Finding High-quality Item Attributes for Recommendation

Xiaolin Zheng *Senior Member, IEEE*, Yanchao Tan, Yan Wang *Senior Member, IEEE*, Xiangyu Wei, Shengjia Zhang, Chaochao Chen, Longfei Li, Carl Yang *Member, IEEE*

Abstract—The sparse interactions between users and items on the web have aggravated the difficulty of their representations in recommender systems. Existing approaches leverage item attributes (e.g., item categories and tags) to alleviate the data sparsity problem, so as to enhance the performance and interpretability of recommendation. However, directly using all attributes of items cannot avoid the negative impacts of low-quality attributes, where manually labeling the quality of attributes is time-consuming. To this end, we propose HQRec to jointly measure the quality of attributes automatically and perform recommendation accurately. Specifically, we first analyze the different qualities among item attributes, and propose to leverage item categories to select high-quality tags via category-guided quality measurement and direction-aware optimization in an unsupervised fashion. Then, we propose to capture the complex relations among users and items based on the high-quality attributes, where a novel quality-aware embedding fusion and quality-aware embedding propagation mechanism for users and items is devised. Extensive experiments on four real-world benchmark datasets show drastic performance gains brought by our proposed HQRec framework, which constantly achieves an average of 14.73% improvement over the state-of-the-art baselines in terms of Recall and NDCG metrics. Insightful case studies also show that our automatic quality measurements are highly accurate and interpretable.

Index Terms—Recommender system, attributed graph, graph representation learning, metric learning

1 INTRODUCTION

Tags and item categories have been used as *item attributes* in recommendation systems (shown in Fig. 1(a)) to characterize items, profile users, and alleviate the sparsity problem. Among them, tags are annotated by users, which are large in volume and rich in semantics. However, there exists a lot of noise (i.e., low-quality attributes) in tags. Compared with tags, item categories are less in amount, but they can usually provide cleaner semantics (i.e., high-quality attributes) due to their always-standardized format. The below Example 1 is introduced to illustrate the qualities of attributes.

Example 1. As shown in Fig. 1(a), based on Linda's interactions with STEAK, "*#Beef Wellington*", and "*#Ribeye steak*", and Jack's interactions with SEAFOOD and "*#Shrimp*", we can distinguish the preferences of Linda and Jack as steak and seafood. However, it is hard to do so based on "*#Eat here*" of Linda and '*#Your plate*" of Jack. Therefore, attributes like STEAK and "*#Beef Wellington*" are high-quality attributes to distinguish users and items, while "*#Eat here*" and "*#Your plate*" are examples of low-quality attributes.

- *X. Zheng, X. Wei, S. Zhang, and C. Chen are with the College of Computer Science, Zhejiang University, Hangzhou 310027, China, Email:{xlzheng, weixy, zhsj, zjuccc}@zju.edu.cn.*
- *Y. Tan is with the College of Computer and Data Science, and Fujian Key Laboratory of Network Computing and Intelligent Information Processing, Fuzhou University, Fuzhou 350116, China, Email: yctan@fzu.edu.cn.*
- *Y. Wang is with School of Computing, Macquarie University, Sydney, NSW 2109, Australia, Email: yan.wang@mq.edu.au.*
- *L. Li is with Ant Financial Services Group, Hangzhou 310058, China, E-mail: longyao.llf@antgroup.com.*
- *C. Yang (Corresponding Author) is with the Department of Computer Science, Emory University, Atlanta 30322, United State, E-mail: j.carlyang@emory.edu.*

Existing attribute-based solutions mainly leverage either tags only [4], [43], [44] or item categories only [6], [40]. They manifest drawbacks as follows: (i) models that use tags only are easily affected by the low-quality tags; (ii) a model that leverages item categories only misses rich semantics from tags and fails to provide finer profiles of items and users, which are illustrated by Example 2.

Example 2. As shown in Fig. 1(b), when profiling users with tags only, tags such as "*#Eat here*" and "*#Your plate*" fail to reflect users' preferences directly. Moreover, the proportion of low-quality tags exceeding meaningful tags can impede the modeling of users and items. When profiling users with item categories only (e.g., STEAK and SEAFOOD), they fail to distinguish Linda from Lisa and Ethan, where Linda likes "*#Ribeye steak*" in addition to "*#Beef Wellington*" while Lisa and Ethan only like "*#Beef Wellington*".

With only tags or item categories, it is difficult to accurately profile users and characterize items based on both clean and rich semantics attributes. Although prevalent approaches leverage both types of attributes based on attributed graphs, they cannot eliminate the influence of low-quality attributes when training the model. Moreover, since the neighbor aggregation and convolution layer are widely used in graph learning, low-quality attributes can easily affect the representation of users and items via high-order relation propagation. For example, as shown by the red dotted lines in Fig. 1(a), after two hops of propagation (i.e., tag-item-user), "*#Your plate*" that indirectly profiles Lisa will be also involved in the embedding learning of Lisa. In this way, the negative impacts of low-quality tags are exaggerated through the graph convolutions.

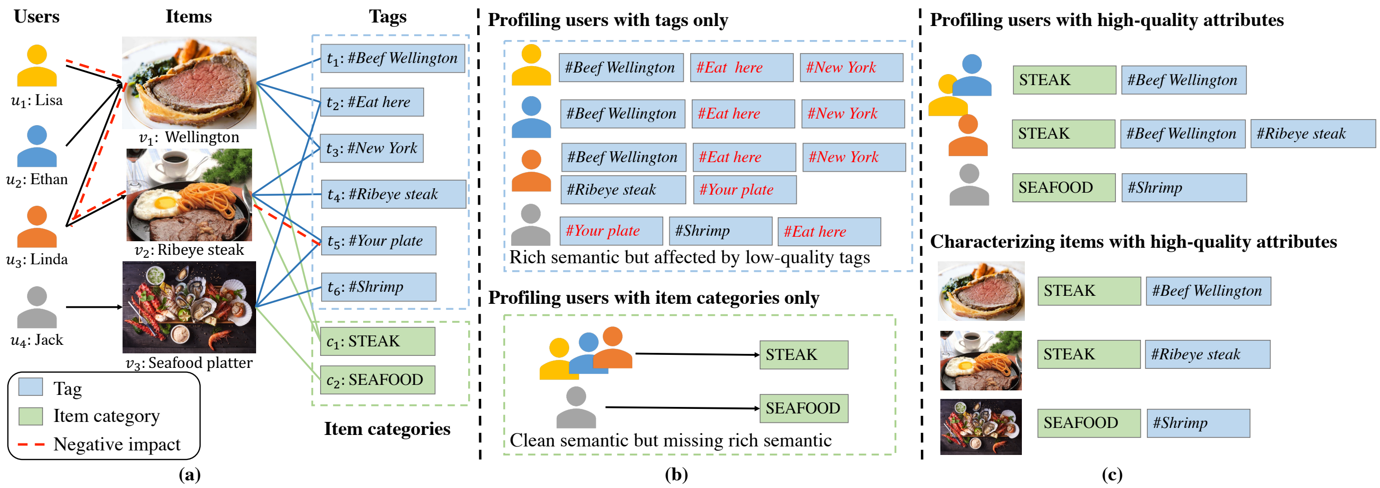


Fig. 1. A toy example motivating example of leveraging both item categories and tags for profiling users and characterizing items. (a) shows an attributed graph network that includes users, items, item categories, and tags. (b) Profiling users with tags only or item categories only. (c) Profiling users and characterizing items with high-quality attributes, which can model users with both clean and rich semantics.

Therefore, it is in high demand to select and leverage high-quality attributes from both item categories and tags for improving the performance of recommendation. As shown in Fig. 1(c), with both clean and rich semantics, we can reduce the influence of noisy attributes and leverage the high-quality attributes to improve the performance of recommendation. Despite these benefits, modeling users and items with high-quality attributes, however, has the following open challenges.

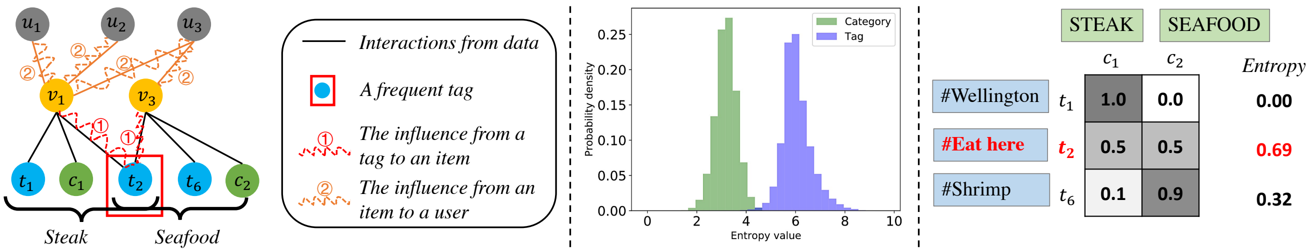
Challenge I: *How to automatically select high-quality item attributes for more accurate recommendation?* A straightforward way is to manually label high-quality attributes. However, such labeling is time-consuming due to a large number of attributes and the significant differences among attributes in different datasets. For example, the number of tags in the ML20M dataset is 8,534. Moreover, the item categories and tags in the NYC-R dataset are relative to food while those in the ML20M dataset are relative to movies. Without any contexts of item categories and tags, most existing works filter attributes in the preprocessing stage directly according to the occurrence frequency of an attribute [19]. However, such methods cannot filter the attributes that show up frequently but are not helpful for distinguishing users and items. Therefore, they fail to assure the quality of selected attributes. Accordingly, we need to exploit an automated quality measurement mechanism for attributes, which can adapt to any datasets and help to profile users and characterize items in recommender systems.

Challenge II: *How to improve the performance of item recommendation based on the high-quality attributes?* Existing attributed graph models for recommendation have provided strong performance via performing the graph convolution for high-order interactions [18], [38]. However, it is better to perform recommendation only under high-quality attributes rather than aggregating all attributes indiscriminately. In this case, the model can avoid the negative impact of noisy information from high-order neighbors and make use of its learning capacity. Since the existing attributed graph models do not explicitly consider the qualities of attributes, there are still open problems on how to improve recommendation performance based on high-quality attributes only.

