- **RQ5:** How does TaxoRec provide interpretability for recommendation?

TABLE I
Statistics of the datasets used in our experiments.

Dataset	#User	#Item	#Interaction	Density(%)	#Tag
Ciao	5,180	8,836	104,905	0.229	28
Amazon-CD	32,589	20,559	515,562	0.077	331
Amazon-Book	79,368	62,385	4,614,162	0.094	510
Yelp	97,462	48,294	2,242,997	0.048	1138

A. Experimental Setup

1) Datasets: In order to comprehensively verify the effectiveness of compared methods, we use four real-world datasets from different application domains with different sizes and interaction densities, *i.e.*, Ciao[2], Amazon CDs & Vinyl (Amazon-CD)[3], Amazon Books (Amazon-Book)[3], and Yelp[4]. These datasets have been widely adopted in previous literature [36], [48], [45], and their statistics are summarized in Table I.

2) Evaluation protocols: We split the data into training, validation, and testing sets based on timestamps given in the datasets to provide a recommendation evaluation setting. For each user, we use the first 60% of data as the training set, 20% data as validation set, and 20% data as the testing set. We evaluate the recommendation performance using two metrics: Recall@K and NDCG@K instead of sampled metrics as suggested in [24]. Intuitively, the Recall metric considers whether the ground-truth is ranked amongst the top K items while the NDCG metric is a position-aware ranking metric.

3) Methods for comparison: The following representative state-of-the-art baselines can be divided into four groups: general recommendation methods (BPRMF, NMF, NeuMF), metric learning methods (CML, TransCF, LRML, SML, HyperML), graph based methods (NGCF, LightGCN, HGCF), and tag based methods (CMLF, AMF, AGCN):

- **BPRMF** [38]: The Bayesian personalized ranking (BPR) model is a popular method for Top-N recommendation and we adopt matrix factorization as the prediction component.
- **NMF** [26]: Non-negative matrix factorization (NMF) is a classic model that learns latent factors from interaction data.
- **NeuMF** [19]: NeuMF is a framework for applying neural networks to collaborative filtering, which combines multiple perceptrons with matrix factorization in its framework.
- **CML** [21]: Collaborative metric learning (CML) is the first model to use metric learning to solve the collaborative filtering problem of recommender systems.
- **TransCF** [36]: TransCF calculates the distance metric by learning the relationship vector between users and items.
- **LRML** [47]: Latent relational metric learning (LRML) employs an augmented memory module to induce a latent relation for each user-item interaction.
- **SML** [27]: Symmetric metric learning (SML) with learnable margins introduces a symmetrical positive item-centric metric to pull and push items via the dynamic margins.

[2]https://www.cse.msu.edu/ tangjili/datasetcode/truststudy.htm
[3]http://jmcauley.ucsd.edu/data/amazon/
[4]https://www.yelp.com/dataset

- **HyperML** [48]: Hyperbolic metric learning (HyperML) aims to bridge the gap between Euclidean and hyperbolic geometry in recommender systems through metric learning approach.
- **NGCF** [50]: NGCF is a graph based collaborative filtering model that follows the standard Graph convolutional neural network (GCN) [16], which iteratively learns user and item representations from aggregating neighbors' embeddings in the previous layers.
- **LightGCN** [18]: LightGCN devises a light graph convolution for training efficiency and generation ability.
- **HGCF** [44]: HGCF is a hyperbolic GCN architecture for collaborative filtering.
- **CMLF** [21]: CMLF integrates tags through a probabilistic interpretation of the mode that is based on CML.
- **AMF** [20]: Aspect-based Matrix Factorization model (AMF) is a MF-based model that decomposes the rating matrix with reviews.
- **AGCN** [51]: Adaptive Graph Convolutional Network (AGCN) leverage an attributed user-item bipartite graph for joint item recommendation and attribute inference.
- **TaxoRec**: TaxoRec is our proposed framework, which jointly learns a recommender system and tag taxonomy in hyperbolic space.

4) Implementation Details: We implement the proposed TaxoRec framework with Pytorch. The full code for this work is available[5]. Implementations of the general recommendation methods are either from open-source project or the original authors (BPRMF/CML[6], NMF[7], NeuMF[8], TransCF[9], LRML[10], SML[11], HyperML[12], NGCF[13], LightGCN[14], and HGCF[15]). Implementations of the tag-based methods are constrained to leverage item tags only according to the original authors (CMLF[6], AMF[16], and AGCF[17]). We optimize the compared Euclidean baselines with standard SGD and the hyperbolic ones with Riemannian SGD. We tune all hyperparameters through grid search. In particular, learning rate in {1e-5, 5e-5, 1e-4, 5e-4, 1e-3}, the number for splitting tag sets K in {2, 3, 4}, the tag score threshold δ in {0.25, 0.50, 0.75}, the number of graph layer L in {1, 2, 3, 4}, the margin m in {0.1, 0.2, 0.3, 0.4}, and the weight λ in {0, 0.01, 0.1, 1.0}. We set the embedding dimension D to 64 for those algorithms that do not include tags information. As for tag-based models (*i.e.*, CMLF, AMF, and our proposed TaxoRec), we set the tag embeddings D_t to 12 and the total embedding dimension D is still 64. The batch size is set to 10000. We also carefully tuned the hyperparameters of all baselines through cross-validation as suggested in the original papers to achieve their best performance.

[5]https://github.com/Melinda315/TaxoRec

[6]https://github.com/cheungdaven/DeepRec

[7]https://github.com/ninghaohello/Polysemous-Network-Embedding/

[8]https://github.com/hexiangnan/neural_collaborative_filtering

[9]https://github.com/pcy1302/TransCF

[10]https://github.com/vanzytay/WWW2018_LRML

[11]https://github.com/MingmingLie/SML

[12]https://github.com/lucasvinhtran/hyperml

[13]https://github.com/xiangwang1223/neural_graph_collaborative_filtering

[14]https://github.com/gusye1234/LightGCN-PyTorch

[15]https://github.com/layer6ai-labs/HGCF

[16]https://github.com/cthurau/pymf

[17]https://github.com/yimutianyang/AGCN

B. Overall Performance Comparison (RQ1)

In general, the proposed TaxoRec outperforms all 14 baselines across all evaluation metrics on all datasets, whose improvements are significant according to the Wilcoxon signed-rank test on 5% confidence level. This answers RQ1, showing that the joint learning of taxonomy construction and recommendation framework is capable of effective collaborative ranking. In particular, the performance gains of TaxoRec on Ciao, Amazon-CD, Amazon-Book, and Yelp range from reasonably large (3.01% achieved with NDCG@10 on Ciao) to significantly large (13.83% achieved with Recall@10 on Yelp). Note that the improvements of TaxoRec are more significant when the numbers of tags are larger and the hierarchies of tags are deeper, like with Yelp, which supports the appropriate design of our model to leverage the explicit hierarchical structure of associative tags. This result also shows that TaxoRec is effective in modeling hierarchical tags, as we will further demonstrate in the ablation study.

Moreover, by considering latent hierarchies in hyperbolic space, HGCF performs better than AGCN in many cases. However, their learned latent hierarchies do not always perfectly match the reality without the help of tag information, and thus AGCN can sometimes achieve better performance by considering flat item tags directly. Compared with AGCN, TaxoRec not only takes hierarchical tags into consideration but also aggregates tag-relevant embeddings with personalized weights in hyperbolic space. Therefore, TaxoRec outperforms AGCN by up to 19.26% in Recall@10 on Amazon-CD. The main differences between TaxoRec and HGCF reside in properly constructing and leveraging the hierarchical tags for recommendation. Specifically, TaxoRec can outperform HGCF by up to 15.16% in Recall@20 on Yelp.

Note that, the most time-consuming part of TaxoRec is the graph convolutional layer, which has also been used in the second runners (*e.g.*, HGCF and ACGN) to capture higher order graph structure. Relative to that, the overhead from our automated tag taxonomy construction is quite minor. Specifically, the time complexity of constructing tag taxonomy is $\mathcal{O}(S)$, where S is the number of tags and is far less than the number of users and items. In our experiments, we also found the runtimes of TaxoRec are in the same scale with the most graph based baselines.

C. Model Ablation (RQ2)

To better understand our proposed techniques, *i.e.*, tag-enhanced aggregation (Agg), taxonomy-aware regularization, and hyperbolic space setting, we study TaxoRec as follows:

- CML is the basic metric learning model in Euclidean space;
- CML + Agg is the model with tag-enhanced aggregation mechanism in Euclidean space, which consider item tags and the higher order relations in representation;

TABLE II
EXPERIMENTAL RESULTS (%) ON FOUR BENCHMARK DATASETS, WHERE * DENOTES A SIGNIFICANT IMPROVEMENT ACCORDING TO THE WILCOXON SIGNED-RANK TEST. THE BEST PERFORMANCES ARE IN BOLDFACE AND THE SECOND RUNNERS ARE UNDERLINED.