To address the above challenges, in this paper, we propose HQRec, which is a framework that can automatically measure and select high-quality attributes for improving the performance of recommendation. The framework contains two components: (i) *Quality measurement module*, which is introduced to obtain quality scores by measuring the distances between item categories and tags based on category-guided metric learning with direction-aware optimization. After calculating the distance between a tag and the categories of one item, we can automatically quantify the quality of attribute under the item and address **Challenge I**. (ii) *Recommendation module*, which includes quality-aware embedding fusion layer and quality-aware embedding propagation layer based on the output of quality measurement module. By leveraging the learned quality scores of attributes as interpretable weights, we are able to jointly train the quality measurement module and recommendation module to accurately profile all items and users and address the problem in **Challenge II**.

Our contributions are summarized as follows:

- *Analysis:* We make a detailed analysis of how different quality of attributes influence models from empirical analysis. This analysis helps us design a proper quality measurement mechanism to achieve better leverage of item categories and tags (Section 3).
- *Model:* We propose HQRec, a novel model for recommendation based on high-quality attributes. In the quality measurement module, we propose *category-guided metric learning* and *direction-aware optimization* to automatically rank the quality of attributes without supervision. In the recommendation module, we introduce *quality-aware embedding fusion* and *quality-aware embedding propagation* based on the weighted high-quality attributes (Section 4).
- *Experiments:* We conduct extensive experiments on two real-world datasets, which demonstrate significant improvements of the proposed HQRec framework on recommendation together with highly accurate and interpretable attribute selection results (Section 5).



(a) The influence of a frequent tag t_2

(b) The histogram of entropy values of user-category and user-tag on NYC-R

(c) The entropy values of tag-category via co-occurred of categories and tags.

Fig. 2. The differences between item categories and tags. (a) shows that a frequent tag "*#Eat here*" fails to distinguish users. (b) shows the entropy distributions of categories and tags over users. A low entropy value means the user's preference is consistent over the selective attributes. (c) shows that "*#Eat here*" with a high entropy value is an example of a low-quality tag.

2 Related Work

2.1 Recommender Systems

Collaborative filtering (CF) has been widely used in recommender systems due to its relatively high performance with easily collected data [10], [20], [33]. In general recommendation, matrix factorization (MF) has become the *de facto* method, which uses inner products to model the similarity of user-item pairs [21], [28]. Recently, metric learning for recommendations has attracted significant research attention [30], [32]. Existing methods in this line seek appropriate distance functions for input points instead of inner products, which can address the limitations of MF. Based on the Euclidean distance, [10] 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 similarity. Since CML has a one-to-many mapping problem which limits the representation of users and items, [24] turned this problem to multiple one-to-one mappings and [35] turned this problem to one-to-one mappings between Euclidean and hyperbolic spaces. Considering that CML has a geometrically restrictive scoring function and it has been proven to be an ill-posed algebraic system, [34] learned latent user-item interaction relations based on memory network and attention mechanism, which helps to alleviate the potential geometric problem. However, CF based models suffer from the cold-start problem and could not perform well when users have limited records [25].

To alleviate the data sparsity problem, many studies have incorporated auxiliary information, e.g., user attributes [3], [11], item attributes [23], [42], social network [17], [29], and so on. Among them, attribute enhanced CF methods are widely studied as attributes can be easily collected and used for recommendation [42]. Factorization machine modeled pairwise interactions between all features and was a generalized model since they can mimic most factorization models with feature engineering [21], [27].

All these feature-enhanced CF models do not consider the quality of attributes. However, attributes can be noisy and low-quality, which may even harm the performance after leveraging these attributes. Instead of removing low-quality attributes manually, we design a model that learns the quality of attributes in an unsupervised fashion.

2.2 Graph Convolutional Neural Networks

Recently, GCNs have shown huge success for graph representation learning and related applications [5], [45]. As the user-item behavior could be naturally regarded as a graph structure, researchers have proposed graph based recommendation models for better user and item embedding learning [7], [39]. For example, LightGCN [7] proposed simplified linear graph convolution operations and residual learning between different layers, which is a state-of-the-art content based GCN model for item recommendation. EGLN [39] made the enhanced graph learning module and the node embedding module iteratively learn from each other via mutual information maximization. Furthermore, it is common to incorporate additional attributes to alleviate the data sparsity issue faced by graph learning models. For example, AGCN [38] integrated linear graph convolution operations with attribute inference based on incomplete user/item attributes. KGCN [11] incorporated social ties into the collaborative filtering architecture as side information to characterize connectivity information across users. AGNN [26] designed an attribute graph based on user profiles and item features, which produced the preference embedding for a cold user/item.

Although the above GCN-based recommendation models can learn weights to choose the important features via weighted sum and attention operation, they cannot filter out the frequent attributes that are not helpful for distinguishing users and items. Therefore, a model that can assure the quality of selected attributes and learn interpretable weights for features is in great need.

3 Preliminary Study

To investigate different qualities among categories[1] and tags, we conduct empirical studies on a real-world dataset NYC-R (dataset statistics can be found in Section 5). Furthermore, we investigate how to select high-quality attributes (e.g., categories and tags).

Although existing studies often select categories and tags that are frequently used by users for improving the performance of recommendations, they fail to filter the frequent attributes that are not helpful for distinguishing users and items, which can make users with different preferences close

1. The category in the following section of our paper typically corresponds to item categories.

to each other. As shown in Fig. 2(a), the tag "#Eat here" is frequently used by all u_1, u_2, and u_3. However, when profiling users with "#Eat here", it fails to distinguish u_3 who likes SEAFOOD from u_1 and u_2 who like STEAK, and thus treats all three users with the same preference. Therefore, it is important to find a metric other than frequency to select high-quality attributes for profiling users.

Since the metric of entropy [16] can represent the data distribution [1], we propose to calculate the entropy of category and tag from an individual user's perspective. The entropy is calculated as follows:

$$H(\boldsymbol{E}) = -\sum_{i=1}^{n} P_i \log P_i, \tag{1}$$

where $\boldsymbol{E} = \{\boldsymbol{E}_1, \ldots, \boldsymbol{E}_i, \ldots, \boldsymbol{E}_n\}$ denotes the set of n possible events. The probabilities of these events are denoted as $\{\boldsymbol{P}_1, \ldots, \boldsymbol{P}_i, \ldots, \boldsymbol{P}_n\}$.

Fig. 2(b) shows histograms of the entropy value via user-category and user-tag matrices. Take category for example, the entropy value of user u's category is calculated via $H(\boldsymbol{E}_c^u) = -\sum_{i=1}^{N_c} \boldsymbol{P}_{c,i}^u \log \boldsymbol{P}_{c,i}^u$, where $\boldsymbol{P}_{c,i}^u$ is the normalized probability of interacting the i-th category (i.e., $\boldsymbol{P}_{c,i}^u = \frac{\text{the number of interaction to category } i}{\text{the number of interacted categories}}$). $N^{\mathbf{c}}$ is the number of categories. A low entropy value of the user means the user's preference is consistent over the selective attributes.

Note that, the user's distribution over categories is more concentrated than that over tags, where most categories have lower-value entropies than tags (e.g., 94.01% of the categories and 0.06% of the tags have entropy values lower than 4 in Fig. 2(b)). Since categories are more discriminative for distinguishing users with different preferences, we have an intuitive finding that *the qualities of categories are higher than tags*.

However, a model that leverages categories only will miss rich semantics from tags and fails to provide finer profiles of items and users. Therefore, it is also needed to leverage high-quality tags besides categories. Since both categories and tags are attributes and categories have high qualities, we then define how to obtain the entropy value of tags by categories. A lower entropy value here means a tag's distribution over categories is concentrated and can serve as a fine-grained description of categories. As shown in Fig. 2(c), given the co-occurrence frequency of tags and categories on items, the entropy of "*#Ribeye steak*" is 0 and the entropy of "*#Eat here*" is 0.69. The low entropy of "*#Beef Wellington*" shows that "*#Beef Wellington*" is closely related to the category STEAK. As mentioned in Example 1, "*#Beef Wellington*" is helpful to distinguish user's preferences and can serve as a fine-grained attribute of the clean semantic STEAK. Therefore, "*#Beef Wellington*" can be regarded as a high-quality attribute. However, "*#Eat here*" with a high entropy value fails to differentiate the relations between STEAK and SEAFOOD, which is a low-quality attribute.

In summary, categories are more consistently used by users and thus more helpful to profile users, so we consider that the qualities of categories are higher than tags. Among a large number of tags, the tags with low entropy over categories can serve as the fine-grained attributes to categories, which can be regarded as high-quality attributes as well.

TABLE 1
Notations

Notation	Description
$\boldsymbol{U}, \boldsymbol{V}$	the embeddings of users and items
$\boldsymbol{C}, \boldsymbol{T}$	the embeddings of categories and tags
M, N, $N^{\mathbf{c}}$, $N^{\mathbf{t}}$	the number of users, items, categories, and tags
D, $D^{\mathbf{a}}$	the dimensions for user/item and attributes
$\mathbf{X}, \hat{\mathbf{X}}$	the original and predicted user-item matrix
$\mathbf{X}^{\mathbf{Ca}}$, $\mathbf{X}^{\mathbf{Ta}}$	the item-category matrix and the item-tag matrix
$\boldsymbol{W}$	the learned quality score for tags

4 Our Framework

In this section, we would introduce our proposed HQRec framework for joint quality assurance and recommendation. We first introduce the problem statement and overall architecture of the proposed model, followed by the detailed model optimization process.