Method	Recall@10	Recall@20	NDCG@10	NDCG@20	Recall@10	Recall@20	NDCG@10	NDCG@20
	Ciao				Amazon-CD			
BPRMF	3.18±0.13	4.90±0.15	2.26±0.10	3.15±0.16	6.18±0.21	9.55±0.26	4.42±0.20	5.37±0.24
NMF	3.05±0.11	4.72±0.16	2.14±0.09	3.09±0.11	4.99±0.16	7.68±0.14	3.85±0.13	4.77±0.15
NeuMF	3.27±0.18	5.13±0.20	2.73±0.19	3.26±0.20	6.06±0.21	8.44±0.23	4.19±0.23	4.96±0.22
CML	3.67±0.23	5.84±0.26	2.68±0.19	3.40±0.21	6.22±0.15	9.60±0.17	4.55±0.14	5.66±0.19
TransCF	3.50±0.18	5.41±0.19	2.53±0.15	3.37±0.16	6.28±0.18	9.81±0.16	4.66±0.19	5.87±0.18
LRML	3.34±0.19	4.92±0.24	2.48±0.21	3.28±0.17	6.32±0.21	9.77±0.19	4.65±0.21	5.92±0.24
SML	3.60±0.17	5.76±0.19	2.75±0.16	3.44±0.15	6.33±0.25	9.83±0.22	4.92±0.18	6.06±0.20
HyperML	3.81±0.21	6.17±0.26	2.96±0.16	3.74±0.21	7.89±0.25	12.03±0.21	5.79±0.21	7.11±0.26
NGCF	4.71±0.14	7.39±0.19	3.78±0.18	4.66±0.21	7.99±0.20	11.70±0.20	6.08±0.18	7.37±0.20
LightGCN	5.17±0.18	7.86±0.14	4.17±0.15	5.10±0.15	9.77±0.14	14.22±0.15	7.47±0.14	9.02±0.17
HGCF	5.98±0.13	9.35±0.11	4.80±0.13	5.90±0.11	10.01±0.12	14.56±0.12	7.58±0.16	9.21±0.17
CMLF	3.73±0.21	5.92±0.24	2.79±0.18	3.52±0.18	6.32±0.25	9.71±0.23	4.72±0.24	5.79±0.24
AMF	3.56±0.21	5.46±0.23	2.65±0.24	3.41±0.28	6.25±0.27	9.61±0.25	4.72±0.22	5.82±0.24
AGCN	6.10±0.08	9.14±0.11	4.99±0.06	5.86±0.10	9.07±0.13	13.63±0.12	7.11±0.10	8.35±0.11
TaxoRec	**6.33±0.07***	**9.71±0.12***	**5.14±0.08***	**6.19±0.10***	**10.82±0.12***	**15.47±0.10***	**8.34±0.11***	**9.84±0.09***
	Amazon-Book				Yelp			
BPRMF	4.14±0.14	7.26±0.13	5.34±0.12	6.23±0.18	3.25±0.14	5.56±0.11	2.83±0.16	3.04±0.10
NMF	3.99±0.12	6.58±0.11	4.72±0.15	5.61±0.11	2.25±0.13	3.75±0.18	1.74±0.09	2.28±0.08
NeuMF	4.22±0.15	7.28±0.16	5.41±0.14	6.31±0.15	3.28±0.11	5.74±0.18	2.89±0.14	3.24±0.14
CML	4.53±0.11	7.64±0.15	5.85±0.09	6.92±0.06	3.56±0.13	6.24±0.16	3.24±0.16	4.21±0.15
TransCF	4.27±0.18	7.32±0.15	5.49±0.17	6.38±0.14	3.44±0.12	6.50±0.13	3.07±0.18	4.17±0.12
LRML	4.34±0.15	7.45±0.19	5.50±0.16	6.41±0.11	3.39±0.19	5.70±0.15	2.95±0.18	3.84±0.17
SML	4.42±0.18	7.57±0.11	5.65±0.11	6.62±0.14	3.67±0.14	6.40±0.13	3.16±0.11	4.35±0.15
HyperML	4.79±0.21	7.94±0.23	6.18±0.17	7.20±0.18	4.01±0.19	6.81±0.17	3.25±0.14	4.10±0.16
NGCF	4.19±0.15	6.84±0.13	5.34±0.12	6.23±0.13	3.12±0.11	5.47±0.12	2.42±0.11	3.25±0.16
LightGCN	4.36±0.10	7.11±0.09	5.53±0.09	6.44±0.12	3.81±0.10	6.70±0.09	3.01±0.09	4.11±0.12
HGCF	4.84±0.12	7.99±0.11	6.15±0.15	7.18±0.15	4.04±0.11	6.92±0.13	3.28±0.14	4.20±0.17
CMLF	4.63±0.14	7.66±0.13	5.87±0.07	6.95±0.12	3.99±0.14	6.62±0.15	3.36±0.09	4.27±0.10
AMF	4.57±0.13	7.60±0.19	5.79±0.18	6.73±0.14	3.58±0.13	6.13±0.18	3.02±0.13	4.18±0.12
AGCN	4.63±0.12	7.67±0.13	5.92±0.11	7.01±0.13	4.05±0.11	7.17±0.12	3.19±0.09	4.22±0.11
TaxoRec	**5.28±0.12***	**8.64±0.11***	**6.82±0.14***	**7.79±0.16***	**4.61±0.08***	**7.97±0.12***	**3.59±0.09***	**4.80±0.11***

TABLE III
ABLATION ANALYSIS OF OUR PROPOSED TAXOREC ON THE FOUR DATASETS.

	Ciao				Amazon-CD			
	Recall@10	Recall@20	NDCG@10	NDCG@20	Recall@10	Recall@20	NDCG@10	NDCG@20
CML	3.67%	5.84%	2.68%	3.40%	6.22%	9.60%	4.55%	5.66%
CML + Agg	5.52%	8.52%	4.54%	5.38%	9.02%	13.07%	6.95%	8.29%
Hyper + CML	3.81%	6.17%	2.96%	3.74%	7.89%	12.03%	5.79%	7.11%
Hyper + CML + Agg	6.10%	9.71%	4.97%	6.16%	10.39%	15.03%	7.92%	9.45%
TaxoRec	6.33%	9.79%	5.14%	6.19%	10.82%	15.47%	8.34%	9.84%
	Amazon-Book				Yelp			
	Recall@10	Recall@20	NDCG@10	NDCG@20	Recall@10	Recall@20	NDCG@10	NDCG@20
CML	4.53%	7.64%	5.85%	6.92%	3.56%	6.24%	3.24%	4.21%
CML + Agg	4.70%	7.70%	6.05%	6.96%	3.93%	6.76%	3.31%	4.41%
Hyper + CML	4.79%	7.94%	6.18%	7.20%	4.03%	6.84%	3.28%	4.29%
Hyper + CML + Agg	4.99%	8.17%	6.51%	7.50%	4.37%	7.53%	3.50%	4.47%
TaxoRec	5.28%	8.64%	6.82%	7.79%	4.61%	7.97%	3.59%	4.80%

- Hyper + CML is the basic collaborative metric learning model in hyperbolic space;
- Hyper + CML + Agg is the model with tag-enhanced aggregation mechanism in hyperbolic space, which also consider item tags and the higher order relations in representation;
- TaxoRec integrates Hyper + CML + Agg with taxonomy-aware regularization to leverage the hierarchies among users, items, and tags in hyperbolic space.

From Table III, we have the following observations:

The performance gains of CML + Agg over CML on four datasets fluctuate, ranging from 1.18% (achieved in Recall@20 on Amazon-Book) to 69.40% (achieved in NDCG@10 on Ciao). Similarly, the corresponding performance gains of Hyper + CML + Agg over Hyper + CML ranges from 2.90% (achieved in Recall@20 on Amazon-Book) to 67.87% (achieved in NDCG@10 on Ciao). These results show the enhancement brought by our tag-enhanced aggregation mechanism regarding both performance and robustness. Interesting, the improvements of tag-enhanced aggregation module are most significant on the Ciao dataset, where the number of

TABLE IV
HYPERPARAMETER STUDIES ON AMAZON-BOOK AND YELP.

Param.	Recall@10	NDCG@10	Recall@10	NDCG@10
	Amazon-Book		Yelp	
K = 2	5.20%	6.69%	4.29%	3.28%
K = 3	**5.28%**	**6.82%**	**4.61%**	**3.59%**
K = 4	5.23%	6.72%	4.46%	3.47%
δ = 0.25	5.24%	6.74%	4.47%	3.42%
δ = 0.50	**5.28%**	**6.82%**	**4.61%**	**3.59%**
δ = 0.75	5.21%	6.71%	4.42%	3.39%
L = 1	4.92%	6.38%	4.17%	3.31%
L = 2	5.19%	6.73%	4.43%	3.49%
L = 3	**5.28%**	**6.82%**	**4.61%**	**3.59%**
L = 4	5.23%	6.77%	4.49%	3.51%
m = 0.1	**5.28%**	**6.82%**	4.54%	3.55%
m = 0.2	5.21%	6.79%	**4.61%**	**3.59%**
m = 0.3	5.16%	6.71%	4.28%	3.24%
m = 0.4	5.08%	6.62%	4.10%	3.13%
λ = 0.0	4.99%	6.51%	4.35%	3.48%
λ = 0.01	5.20%	6.78%	4.40%	3.49%
λ = 0.1	**5.28%**	**6.82%**	4.51%	3.53%
λ = 1.0	5.12%	6.61%	**4.61%**	**3.59%**

tags is only 28. Such observation strongly indicates that the tag-enhanced aggregation module is more useful when the tags are neat and lack hierarchy.

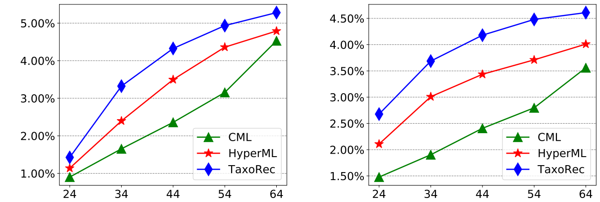
The performance gains of TaxoRec over Hyper + CML + Agg ranges from 0.56% (achieved in NDCG@20 on Ciao) to 7.38% (achieved in NDCG@20 on Yelp). The result shows that: (1) the explicitly taxonomy-aware regularization can further improve the performance in hyperbolic space, where hyperbolic space implicitly capture the latent hierarchies; (2) on the datasets (*e.g.*, Yelp) that have a larger number of tags and deeper hierarchies among tags, the improvement of taxonomy-aware regularization are more significant by properly arranging tag embedding according to the context and structure information in the taxonomy.