4.1 Problem Statement

We use $\boldsymbol{u} \in \boldsymbol{U}$, $\boldsymbol{v} \in \boldsymbol{V}$, $\boldsymbol{c} \in \boldsymbol{C}$, and $\boldsymbol{t} \in \boldsymbol{T}$ to denote the embeddings of user u, item v, category c, and tag t, where $\boldsymbol{U} \in \mathbb{R}^{D \times M}$, $\boldsymbol{V} \in \mathbb{R}^{D \times N}$, $\boldsymbol{C} \in \mathbb{R}^{D^{\mathbf{a}} \times N^{\mathbf{c}}}$, and $\boldsymbol{T} \in \mathbb{R}^{D^{\mathbf{a}} \times N}$. D and $D^{\mathbf{a}}$ are the dimensions. M, N, $N^{\mathbf{c}}$, and $N^{\mathbf{t}}$ are the number of users, items, categories, and tags. $\mathbf{X} \in \mathbb{R}^{M \times N}$ denotes the implicit feedback matrix between users and items. $\mathbf{X}^{\mathbf{Ca}} \in \mathbb{R}^{N \times N^{\mathbf{c}}}$ and $\mathbf{X}^{\mathbf{Ta}} \in \mathbb{R}^{N \times N^{\mathbf{t}}}$ denote the item-category matrix and the item-tag matrix. $\mathbf{X}$, $\mathbf{X}^{\mathbf{Ca}}$, and $\mathbf{X}^{\mathbf{Ta}}$ are obtained from the original data.

In the real world, attributes have different qualities. We first select high-quality tags via ranking the learned quality score $\boldsymbol{W} \in \mathbb{R}^{N \times N^{\mathbf{t}}}$, and then leverage high-quality attributes for better recommendation via the predicted preference $\hat{\mathbf{X}}$. The input and output are defined as follows:

- *Quality Measurement Module:* The goal of this module is to leverage categories to select the high-quality tag t_i for item v according to the quality score $W_{i,v}$. The higher value of score $W_{i,v}$, the higher quality of tag t_i in item v.
- *Recommendation Module:* The goal of this module is to predict users' preferences to all items as: $\hat{\mathbf{X}} = g([\mathbf{X}, \boldsymbol{W}, \boldsymbol{U}, \boldsymbol{V}, \boldsymbol{C}, \boldsymbol{T}, \mathbf{X}^{\mathbf{Ca}}, \mathbf{X}^{\mathbf{Ta}}])$, where $\hat{\mathbf{X}} \in \mathbb{R}^{M \times N}$ denotes the predicted rating matrix.

4.2 HQRec Overview

We summarize the main components of the framework HQRec in Fig. 3 to provide an overview. The inputs of HQRec are interactions of user-item, item-category, and item-tag matrix. We jointly perform quality measurement for attributes and item recommendation as follows:.

- In *quality measurement module*, inspired by the empirical and theoretical analysis in Section 3, we propose to leverage categories to automatically measure the quality of tags via proposed category-guided metric learning and direction-aware optimization.
- In *recommendation module*, HQRec presents a quality-aware embedding fusion layer to integrate high-quality attributes to the representation of items. The aggregated attribute representations are further taken as an input of the proposed quality-aware embedding propagation layer to obtain high-order relations among users and items.

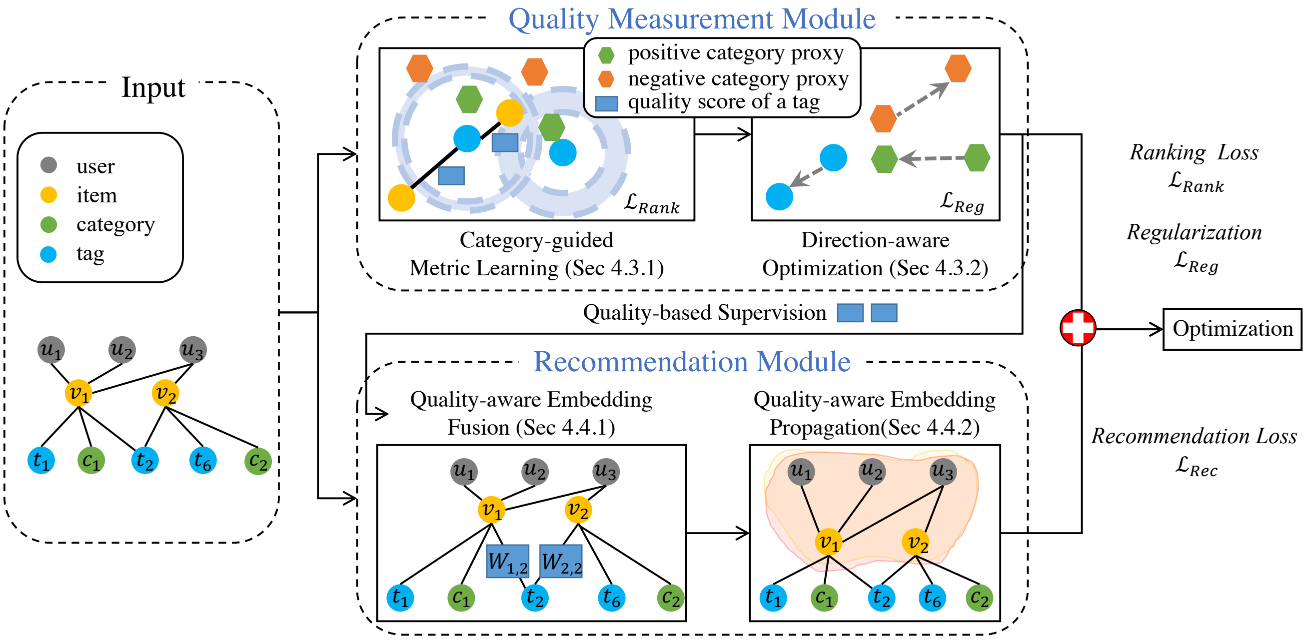


Fig. 3. The overall design of our proposed HQRec framework.

After the joint training of quality measurement module and recommendation module, the learned user and item embeddings are used for recommendation.

4.3 Quality Measurement Module

Different attributes have different qualities, where high-quality attributes are helpful by providing clean and rich semantics while low-quality attributes bring about noises. Therefore, directly leveraging all of them cannot improve the performance of recommendation. As shown in Fig. 2(c), the attribute in the red rectangle can influence both items and users at the same time. To select the high-quality attributes for recommendation, a straightforward way is to manually label high-quality attributes from both categories and tags. However, manual inspection and correction are labor-intensive and hence scale poorly to large datasets. Though it is widely used to filter attributes directly according to their occurrence frequency in the pre-processing stage, these methods fails to filter the frequent attributes that are not helpful for distinguishing users and items (e.g., "*#Eat here*" in Fig. 2(b)).

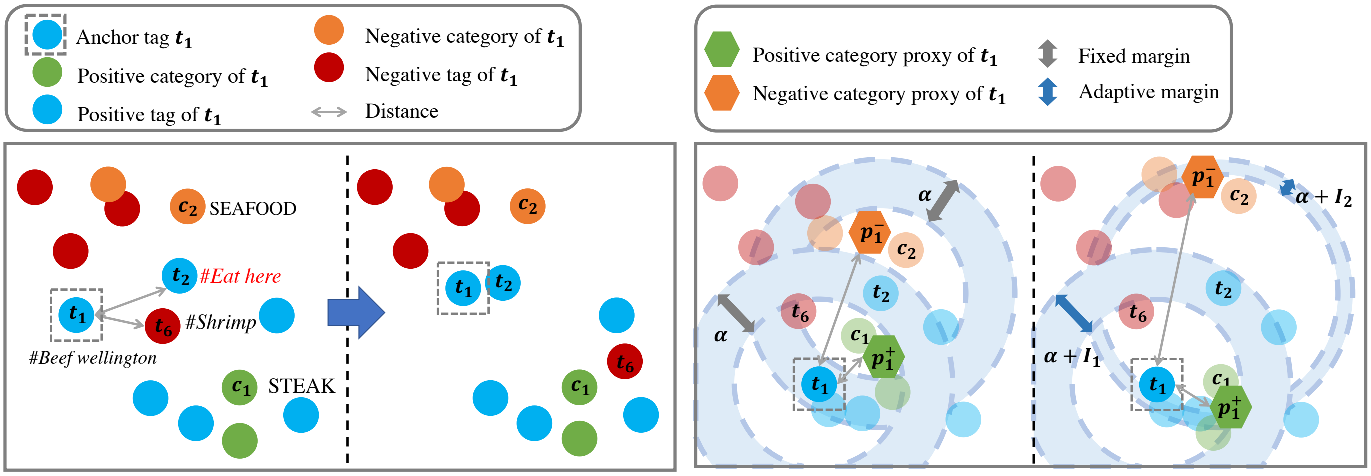
To address the above problem, we propose to leverage categories to automatically select high-quality attributes according to the observations in Section 3. Specifically, in this section, we introduce category-guided metric learning and direction-aware optimization.

4.3.1 Category-guided Metric Learning

Since the quality of most categories are better than that of tags, we can directly leverage all categories according to $\mathbf{X}^{Ca}$ without selection. However, only leveraging the information bring by categories is not enough, since it fails to distinguish users and items via rich semantics of tags. As shown in Fig. 1(b) and Example 2, only leveraging the category STEAK fails to distinguish Linda from Lisa and Ethan, where Linda likes "*#Ribeye steak*" besides "*#Beef Wellington*" while Lisa and Ethan only like "*#Beef Wellington*". Therefore, it is important to make good use of the information of tags in addition to categories.

To properly leverage tags, we propose to select high-quality tags to ensure that we only bring in rich semantic tags instead of noisy ones. Particularly, as shown in Fig. 1, we only hope to adopt high-quality tags like "*#Beef Wellington*" and "*#Ribeye steak*" instead of low-quality tags like "*#Eat here*" and "*#New York*". Based on the intuition that the co-occurring tags should be closer to each other than the not co-occurring ones, a straightforward way is to leverage the relations among tags and perform a tag-guided metric learning to select high-quality tags. However, such a solution fails to distinguish low-quality attributes. Moreover, it is easy to get negative impacts from low-quality attributes. For example, due to the co-occurred of t_1 "*#Beef Wellington*" and t_2 "*#Eat here*" on item v_1 (shown in Fig. 1), the distance between t_1 and t_2 should be closer to the one between t_1 and t_6 "*#Shrimp*". However, as shown in Fig. 4(a), the tag-guided metric learning misleads "*#Beef Wellington*" to stay close to SEAFOOD and "*#Shrimp*" to move close to STEAK, leading to an opposite modeling on the semantics of attributes. Therefore, selecting high-quality tags is challenging without the supervision of exact labels.

Inspired by the analysis that the qualities of most categories are better than those of tags (cf. Section 2), we propose to project categories and tags into one metric space and provide a mechanism for learning the qualities of tags under the guidance of high-quality categories. In this way, tags with clean semantics can be simultaneously close to their co-occurred categories (corresponding to the high-quality tags), while tags with mixed semantics can be far away from their co-occurred categories (corresponding to the low-quality tags), which effectively select high-quality tags in an unsupervised fashion. As shown in Fig. 2(b), "*#Eat here*" will stay in the middle of STEAK and SEAFOOD while



(a) Tag-guided metric learning

(b) Category-guided metric learning with fixed vs. adaptive margin

Fig. 4. An illustration of tag-guided metric learning vs. category-guided metric learning. In (a), the low-quality tag t_2 "*#Eat here*" negatively influences the modeling of categories and tags, which misleads t_1 "*#Beef Wellington*" to stay closer to c_2 SEAFOOD and t_6 "*#Shrimp*" to stay closer to c_1 STEAK. In the left handside of (b), an anchor tag is guided with the proxies that are generated by categories rather than tags, which can force t_1 to stay closer to c_2 instead of c_1 and thus overcome the limitation in (a). Furthermore, with adaptive margins, we can obtain a better metric space by allowing categories with different semantics to stay away from each other.

"*#Beef Wellington*" can be close to STEAK. Moreover, since it is unclear about the exact relations between the tags and the categories of one item, we define a category proxy as the mean pooling of the sampled categories. To make the model learning more robust, we sample K co-occur categories of the tag t_i to form a positive category proxy $\boldsymbol{p}_i^+$ and K not co-occurring categories to form the negative category proxy $\boldsymbol{p}_i^-$. Specifically, $\boldsymbol{p}_i^+ = \frac{1}{K}\sum_{k=1}^{K} \boldsymbol{c}_{j,k}$ denotes the average embedding of the sampled K categories, where the k-th sampled category $c_{j,k}$ is selected according to the co-occurring probability $\frac{\boldsymbol{\Phi}_{i,j}}{\sum_{z=1}^{N^{\mathrm{c}}} \boldsymbol{\Phi}_{i,z}}$. The tag-category matrix $\boldsymbol{\Phi}$ is obtained via multiplying tag-item matrix (i.e., $(\mathbf{X}^{\mathbf{Ta}})^T$) and item-category matrix (i.e., $\mathbf{X}^{\mathbf{Ca}}$). The higher $\boldsymbol{\Phi}_{i,j}$ value means that t_i co-occurs more times with c_j. If two categories have the same number of co-occurrence times with t_i, one of them is randomly selected. $\boldsymbol{p}_i^-$ denotes the average embedding of the K sampled categories that do not co-occur with tag t_i, which are randomly sampled from the categories with $\boldsymbol{\Phi}_{i,j} = 0$.

Thereby, category-guided metric learning can be formulated as follows:

$$\mathcal{L}_{triplet} = \sum_{t_i \in \mathcal{T}} \|\boldsymbol{t}_i - \boldsymbol{p}_i^+\|^2 - \|\boldsymbol{t}_i - \boldsymbol{p}_i^-\|^2 + \alpha, \tag{2}$$

where α is the tag margin to enforce the difference between positive and negative triplets. $\boldsymbol{t}_i$ denotes the embedding of tag t_i that belongs to tag set $\mathcal{T}$.

Note that, in Eq. 2, margin hyperparameter α is fixed for all tags. We find it largely limits the flexibility of arranging the tags with different qualities in the metric space, thus hindering the modeling of relations between tags and categories. Particularly, as shown in the left handside of Fig. 4(b), based on the relations between t_1 and t_2 towards the sampled proxy p_1^- and p_1^+, a fixed margin α makes it impossible to keep distance between p_1^- and p_1^+. In consequence, it is hard to distinguish STEAK and SEAFOOD. However, if we allow tags to have adaptive margins like $\alpha + I_1$ and $\alpha + I_2$ in Fig. 4(b), the model has more freedom to arrange the metric spaces, which allows the categories with different semantics to stay away from each other.

Inspired by the observation in Section 3(a), we associate the margin α in Eq. 2 with a clear physical meaning, which is related to the entropy of each tag. Specifically, tags have a lower entropy value, meaning the tag is more likely to interact with dedicated categories while others are not. Therefore, a tag with a low entropy value can have a larger margin than the one with a large value of entropy. Based on this intuition about the negative correlation between the entropy of tags and the margins, we propose to directly compute the entropy of tags from the given data:

$$E_{i,j} = \frac{\boldsymbol{\Phi}_{i,j}}{\sum_{z=1}^{N^{\mathrm{c}}} \boldsymbol{\Phi}_{i,z}}, \tag{3}$$

where $\boldsymbol{\Phi}$ corresponds to tag-category matrix and is obtained via multiplying tag-item matrix $(\mathbf{X}^{\mathbf{Ta}})^T$ and item-category matrix $\mathbf{X}^{\mathbf{Ca}}$. Then, we adaptively set the margins of tags:

$$I_i = 1 - \frac{1}{\log N^{\mathrm{c}}} \sum_{j=1}^{N^{\mathrm{c}}} E_{i,j} \log E_{i,j}, \tag{4}$$

where N^{c} denotes the number of categories and $I_i \in [0, 1]$. The idea behind Eq. 4 is to leverage the one-hop neighbors of t_i on the bipartite tag-category graph to represent its information level, i.e., the more diverse one-hop neighbor t_i, the more unstable connections between t_i and specific categories. In this case, t_i does not need to maintain a large margin between positive and negative categories. With such adaptive margins, we rewrite Eq. 2 as follows:

$$\mathcal{L}_{Rank} = \sum_{t_i \in \mathcal{T}} \|\boldsymbol{t}_i - \boldsymbol{p}_i^+\|^2 - \|\boldsymbol{t}_i - \boldsymbol{p}_i^-\|^2 + (\alpha + I_i), \tag{5}$$

Finally, based on the learned embeddings of categories and tags, we can quantify the quality of a tag so as to provide interpretable weights for tags to be aggregated.

Specifically, we calculate a quality score of the tag t_i under item v according to its similarity across categories of the item as follows

$$\boldsymbol{W}_{v,i} = e^{-\|\boldsymbol{t}_i - \boldsymbol{p}_v^{\cup}\|^2}, \tag{6}$$

where $\boldsymbol{p}_v^{\cup} = \frac{1}{\|\mathbf{X}_v^{\mathbf{Ca}}\|}\mathbf{X}_v^{\mathbf{Ca}}\boldsymbol{C}$ denotes the union category proxy by aggregating all the categories of item v.

4.3.2 Direction-aware Optimization

Our proposed CQM module adaptively pushes the negative categories radially outward with respect to the anchor tag as illustrated in Fig. 4(b). In Eq. 5, though the anchor tag is shifted away from the negative proxy of categories and attempts to move closer to the positive proxy of categories, the method does not account for the fact that the positive proxy locates close to the negative proxy. For example, as shown in the dotted gray line in the left of Fig. 5(a), the distance between positive and negative proxies is small, where the angle between $<t_2, p_2^+>$ (i.e., $\boldsymbol{t}_2\boldsymbol{p}_2^+$) and $<t_2, p_2^->$ (i.e., $\boldsymbol{t}_2\boldsymbol{p}_2^-$) is acute. In this case, it is hard to keep moving close to the positive proxy while pushing the negative proxy away from the anchor tag, where the distance of $\tilde{\boldsymbol{t}}_2\tilde{\boldsymbol{p}}_2^-$ after moving is almost the same as $\boldsymbol{t}_2\boldsymbol{p}_2^-$. However, compared with $<t_2, p_2^+, p_2^->$, $<t_4, p_4^+, p_4^->$ with obtuse angle between $\boldsymbol{t}_4\boldsymbol{p}_4^+$ and $\boldsymbol{t}_4\boldsymbol{p}_4^-$ can better separate the positive and negative proxies, which make more contributions to obtain even more discriminative metric spaces. Note that, due to the accumulation of gradients, such "misleading" behaviors of $<t_2, p_2^+, p_2^->$ can even impact the optimization in further iterations, leading to the waste of model's learning capacity.

In light of this, we propose to regularize the triplets with acute angles between $\boldsymbol{tp}^+$ and $\boldsymbol{tp}^-$ rather than naively forcing all negative proxies away from the anchor tags, which is inspired by the metric optimization in computer vision [15]. Specifically, we propose a direction-aware optimization strategy for HQRec, which takes the angle between $\boldsymbol{tp}^+$ and $\boldsymbol{tp}^-$ into consideration. To quantify the direction of optimization and regularize the above triplets with acute angle, we propose to leverage Cos metric. Based on the above intuition about the negative correlation between the angle and the degree of gradient penalty, we formulate the regularization as follows:

$$\begin{aligned}\mathcal{L}_{Reg} = -\gamma Cos(\boldsymbol{t}_i\boldsymbol{p}_i^-, \boldsymbol{t}_i\boldsymbol{p}_i^+) &= -\gamma\frac{(\boldsymbol{p}_i^- - \boldsymbol{t}_i)}{\|\boldsymbol{p}_i^- - \boldsymbol{t}_i\|}\cdot\frac{(\boldsymbol{p}_i^+ - \boldsymbol{t}_i)}{\|\boldsymbol{p}_i^+ - \boldsymbol{t}_i\|} \\ &= -\gamma\frac{\boldsymbol{t}_i\cdot\boldsymbol{t}_i - \boldsymbol{p}_i^+\cdot\boldsymbol{t}_i}{\|\boldsymbol{p}_i^- - \boldsymbol{t}_i\|\|\boldsymbol{p}_i^+ - \boldsymbol{t}_i\|},\end{aligned} \tag{7}$$

where γ is a parameter which controls the weight of regularization applied to the original $\mathcal{L}_{Rank}$ loss (cf. Eq. 2). The triplets with smaller acute angle between $\boldsymbol{tp}^+$ and $\boldsymbol{tp}^-$ will be regularized more on the gradients.