Compared with the models optimized in Euclidean space, the models in hyperbolic space leads to significant performance gain. For example, Hyper + CML in hyperbolic space outperforms CML in Euclidean space by up to 10.45% on Ciao, 27.25% on Amazon-CD, 5.74% on Amazon-Book, and 13.20% on Yelp. Moreover, even though CML + Agg has already integrated tag information through tag-enhanced aggregation mechanism in Euclidean space, Hyper + CML + Agg can still improve the performance by up to 14.44% on Ciao, 15.24% on Amazon-CD, 7.73% on Amazon-Book, and 11.39% on Yelp. Such results are consistent with those in Table II, showing the effectiveness of applying hyperbolic space.

D. Effect of Hyperparameters (RQ3)

Our proposed TaxoRec framework mainly introduces six hyperparameters, *i.e.*, K, δ, L, m, λ, and D.

From Table IV, we have the following observations: (1) K is used for splitting tag sets, where we found that the optimal K is about 3. (2) δ is used for selecting representative tags, where we found that the optimal δ is about 0.5. The rules for selecting K and δ could be the rule-of-thumb in practice across the used datasets. (3) L is the layer of GCN. TaxoRec achieves the best performance with $L = 3$. Since both Amazon-Book and Yelp have sparse interactions with 0.094% and 0.048% density, more neighbor aggregation can alleviate the data sparsity issue. When L continues to increase to 4, too many neighbors will lead over smoothing on the graph, and the performance decreases on all datasets. (4) m is the margin to enforce the difference between positive and negative triplets. The optimal m values on Amazon-Book and Yelp are about 0.1 and 0.2, respectively. In the range of [0.1, 0.2], the optimal m can be obtained by slight tuning, which is consistent with [44]. (5) λ controls the weight of the taxonomy-aware regularization, which aims to enforce the tag embeddings to be close to the weighted center of nodes in the taxonomy. Too small λ will cause the tag embeddings likely be spread out, while too large λ will likely cause the model to overfit. The optimal λ values on Amazon-Book and Yelp are about 0.1 and 1.0, respectively. Therefore, TaxoRec is reasonably sensitive to λ. In the range of [0.1, 1], the optimal λ can be obtained by slight tuning.



(a) Amazon-Book (b) Yelp

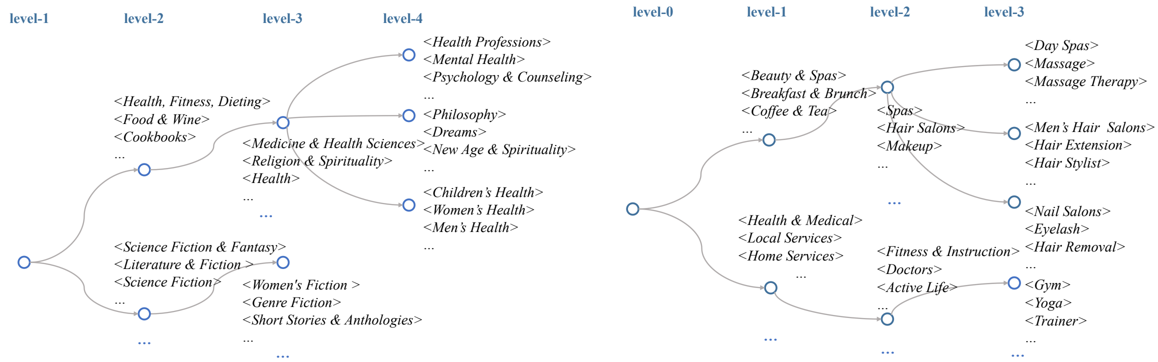
Fig. 5. Performance regarding Recall@10 of CML, HyperML and our proposed TaxoRec with varying dimension D on two of the datasets.

Furthermore, Fig. 5 shows the performance of CML, HyperML, and the proposed TaxoRec with varying settings of embedding dimension D. The total embedding dimension of three models are the same, where TaxoRec leave 12 dimensions for the tag-relevant embedding. Overall, we observe that all three models have performance gains when increasing D. Compared with CML that is in Euclidean space, HyperML and TaxoRec in hyperbolic metric space can achieve good results even when D is small. These results show the effectiveness of representation learning in hyperbolic space.

E. Tag Taxonomy Analysis (RQ4)

In this subsection, we demonstrate fine-grained taxonomies that we automatically learn on Amazon-Book and Yelp. As shown in Fig. 6(a), our proposed TaxoRec splits the tag set in level-1 into two fine-grained tag sets: (1) *Tag set 1:* {*<Health, Fitness, Dieting>*, *<Food & Wine>*, *<Cookbooks>*...}; (2) *Tag set 2:* {*<Science Fiction & Fantasy>*, *<Literature & Fiction>*, *<Science Fiction>*...}. In Fig. 6(b), we also show how TaxoRec splits the root in level-0 into the level-1 tag sets: (1) {*<Beauty & Spas>*, *<Breakfast & Brunch>*, *<Coffee & Tea>*...}; (2) *Tag set 2:* {*<Health & Medical>*, *<Local Services>*, *<Home Services>*...}.

Taking {*<Beauty & Spas>*, *<Breakfast & Brunch>*, *<Coffee & Tea>*...} in level-1 on Yelp as an example, we



(a) The taxonomy under level-1 on the Amazon-Book dataset. (b) The taxonomy under level-0 on the Yelp dataset.

Fig. 6. Parts of the tag taxonomies automatically constructed by the proposed TaxoRec.

TABLE V
EXAMPLES OF TAG-BASED USER PROFILES MODELED BY THE PROPOSED TAXOREC AND THE CORRESPONDING RECOMMENDATIONS.

	User	Tag	Item
AZ-Book	User1	*<Science & Mathematics>*; *<Technology>*; *<Software>*; *<Web Development & Design>*...	The Heart of Mathematics: An Invitation to Effective; How to Do Everything with JavaScript; ...
	User2	*<Health, Fitness, Dieting>*; *<Health>*; *<Mental Health>*; *<Exercise & Fitness>*...	Walking for Fitness; Perspectives in Nutrition; Elements of Yoga; Complete Weight Training Book...
Yelp	User3	*<Appliances & Repair>*; *<Auto Repair>*; *<Health & Medical>*; *<Fitness & Instruction>*...	Best Auto Repair; Kendal Green Service Center; Shamrock Appliance Repair; Mayfield Bodyworks Massage...
	User4	*<Food>*; *<Fast Food>*; *<Nightlife>*; *<Cocktail Bars>*...	Chick-fil-A; Virgils Gullah Kitchen & Bar; The Butterfly Bar; Lucky 13 Cocktail...

can observe that:

- at the level-2, TaxoRec can successfully find one of the major areas as: {*<Spas>*, *<Hair Salons>*, *<Makeup>*...};
- at the level-3, TaxoRec can further split the tag set of node in level-2 into: (1) {*<Day Spas>*, *<Massage>*, *<Massage Therapy>*...}; (2) {*<Men's Hair Salons>*, *<Hair Extension>*, *<Hair Stylist>*...}; (3) {*<Nail Salons>*, *<Eyelash>*, *<Hair Removal>*...}.

As we observe, the taxonomies that can be constructed from scratch are pretty accurate and highly interpretable, which can provide knowledge about the rich relations among tags. Specifically, the hierarchy for tags is constructed automatically and shows reasonable hypernym-hyponym relations among tags – these tags are semantically coherent and cover different aspects and expressions of the same parent tags.

F. Interpretable Case Studies (RQ5)

To provide more insights into the advantages of TaxoRec in providing interpretable recommendations, we demonstrate four random users with their closest tags retained by TaxoRec, and the corresponding items recommended by TaxoRec, on Amazon-Book and Yelp. Since the relations among users and tags can be measured through user-tag distances in the metric space, we obtain each user's top 4 tags by ranking the distances between the user to all tags, where the hyperbolic representations of users, items and tags are learned by TaxoRec.

From Table V, we observe that the tags retained for each user are highly coherent and form clear hierarchies, such as *<Technology>* $\rightarrow$ *<Software>* $\rightarrow$ *<Web Development & Design>* for User 1 on Amazon-Book, as well as *<Health & Medical>* $\rightarrow$ *<Fitness & Instruction>* for User 3 on Yelp. As a consequence, the items recommended to them are highly rational, such as How to Do Everything with JavaScript for User 1 on Amazon-Book, as well as Mayfield Bodyworks Massage for User 3 on Yelp.

Note that, such user tags, while directly extracted from the implicit feedback data in an unsupervised fashion, deliver rather valuable insights into meaningful and representative user types, which provides potential for more accurate user profiling and personalized recommendation in the future.

VI. CONCLUSION

In this paper, we propose to automatically construct an explicit tag taxonomy solely based on existing item tags and user-item interactions, which can effectively enhance recommendation from both accuracy and interpretability perspectives. Specifically, we propose a novel hyperbolic metric learning framework TaxoRec to simultaneously optimize a tag taxonomy and tag-enhanced representations for users and items, by leveraging the individual strengths of two hyperbolic models. Extensive experiments demonstrate the clear improvements of TaxoRec over the state-of-the-art baselines and insightful case studies show the accuracy and interpretability of our automatically constructed tag taxonomies.

In the future, it would be interesting to consider the incorporation and improvements of existing taxonomies when they are available, and the further application of fine-grained taxonomies and user-item-tag relations for tasks such as accurate user profiling and personalized recommendation.

VII. Acknowledgments

This work was supported in part by the National Key R&D Program of China (No. 2018YFB1403001), National Natural Science Foundation of China (No. 62172362), and the internal funding of Emory University.