Taking the derivatives of $\mathcal{L}_{Rank} + \gamma\mathcal{L}_{Reg}$, we get the new gradients. Take t_2 for example, the gradients of t_2, p_2^+, and p_2^- are as follows:

$$\begin{aligned}\frac{\partial\mathcal{L}}{\partial\boldsymbol{t}_2} &= 2\left(\boldsymbol{p}_2^- - \boldsymbol{p}_2^+\right) - \gamma\frac{\left(\boldsymbol{t}_2 - \boldsymbol{p}_2^+\right)}{\|\boldsymbol{p}_2^- - \boldsymbol{t}_2\|\|\boldsymbol{t}_2 - \boldsymbol{p}_2^+\|} - \gamma\frac{\|\boldsymbol{t}_2 - \boldsymbol{p}_2^+\|\left(\boldsymbol{p}_2^- - \boldsymbol{t}_2\right)}{\|\boldsymbol{p}_2^- - \boldsymbol{t}_2\|^3}, \\ \frac{\partial\mathcal{L}}{\partial\boldsymbol{p}_2^+} &= 2\left(\boldsymbol{p}_2^+ - \boldsymbol{t}_2\right) - \gamma\frac{\left(\boldsymbol{p}_2^+ - \boldsymbol{t}_2\right)}{\|\boldsymbol{p}_2^- - \boldsymbol{t}_2\|\|\boldsymbol{t}_2 - \boldsymbol{p}_2^+\|}, \\ \frac{\partial\mathcal{L}}{\partial\boldsymbol{p}_2^-} &= 2\left(\boldsymbol{t}_2 - \boldsymbol{p}_2^-\right) - \gamma\frac{\left(\boldsymbol{t}_2 - \boldsymbol{p}_2^-\right)}{\|\boldsymbol{p}_2^- - \boldsymbol{t}_2\|^3\|\boldsymbol{t}_2 - \boldsymbol{p}_2^+\|}.\end{aligned} \tag{8}$$

As shown in Fig. 5(b), different from the gradient of anchor tag t_2 in Fig. 5(a) based on Eq. 2, the additional term $\boldsymbol{t}_2 - \boldsymbol{p}_2^+$ in $\frac{\partial\mathcal{L}}{\partial\boldsymbol{t}_2}$ exerts a greater force on t_2 in the direction leading away from both positive and negative proxies, thereby prioritizing increasing the learning of other informative triplets (e.g., the triplet in the right of Fig. 5). The final gradient of t_2 is in the dotted blue line, which considers the gradients in both red line and black line. Moreover, the gradient of positive proxy of category p_2^+ behaves similar to the original gradient of $\mathcal{L}_{Rank}$ in Eq. 2, unless the negative proxy of categories is very close to the anchor tag. As shown in the left of Fig. 5(b), the term $\frac{(\boldsymbol{p}_2^+ - \boldsymbol{t}_2)}{\|\boldsymbol{p}_2^- - \boldsymbol{t}_2\|\|\boldsymbol{t}_2 - \boldsymbol{p}_2^+\|}$ exerts an opposite force on p_2^+ (in dotted red line on p_2^+) in the direction to avoid being too close to the low-quality tag t_2 as compared to the previous formulation in Eq. 2, where $\mathcal{L}_{Reg}$ effectively update p_2^+ to $\tilde{p}_2^+$ in the dotted blue line. Similarly, the gradient of the negative proxy of category p_2^- would also not be shifted significantly owing to the anchor tag with low-quality.

Note that, the proposed direction-aware optimization inherently computes pair weighting based on the forces acting upon the current anchor tags and hence leads to the model mining for more informative examples to update the embedding space if the current tag is low-quality.

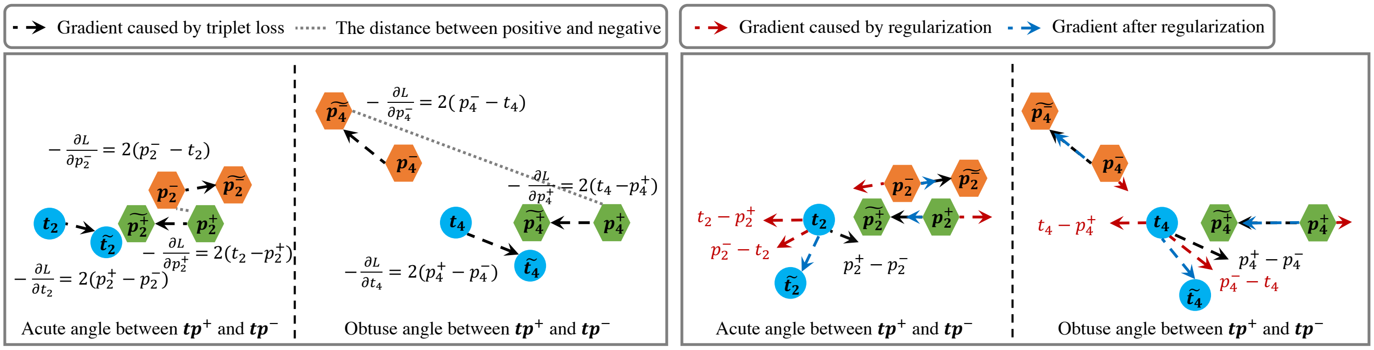
4.4 Recommendation Module

As motivated in Section 1, it is important to consider the quality of attributes in recommendation. Therefore, in this section, we introduce quality-aware graph learning module for recommendation. Specifically, the module contains two components: the quality-aware embedding fusion layer and the quality-aware embedding propagation layers. The embedding fusion layer can fuse attribute-irrelevant and attribute-relevant embeddings under the supervision of the qualities of attributes. The quality-aware embedding propagation mechanism can capture the higher-order graph structure for both user and item representation learning via propagating the quality-aware embeddings.

4.4.1 Quality-aware Embedding Fusion

Since we cannot identify whether a user interacts with an item because of its attributes, it is important to model users and items from both attribute-irrelevant and attribute-relevant perspectives. For example, in Fig. 1, Linda may be easily attracted by items' attributes, and her interacted Beef Wellington and Ribeye steak are both labeled as STEAK. In this case, the interacted attributes can reflect Linda's preference. However, the reason why Jack interacted with Seafood platter may be the recommendation from his friends. In this case, it is not suitable to model Jack as he likes SEAFOOD and "*#Shrimp*". To comprehensively model users and items, a straightforward idea is to directly combine the output of the quality assurance module with learnable user and item embeddings. However, this method separates the quality assurance and recommendation, and thus fails to leverage user-item interactions for refining attribute embeddings and enhancing recommendation.

To jointly achieve high-quality attributes selection and recommendation in a unified attribute-enhanced framework, we propose to leverage the relations between items



(a) The gradients of different angles **without** regularization $\mathcal{L}_{Reg}$

(b) The gradients of different angles **with** regularization $\mathcal{L}_{Reg}$

Fig. 5. An illustration of gradients without regularization vs. with regularization. In (a), the optimization directions of t_2 and t_4 are both towards their positive proxies, which does not consider whether the angle between $<t, p^+>$ is acute and $<t, p^->$ is obtuse. However, after adding a regularization $\mathcal{L}_{Reg}$ in (b), the gradients of t_2 and t_4 are in different directions (see dotted blue lines of t_2 and t_4), i.e., t_2 moves away from both positive and negative proxies while t_4 still moves towards its positive proxy.

and attributes as connections. The insight is that, an item can be represented by its attributes that are learned from the quality assurance module. After that, we can simultaneously update the representation of users, items, and attributes. Specifically, we first denote learnable attribute-irrelevant embeddings (i.e., $\boldsymbol{u}^{ir}$ and $\boldsymbol{v}^{ir}$) to capture the collaborative latent representations of users and items. Since the attribute information of users is hard to obtain, we propose to learn attribute-relevant embeddings for users (i.e., $\boldsymbol{u}^{re}$) together with the attribute-irrelevant embeddings.

Note that items can have multiple attributes, representing items by all of their attributes equally may harm the recommendation performance. As shown in Fig. 1(b), if we profile Jack with the combination of $\frac{1}{3}$"*#Your plate*", $\frac{1}{3}$"*#Shrimp*", and $\frac{1}{3}$"*#New York*", we may fail to stress Jack's preference to SEAFOOD but mislead his preference to STEAK since the tag "*#New York*" appears frequently together with the STEAK category.

To accurately represent items' attribute-relevant embeddings and obtain attribute-enhanced representation for both users and items, we propose a quality-aware embedding fusion mechanism, so as to properly fuse attribute embeddings under the supervision of the quality of attributes. Specifically, since categories are in high-quality and tags have quality scores, we first aggregate attribute embeddings $\boldsymbol{C}$ and $\boldsymbol{T}$ with interpretable weights as follows:

$$\boldsymbol{v}^{re} = \left[\mathbf{X}^{\mathbf{Ca}}, \boldsymbol{W}\right] \left[\begin{array}{c} \boldsymbol{C} \\ \boldsymbol{T} \end{array} \right], \tag{9}$$

where $\mathbf{X}^{\mathbf{Ca}}$ denotes the item-category relations and $\boldsymbol{W}$ is the learned quality score in Eq. 6. Then, we concatenate the attribute-irrelevant embedding and the attribute-relevant embedding to get the fused embeddings as follows:

$$\boldsymbol{u}_i^0 = [\boldsymbol{u}_i^{ir}, \boldsymbol{u}_i^{re}], \qquad \boldsymbol{v}_j^0 = [\boldsymbol{v}_j^{ir}, \boldsymbol{v}_j^{re}]. \tag{10}$$

Along this line, HQRec could represent users and items with both the collaborative signal and the attribute signal via interpretable weights.