References

[1] I. Bartolini and P. Ciaccia. Imagination: exploiting link analysis for accurate image annotation. In *International Workshop on Adaptive Multimedia Retrieval*, 2007.
[2] I. Bartolini and P. Ciaccia. Scenique: a multimodal image retrieval interface. In *AVI*, 2008.
[3] I. Bartolini, M. Patella, and G. Stromei. The windsurf library for the efficient retrieval of multimedia hierarchical data. In *SIGMAP*, 2011.
[4] S. Bonnabel. Stochastic gradient descent on riemannian manifolds. *IEEE Transactions on Automatic Control*, 2013.
[5] J. Bose, A. Smofsky, R. Liao, P. Panangaden, and W. Hamilton. Latent variable modelling with hyperbolic normalizing flows. In *ICML*, 2020.
[6] J. W. Cannon, W. J. Floyd, R. Kenyon, W. R. Parry, et al. Hyperbolic geometry. *Flavors of geometry*, 1997.
[7] P. Cao, Y. Chen, K. Liu, J. Zhao, S. Liu, and W. Chong. Hypercore: Hyperbolic and co-graph representation for automatic icd coding. In *ACL*, 2020.
[8] I. Chami, A. Gu, V. Chatziafratis, and C. Ré. From trees to continuous embeddings and back: Hyperbolic hierarchical clustering. In *NIPS*, 2020.
[9] I. Chami, A. Wolf, D.-C. Juan, F. Sala, S. Ravi, and C. Ré. Low-dimensional hyperbolic knowledge graph embeddings. In *ACL*, 2020.
[10] C. Chen, X. Zheng, Y. Wang, F. Hong, and D. Chen. Capturing semantic correlation for item recommendation in tagging systems. In *AAAI*, 2016.
[11] D. Downey, C. Bhagavatula, and Y. Yang. Efficient methods for inferring large sparse topic hierarchies. In *ACL*, 2015.
[12] S. Feng, L. V. Tran, G. Cong, L. Chen, J. Li, and F. Li. Hme: A hyperbolic metric embedding approach for next-poi recommendation. In *SIGIR*, 2020.
[13] O. Ganea, G. Bécigneul, and T. Hofmann. Hyperbolic entailment cones for learning hierarchical embeddings. In *ICML*, 2018.
[14] J. Gao, Y. He, Y. Wang, X. Wang, J. Wang, G. Peng, and X. Chu. Star: Spatio-temporal taxonomy-aware tag recommendation for citizen complaints. In *CIKM*, 2019.
[15] J. Gao, X. Wang, Y. Wang, and X. Xie. Explainable recommendation through attentive multi-view learning. In *AAAI*, 2019.
[16] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl. Neural message passing for quantum chemistry. In *ICML*, 2017.
[17] C. Gulcehre, M. Denil, M. Malinowski, A. Razavi, R. Pascanu, K. M. Hermann, P. Battaglia, V. Bapst, D. Raposo, A. Santoro, et al. Hyperbolic attention networks. In *ICLR*, 2018.
[18] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *SIGIR*, 2020.
[19] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua. Neural collaborative filtering. In *WWW*, 2017.
[20] Y. Hou, N. Yang, Y. Wu, and S. Y. Philip. Explainable recommendation with fusion of aspect information. *WWW*, 2019.
[21] C.-K. Hsieh, L. Yang, Y. Cui, T.-Y. Lin, S. Belongie, and D. Estrin. Collaborative metric learning. In *WWW*, 2017.
[22] J. Huang, Z. Ren, W. X. Zhao, G. He, J.-R. Wen, and D. Dong. Taxonomy-aware multi-hop reasoning networks for sequential recommendation. In *WSDM*, 2019.
[23] V. Khrulkov, L. Mirvakhabova, E. Ustinova, I. Oseledets, and V. Lempitsky. Hyperbolic image embeddings. In *CVPR*, 2020.
[24] W. Krichene and S. Rendle. On sampled metrics for item recommendation. In *KDD*, 2020.
[25] M. Law, R. Liao, J. Snell, and R. Zemel. Lorentzian distance learning for hyperbolic representations. In *ICML*, 2019.
[26] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 1999.
[27] M. Li, S. Zhang, F. Zhu, W. Qian, L. Zang, J. Han, and S. Hu. Symmetric metric learning with adaptive margin for recommendation. In *AAAI*, 2020.
[28] T. Li, Y. Ma, J. Xu, B. Stenger, C. Liu, and Y. Hirate. Deep heterogeneous autoencoders for collaborative filtering. In *ICDM*, 2018.
[29] W. Liu, J. Su, C. Chen, and X. Zheng. Leveraging distribution alignment via stein path for cross-domain cold-start recommendation. *NIPS*, 2021.
[30] Y. Liu, X. Xia, L. Chen, X. He, C. Yang, and Z. Zheng. Certifiable robustness to discrete adversarial perturbations for factorization machines. In *SIGIR*, 2020.
[31] W. Ma, M. Zhang, Y. Cao, W. Jin, C. Wang, Y. Liu, S. Ma, and X. Ren. Jointly learning explainable rules for recommendation with knowledge graph. In *WWW*, 2019.
[32] L. Mirvakhabova, E. Frolov, V. Khrulkov, I. Oseledets, and A. Tuzhilin. Performance of hyperbolic geometry models on top-n recommendation tasks. In *RecSys*, 2020.
[33] N. Monath, M. Zaheer, D. Silva, A. McCallum, and A. Ahmed. Gradient-based hierarchical clustering using continuous representations of trees in hyperbolic space. In *KDD*, 2019.
[34] M. Nickel and D. Kiela. Poincar\'e embeddings for learning hierarchical representations. In *NIPS*, 2017.
[35] M. Nickel and D. Kiela. Learning continuous hierarchies in the lorentz model of hyperbolic geometry. In *ICML*, 2018.
[36] C. Park, D. Kim, X. Xie, and H. Yu. Collaborative translational metric learning. In *ICDM*, 2018.
[37] J. Park, B. J. Hescott, and D. K. Slonim. Towards a more molecular taxonomy of disease. *Journal of biomedical semantics*, 2017.
[38] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *UAI*, 2009.
[39] F. Sala, C. De Sa, A. Gu, and C. Ré. Representation tradeoffs for hyperbolic embeddings. In *ICML*, 2018.
[40] R. Sarkar. Low distortion delaunay embedding of trees in hyperbolic plane. In *International Symposium on Graph Drawing*, 2011.
[41] J. Shang, X. Zhang, L. Liu, S. Li, and J. Han. Nettaxo: Automated topic taxonomy construction from text-rich network. In *WWW*, 2020.
[42] J. Shen, Z. Wu, D. Lei, C. Zhang, X. Ren, M. T. Vanni, B. M. Sadler, and J. Han. Hiexpan: Task-guided taxonomy construction by hierarchical tree expansion. In *KDD*, 2018.
[43] K. Song, F. Nie, J. Han, and X. Li. Parameter free large margin nearest neighbor for distance metric learning. In *AAAI*, 2017.
[44] J. Sun, Z. Cheng, S. Zuberi, F. Pérez, and M. Volkovs. Hgcf: Hyperbolic graph convolution networks for collaborative filtering. In *WWW*, 2021.
[45] C.-Y. Tai, M.-R. Wu, Y.-W. Chu, S.-Y. Chu, and L.-W. Ku. Mvin: Learning multiview items for recommendation. In *SIGIR*, 2020.
[46] Y. Tan, C. Yang, X. Wei, Y. Ma, and X. Zheng. Multi-facet recommender networks with spherical optimization. In *ICDE*, 2021.
[47] Y. Tay, L. Anh Tuan, and S. C. Hui. Latent relational metric learning via memory-based attention for collaborative ranking. In *WWW*, 2018.
[48] 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 *WSDM*, 2020.
[49] X. Wang, X. He, Y. Cao, M. Liu, and T.-S. Chua. Kgat: Knowledge graph attention network for recommendation. In *KDD*, 2019.
[50] X. Wang, X. He, M. Wang, F. Feng, and T.-S. Chua. Neural graph collaborative filtering. In *SIGIR*, 2019.
[51] L. Wu, Y. Yang, K. Zhang, R. Hong, Y. Fu, and M. Wang. Joint item recommendation and attribute inference: An adaptive graph convolutional network approach. In *SIGIR*, 2020.
[52] C. Yang, A. Pal, A. Zhai, N. Pancha, J. Han, C. Rosenberg, and J. Leskovec. Multisage: Empowering gcn with contextualized multi-embeddings on web-scale multipartite networks. In *KDD*, 2020.
[53] C. Yang, J. Zhang, and J. Han. Co-embedding network nodes and hierarchical labels with taxonomy based generative adversarial networks. In *ICDM*, 2020.
[54] C. Zhang and J. Gao. Hype-han: Hyperbolic hierarchical attention network for semantic embedding. In *IJCAI*, 2020.
[55] C. Zhang, F. Tao, X. Chen, J. Shen, M. Jiang, B. Sadler, M. Vanni, and J. Han. Taxogen: Unsupervised topic taxonomy construction by adaptive term embedding and clustering. In *KDD*, 2018.
[56] K. Zhang, H. Qian, Q. Cui, Q. Liu, L. Li, J. Zhou, J. Ma, and E. Chen. Multi-interactive attention network for fine-grained feature learning in ctr prediction. In *WSDM*, 2021.
[57] Y. Zhang, A. Ahmed, V. Josifovski, and A. Smola. Taxonomy discovery for personalized recommendation. In *WSDM*, 2014.
[58] X. Zheng, M. Wang, C. Chen, Y. Wang, and Z. Cheng. Explore: Explainable item-tag co-recommendation. *Information Sciences*, 2019.
[59] C.-N. Ziegler, G. Lausen, and L. Schmidt-Thieme. Taxonomy-driven computation of product recommendations. In *CIKM*, 2004.