4.4.2 *Quality-aware Embedding Propagation*

To better capture similarity across user-item bipartite graphs and encode such information into the final representation, we propose to leverage higher-order information via stack more propagation layers, which is under the graph convolutional networks (GCNs) [7], [31], [38] to produce attribute enhanced representations. The inputs of these layers are the fused user embeddings $\boldsymbol{u}_i^l$ and fused item embeddings $\boldsymbol{v}_j^l$. With the help of quality-aware embedding fusion, the high-quality item attributes will be propagated along with the item attribute-irrelevant embeddings in GCNs.

To be specific, let $\boldsymbol{u}_i^l$ and $\boldsymbol{v}_j^l$ denote the embeddings of user i and item j in the l-th layer. Their embeddings in the (l + 1)-th layer can be defined by their fusion embeddings and the aggregation of corresponding connected items (users) embeddings in l-th layer. Inspired by LightGCN [7] that achieves the state-of-the-art performance with a very light design, Our model also removes the feature transformation and nonlinear activation module. This process can be formulated as follows:

$$\boldsymbol{u}_i^{l+1} = \boldsymbol{u}_i^l + \sum_{j \in \mathcal{N}_u} \frac{1}{|\mathcal{N}_u|} \boldsymbol{v}_j^l, \quad \boldsymbol{v}_j^{l+1} = \boldsymbol{v}_j^l + \sum_{i \in \mathcal{N}_v} \frac{1}{|\mathcal{N}_v|} \boldsymbol{u}_i^l, \tag{11}$$

where $\mathcal{N}_u = \{v | \mathbf{X}_{uv} = 1\} \in \mathcal{V}$ is the item set that user u interacts with. Similarly, $\mathcal{N}_v = \{u | \mathbf{X}_{uv} = 1\} \in \mathcal{U}$ is the user set who interact with item v. Both $\frac{1}{|\mathcal{N}_u|}$ and $\frac{1}{|\mathcal{N}_v|}$ are symmetric normalization terms, which can avoid the scale of embeddings increasing with graph convolution operations [12]. Please note that, in the above equations, we do not use any convolutional operations and any non-linear activations, as its effectiveness has been well demonstrated in [7] and it can alleviate the over-smoothing problem to some extent. After L layers graph convolution, the final embeddings of a user u and an item v are the L-th layer as $\boldsymbol{u}_i^L$ and $\boldsymbol{v}_i^L$.

We would like to highlight the benefits of the quality-aware embedding fusion when performing embedding propagation. Since the fused embeddings are quality-aware, the embeddings of high-quality attributes will contribute more to the embedding learning than other nodes via high-order relations. This can largely avoid the noisy information propagated from low-quality attributes.

To learn attribute-enhanced similarity, we utilize the

TABLE 2
Statistics of the datasets used in our experiments.

Dataset	#User	#Item	#Interaction	#Category	#Tag
NYC-R	15K	4.4K	130K	174	1,905
Amazon-CD	51K	26K	875K	389	5,373
ML-20M	62K	27K	17M	20	8,534
Amazon-Book	63K	73K	5M	518	10,091

largest margin nearest neighbour algorithm for optimizing:

$$\mathcal{L}_{Rec} = \sum_{(\boldsymbol{u},\boldsymbol{v}_p)\in\mathcal{I}} \sum_{(\boldsymbol{u},\boldsymbol{v}_q)\notin\mathcal{I}} \left[m + g(\boldsymbol{u},\boldsymbol{v}_p) - g(\boldsymbol{u},\boldsymbol{v}_q)\right]_+ , \quad (12)$$

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

The final objective function of the proposed HQRec is given by considering the quality assurance module in Section 4.3 as follows:

$$\min \mathcal{L}_{Rec} + \beta\mathcal{L}_{Rank} + \gamma\mathcal{L}_{Reg}, \quad (13)$$

where β is to control the weight of category-guided metric learning, and γ is to control the weight of category-guided optimization.

5 EXPERIMENTS

In this section, we evaluate our proposed category-aware recommendation frameworks focusing on the following four research questions:

- **RQ1:** How does our HQRec framework perform compared to state-of-the-art recommendation methods?
- **RQ2:** What are the effects of different model components?
- **RQ3:** How do the hyperparameters affect the recommendation performance and how to choose optimal values?
- **RQ4:** How does our HQRec framework improve the interpretability of recommendations?

5.1 Experimental Setup

5.1.1 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 densities, i.e., the restaurants in New York City (NYC-R)[2], Amazon-CD[3], ML-20M[4], and Amazon-Book[3]. These datasets have been widely adopted in previous literature [22], [33], [37], and their statistics are summarized in Table 2.

5.1.2 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@κ and NDCG@κ instead of sampled metrics as suggested in [13]. Intuitively, the Recall metric considers whether the ground-truth is ranked amongst the top κ items while the NDCG metric is a position-aware ranking metric.

2. https://www.tripadvisor.com.sg
3. https://nijianmo.github.io/amazon/index.html/
4. https://grouplens.org/datasets/movielens

5.1.3 Methods for comparison

The following representative state-of-the-art baselines for comparison can be divided into 1) collaborative filtering (CF) based methods, 2) metric learning based methods, and 3) attribute based methods.

1) CF based methods:

- **BPR** [28]: The Bayesian personalized ranking (BPR) model is a popular method for Top-N recommendation. We adopt matrix factorization for prediction.
- **NeuMF** [8]: NeuMF combines multiple perceptrons with matrix factorization in its framework.
- **LightGCN** [7]: LightGCN devises a light graph convolution for training efficiency and generation ability.
- **EGLN** [39]: Enhanced graph learning network (EGLN) lets the enhanced graph learning module and the node embedding module iteratively learn from each other without relying on any feature input.

2) Metric learning based methods:

- **CML** [10]: Collaborative metric learning (CML) is the first model to use metric learning to solve the collaborative filtering problem of recommender systems.
- **LRML** [34]: Latent relational metric learning (LRML) employs an augmented memory module to induce a latent relation for each user-item interaction.
- **SML** [14]: Symmetric metric learning with learnable margins introduces a symmetrical positive item-centric metric to pull and push items via the dynamic margins.

3) Attribute based methods:

- **FM** [27]: Factorization Machines (FM) is a generalized MF model that captures interactions between categorical variables (e.g., item categories) by projecting them into a joint dot-product space.
- **CMLF** [10]: CMLF that based on CML integrates categories through a probabilistic interpretation of the model.
- **AMF** [9]: Aspect-based Matrix Factorization model (AMF) is a MF-based model that decomposes the rating matrix with reviews.
- **KGAT** [36]: Knowledge Graph Attention Network (KGAT) explicitly models the high-order relations in collaborative knowledge graph to provide better recommendation with item side information.
- **AGCN** [38]: Adaptive Graph Convolutional Network (AGCN) leverages an attributed user-item bipartite graph for joint item recommendation and attribute inference.
- **AGNN** [26]: Attribute Graph Neural Networks (AGNN) designs an attribute graph based on user/item attributes and utilizes a variational auto-encoder to produce the preference embedding for a strict cold user/item.

5.1.4 Implementation Details

We implement the proposed HQRec with Pytorch. Implementations of the general recommendation methods are either from open-source project or the original authors

TABLE 3
Experimental results on four benchmark datasets. The best performance is in boldface and the second runners are underlined. HQRec achieves the best performance on all datasets, where * denotes a significant improvement according to the Wilcoxon signed-rank test.

Method	Recall@5	Recall@10	NDCG@5	NDCG@10	Recall@5	Recall@10	NDCG@5	NDCG@10
	NYC-R				Amazon-CD			
BPR	0.0126	0.0230	0.0104	0.0149	0.0140	0.0235	0.0138	0.0172
NeuMF	0.0112	0.0196	0.0097	0.0131	0.0132	0.0210	0.0129	0.0167
LightGCN	0.0244	0.0479	0.0208	0.0307	0.0206	0.0447	0.0216	0.0305
EGLN	0.0250	0.0484	0.0212	0.0310	0.0211	0.0458	0.0218	0.0308
CML	0.0144	0.0281	0.0122	0.0180	0.0158	0.0355	0.0169	0.0252
LRML	0.0147	0.0289	0.0125	0.0187	0.0164	0.0361	0.0173	0.0265
SML	0.0156	0.0298	0.0134	0.0193	0.0172	0.0379	0.0181	0.0273
FM	0.0125	0.0231	0.0101	0.0139	0.0124	0.0206	0.0116	0.0146
CMLF	0.0175	0.0304	0.0155	0.0210	0.0177	0.0392	0.0191	0.0275
AMF	0.0116	0.0274	0.0080	0.0147	0.0138	0.0237	0.0133	0.0169
KGAT	0.0225	0.0458	0.0193	0.0294	0.0199	0.0421	0.0192	0.0292
AGCN	0.0251	0.0482	0.0217	0.0321	0.0214	0.0461	0.0225	0.0311
AGNN	0.0231	0.0463	0.0198	0.0307	0.0207	0.0432	0.0211	0.0304
HQRec	**0.0306***	**0.0530***	**0.0282***	**0.0375***	**0.0316**	**0.0507**	**0.0294**	**0.0371**
% Improv.	21.91	9.96	29.95	16.82	33.64	9.98	30.67	19.29
	ML-20M				Amazon-Book			
BPR	0.0690	0.1187	0.4678	0.4312	0.0119	0.0241	0.0486	0.0473
NeuMF	0.0648	0.1147	0.4168	0.3884	0.0103	0.0213	0.0462	0.0453
LightGCN	0.0817	0.1401	0.5189	0.4767	0.0239	0.0376	0.1027	0.0945
EGLN	0.0825	0.1414	0.5239	0.4813	0.0241	0.0380	0.1037	0.0954
CML	0.0643	0.1199	0.4395	0.4119	0.0154	0.0281	0.0702	0.0676
LRML	0.0627	0.1204	0.4398	0.4092	0.0166	0.0293	0.0714	0.0685
SML	0.0667	0.1298	0.4481	0.4278	0.0178	0.0304	0.0729	0.0704
FM	0.0661	0.1188	0.4304	0.3928	0.0114	0.0227	0.0474	0.0466
CMLF	0.0660	0.1248	0.4450	0.4126	0.0194	0.0348	0.0834	0.0798
AMF	0.0635	0.1146	0.4230	0.3908	0.0123	0.0272	0.0491	0.0483
KGAT	0.0683	0.1203	0.4566	0.4259	0.0228	0.0361	0.0919	0.0896
AGCN	0.0824	0.1409	0.5142	0.4737	0.0243	0.0385	0.1174	0.1005
AGNN	0.0724	0.1351	0.4951	0.4640	0.0232	0.0370	0.0975	0.0943
HQRec	**0.0874***	**0.1487***	**0.5514***	**0.5049***	**0.0276**	**0.0428**	**0.1288**	**0.1146**
% Improv.	5.96	5.13	5.25	4.91	13.58	11.17	9.71	8.06

(BPR/CML[5], NeuMF[6], LightGCN[7], EGLN[8], LRML[9], and SML[10]). Specifically, we apply TFIDF method for the review-based datasets, where we treat all reviews as a document to extract keywords. We only keep the keywords that appear at least 20 times to serve as tags. Then we can leverage these tags to find high-quality attributes for review-based datasets. Implementations of the attribute based methods are constrained to leverage both categories and tags according to the original authors algorithm (FM/CMLF[5], AMF[11], KGAT[12], AGCN[13], and AGNN[14]). We optimize the compared baselines with standard Adam optimizer. We tune all hyperparameters through grid search. In particular, learning rate in {1e-4, 5e-4, 1e-3, 5e-3, 1e-2}, the tag margin α in {0, 0.25, 0.50, 0.75, 1.0}, the weight for direction-aware optimization γ in {0, 0.001, 0.01, 0.1, 1.0}, the number of graph layer L in {1, 2, 3, 4}, the collaborative margin m in {0.00, 0.25, 0.50, 0.75, 1.00} We set the embedding dimension D to 64 for those algorithms that do not include attribute information. As for attribute based models (i.e., CMLF, AMF, KGAT, AGCN, AGNN and HQRec, we set the attribute embeddings D_a to 12 and the total embedding dimension D is still 64. The batch size is set to 10000. We 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/cheungdaven/DeepRec
6. https://github.com/hexiangnan/neural_collaborative_filtering
7. https://github.com/gusye1234/LightGCN-PyTorch
8. https://github.com/yimutianyang/SIGIR2021-EGLN
9. https://github.com/vanzytay/WWW2018_LRML
10. https://github.com/MingmingLie/SML
11. https://github.com/cthurau/pymf
12. https://github.com/LunaBlack/KGAT-pytorch
13. https://github.com/yimutianyang/AGCN
14. https://github.com/lylbaidu/AGNN

5.2 Overall Performance Comparison (RQ1)

In general, HQRec outperforms all 13 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 our proposed quality-aware recommendation framework is capable of effective collaborative ranking. Compared with the second best models (i.e., EGLN and AGCN), the performance gains of HQRec on the two datasets range from reasonably large (4.91% achieved with NDCG@10 on the ML-20M dataset) to significantly large (33.64% achieved with Recall@5 on the Amazon-CD dataset).

In particular, the six models that consider attributes (i.e., FM, CMLF, AMF, KGAT, AGCN, and AGNN) cannot outperform some competitors without attributes (e.g., LightGCN and EGLN). When the auxiliary data (e.g., reviews and user profile) are not all attainable, those attribute based methods cannot maintain competitive results. However, our HQRec with quality measurement module for attributes can achieve the best performance. This is particularly evident on sparser datasets, where HQRec significantly outperforms

TABLE 4
Ablation analysis of our proposed HQRec on the NYC-R dataset.

Method	Recall@5	Recall@10	Recall@20	NDCG@5	NDCG@10	NDCG@20
HQRec-attribute	0.0175	0.0304	0.0518	0.0155	0.0210	0.0282
HQRec-graph	0.0212	0.0464	0.0851	0.0161	0.0272	0.0411
HQRec-naïve	0.0216	0.0475	0.0852	0.0162	0.0271	0.0402
HQRec-CML	0.0232	0.0488	0.0873	0.0181	0.0290	0.0422
HQRec-remove	0.0248	0.0404	0.0706	0.0242	0.0209	0.0402
HQRec	0.0306	0.0530	0.0906	0.0278	0.0375	0.0494

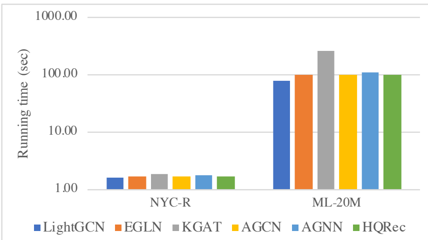


Fig. 6. Runtime analysis (per epoch) compared with graph based methods on different datasets.

TABLE 5
Performance comparisons of different propagation depth L.

Layer	Recall@5	NDCG@10	NDCG@5	NDCG@10
L = 1	0.0127	0.0243	0.0105	0.0154
L = 2	0.0119	0.0285	0.0099	0.0169
L = 3	0.0306	0.0530	0.0278	0.0375
L = 4	0.0190	0.0270	0.0168	0.0243

AGCN by 29.95% with NDCG@5 on NYC-R and 33.64% with Recall@5 on Amazon-CD.

Note that, the most time-consuming part of HQRec is with the graph convolutional layers, which have also been used in the second runners (e.g., EGLN and ACGN) to capture higher-order graph structure. Relative to that, the overhead from our quality measurement is quite minor. Specifically, the time complexity of measuring qualities of attributes is $\mathcal{O}(N_c + N_t)$, where N_c and N_t are the number of categories and tags. Both N_c and N_t are far less than the number of users and items. As shown in Fig. 6, we also found the runtimes of HQRec are on the same scale as most graph based baselines. These five baselines are selected because of their good performance in terms of Recall and NDCG (shown in Table 3).

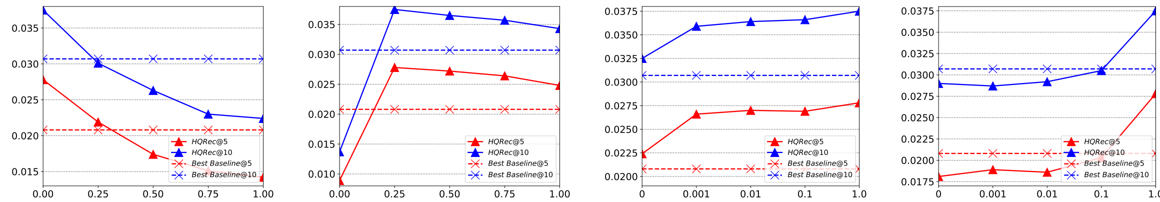
5.3 Model Ablation (RQ2)

To better understand our proposed techniques, i.e., category-guide metric learning (CML), direction-aware optimization (DAO), and quality-aware graph learning, we study HQRec as follows:

- **HQRec-attributes** is the basic metric learning model based on attributes.
- **HQRec-graph** is the model with graph-enhanced representation for both users and items.
- **HQRec-naïve** is the model based on HQRec-graph, where the similarities between categories and tags are calculated from the co-occurrence matrix of categories and tags. The more frequent appearance of a category and a tag, the high similarity score of these two attributes, and thus the high quality of the tag.
- **HQRec-CML** is the model with category-guide quality metric learning (CML), which integrates with both quality-aware embedding fusion and quality-aware embedding propagation in graph.
- **HQRec-remove** is the model with direction-aware optimization (DAO) and will remove the informative attributes via a global threshold σ. The attribute with quality scores lower than σ will be removed directly. In the experiment, we search the hyperparameter σ in $\{0.8, 0.85, 0.9, 0.95\}$ and find that HQRec-remove achieves the best performance when $\sigma = 0.8$.
- **HQRec (full)** is the model based on HQRec-CML with direction-aware optimization (DAO), where the low-quality attributes are leveraged via different weights.

From Table 4, we have the following observations:

- In general, the performance of HQRec-graph is better than the basic HQRec-attribute in all cases, where the performance gains of HQRec-graph over HQRec-attribute achieve an average improvement of 46.02% on Recall and 26.38% on NDCG, respectively. These results also corroborate our conjecture that capturing the higher-order graph structure for both user and item representation learning is more helpful to improve performance than directly combing all the attributes, which is consistent with the observation in Table 3.
- We observe that HQRec-naïve can slightly improve the performance of HQRec-graph on Recall but will harm the performance on NDCG. Since low-quality tags can appear frequently with some categories, the similarities between such a tag and the co-occurred categories are high. In this case, HQRec-naïve that fully leverages the low-quality attributes cannot avoid their negative impacts, and thus decrease the performance of recommendation.
- One step further, HQRec-CML that measures the quality of attributes performs better than HQRec-naïve by up to 7.41% on Recall and 11.73% on NDCG, respectively. Compared with HQRec-naïve, HQRec-CML not only leverages the information of category-tag co-occurrence matrix, but also considers the distribution of tags over the categories, which can help distinguish low-quality tags from high frequent tags. In this case, HQRec-CML can leverage high-quality attributes from both categories and tags and model users and items via clean semantics of categories and rich semantics of tags.
- HQRec-remove can outperform HQRec-CML with 6.90% on Recall@5 and 6.08% on NDCG@5 but decrease the



(a) α (b) m (c) β (d) γ

Fig. 7. Performance regarding NDCG@5 and NDCG@10 of the best-performing baseline and HQRec with varying hyperparameters on NYC-R.

performance with an average of -12.92% on the remaining cases. Note that, when removing the low-quality tags, it also removes the relations of the items that they are connected to. Such information can be useful especially when the interactions between users and items are sparse. Moreover, it has also been proved that retaining all information with different weights in a probabilistic way is better than removing all of them in a deterministic way [2], [41].

- The performance gains of HQRec over HQRec-CML fluctuate, ranging from 3.78% (achieved on Recall@20) to 25.79% (achieved with NDCG@5), showing that it is good to take advantage of the existence of the positive category proxy for a more optimal direction. Moreover, the performance gains of HQRec over HQRec-remove achieve an average of 27.63% on Recall and 37.49% on NDCG. By keeping the low-quality tags and the infrequent ones with low entropy into the model training, HQRec can leverage the proposed direction-aware optimization to retain the informative relations between items and alleviate the negative impacts brought by low-quality tags. Such results are consistent with those in Table 3, showing the effectiveness of applying quality measurement for reasonably weighting attributes for recommendation.

5.4 Hyperparameter Study (RQ3)

Our proposed HQRec framework introduces five hyperparameters, which are α, m, β, γ, and L.

From Fig. 7, we have the following observations: (1) α is the basic margin to enforce the difference between positive and negative triplets, where we found that the optimal α is about 0. Though the result seems counter-intuitive, this special margin is designed for the low-quality attributes, which decreases their contributions to the gradients. (2) m is the margin to enforce the difference between positive and negative triplets for users and items. The optimal m value is about 0.25 and the optimal m can be obtained by slight tuning. (3) β controls the weight of the category-guide metric learning, which aims to organize the embeddings of both categories and tags to select high-quality attributes. The optimal β is about 1.0 (4) γ controls the weight of the direction-aware optimization, where we found that the optimal γ is about 0.5. The rules for selecting β and γ could be the rule-of-thumb in practice across the used datasets.

Furthermore, Table 5 shows the performance of HQRec with the varying layer depths of GCN L. HQRec achieves the best performance with $L = 3$. Since NYC-R has sparse interactions with 0.20% 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.

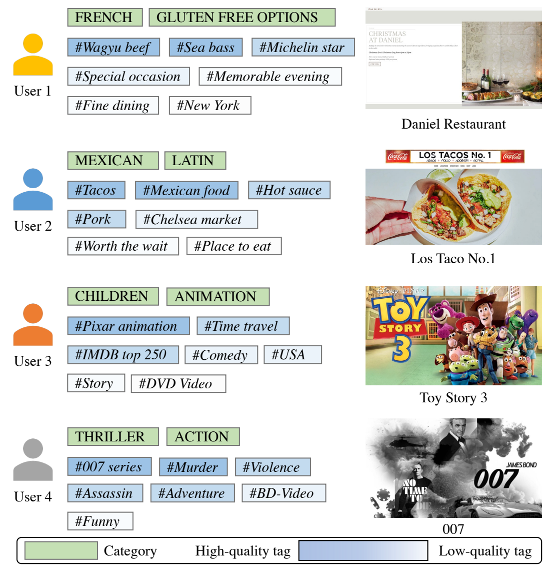


Fig. 8. The quality-aware user profiles are obtained by our proposed HQRec, which is consistent with the interacted items of users.

5.5 Interpretable Case Studies (RQ4)

To provide more insights into the advantages of HQRec in providing high-quality attributes for recommendations, we demonstrate four random users with their closest attributes retained by HQRec, and their interacted items from original data, on NYC-R and ML-20M. The darker blue represents the higher quality of the tag. Since the relations among users and attributes can be measured through user-categories and user-tag distance in the metric space, we obtain each user's top 2 categories and top 7 tags by ranking the distance between each user to all attributes, where the representations of users, items, categories, and tags are learned by HQRec.

From Fig. 8, besides the high-quality categories, we observe that the tags retained for each user are also in high-quality and highly coherent to the selected categories, such as "*#Wagyu beef*" for User 1 on NYC-R and "*#Pixar animation*" for User 3 on ML-20M, which can be regarded as the fine-grained attributes of FRENCH and ANIMATION, respectively. As a consequence, the quality-aware user profiles are highly rational, which are consistent with the interacted items (e.g., Danial Restaurant for User 1 on NYC-R, as well as Toy Story 3 for User 3 on ML-20M).

Note that the exact quality scores of tags we represent here are not perfectly accurate due to the implicit nature of attributes' qualities. However, they nonetheless provide valuable insights into the users' profiling in an unsupervised fashion. The interpretable users' categories and tags can provide the potential for further user profiling and personalization, which has not been fully leveraged by existing systems. In the meantime, such organic integration of users' preferences towards attributes also hints at the application of our framework in more novel recommendation scenarios such as adaptive recommendation, where a user can freely specify certain attributes, and let our framework retrieve items that are both relevant to the specified attributes and the user's own preferences on other attributes.

6 Conclusion

In this paper, we have proposed quality-aware recommendation framework based on item attributes and user-item interactions, which can effectively enhance recommendation in both accuracy and interpretability. Specifically, we have proposed to leverage item categories to automatically select high-quality tags without unsupervison. Extensive experiments have demonstrated clear improvements of HQRec over the state-of-the-art baselines and the insightful case studies have showed the accuracy and interpretability of our automatic quality assurance module.

Acknowledgement

This work was supported in part by the National Natural Science Foundation of China (No.62172362) and Leading Expert of "Ten Thousands Talent Program" of Zhejiang Province (No.2021R52001).

References

[1] S. Cui, S. Wang, J. Zhuo, L. Li, Q. Huang, and Q. Tian. Towards discriminability and diversity: Batch nuclear-norm maximization under label insufficient situations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3941–3950, 2020.

[2] S. De, Y. Hu, V. V. Meduri, Y. Chen, and S. Kambhampati. Bayeswipe: A scalable probabilistic framework for improving data quality. *ACM J. Data Inf. Qual.*, 8(1):5:1–5:30, 2016.

[3] I. Fernández-Tobías, M. Braunhofer, M. Elahi, F. Ricci, and I. Cantador. Alleviating the new user problem in collaborative filtering by exploiting personality information. *User Modeling and User-Adapted Interaction*, 26(2):221–255, 2016.

[4] Y. Gao, J. Sang, C. Fu, Z. Wang, T. Ren, and C. Xu. Metadata connector: Exploiting hashtag and tag for cross-osn event search. *IEEE Transactions on Multimedia*, 23:510–523, 2020.

[5] W. L. Hamilton, R. Ying, and J. Leskovec. Representation learning on graphs: Methods and applications. *arXiv:1709.05584*, 2017.

[6] J. He, X. Li, and L. Liao. Category-aware next point-of-interest recommendation via listwise bayesian personalized ranking. In *IJCAI*, pages 1837–1843, 2017.

[7] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 639–648, 2020.

[8] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*, pages 173–182, 2017.

[9] Y. Hou, N. Yang, Y. Wu, and S. Y. Philip. Explainable recommendation with fusion of aspect information. *World Wide Web*, 22(1):221–240, 2019.

[10] C.-K. Hsieh, L. Yang, Y. Cui, T.-Y. Lin, S. Belongie, and D. Estrin. Collaborative metric learning. In *Proceedings of the 26th international conference on world wide web*, pages 193–201, 2017.

[11] C. Huang, H. Xu, Y. Xu, P. Dai, L. Xiao, M. Lu, L. Bo, H. Xing, X. Lai, and Y. Ye. Knowledge-aware coupled graph neural network for social recommendation. In *35th AAAI Conference on Artificial Intelligence*, 2021.

[12] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. *arXiv:1609.02907*, 2016.

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

[14] M. Li, S. Zhang, F. Zhu, W. Qian, L. Zang, J. Han, and S. Hu. Symmetric metric learning with adaptive margin for recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 4634–4641, 2020.

[15] Y. Li, J. Hu, C. Zhai, and Y. Chen. Improving one-class collaborative filtering by incorporating rich user information. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 959–968, 2010.

[16] J. Lin. Divergence measures based on the shannon entropy. *IEEE Transactions on Information Theory*, 1991.

[17] J. Lin, K. Sugiyama, M.-Y. Kan, and T.-S. Chua. Addressing cold-start in app recommendation: latent user models constructed from twitter followers. In *Proceedings of the 36th international ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 283–292, 2013.

[18] F. Liu, Z. Cheng, L. Zhu, Z. Gao, and L. Nie. Interest-aware message-passing gcn for recommendation. In *Proceedings of the Web Conference 2021*, pages 1296–1305, 2021.

[19] Q. Liu, R. Xie, L. Chen, S. Liu, K. Tu, P. Cui, B. Zhang, and L. Lin. Graph neural network for tag ranking in tag-enhanced video recommendation. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 2613–2620, 2020.

[20] W. Liu, X. Zheng, J. Su, M. Hu, Y. Tan, and C. Chen. Exploiting variational domain-invariant user embedding for partially overlapped cross domain recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 312–321, 2022.

[21] Y. Liu, X. Xia, L. Chen, X. He, C. Yang, and Z. Zheng. Certifiable robustness to discrete adversarial perturbations for factorization machines. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 419–428, 2020.

[22] C. Ma, P. Kang, B. Wu, Q. Wang, and X. Liu. Gated attentive-autoencoder for content-aware recommendation. In *Proceedings of the twelfth ACM international conference on web search and data mining*, pages 519–527, 2019.

[23] W. Meng, D. Yang, and Y. Xiao. Incorporating user micro-behaviors and item knowledge into multi-task learning for session-based recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1091–1100, 2020.

[24] C. Park, D. Kim, X. Xie, and H. Yu. Collaborative translational metric learning. In *2018 IEEE International Conference on Data Mining*, pages 367–376, 2018.

[25] S.-T. Park and W. Chu. Pairwise preference regression for cold-start recommendation. In *Proceedings of the third ACM conference on Recommender systems*, pages 21–28, 2009.

[26] T. Qian, Y. Liang, Q. Li, and H. Xiong. Attribute graph neural networks for strict cold start recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 2020.

[27] S. Rendle. Factorization machines. In *2010 IEEE International Conference on Data Mining*, pages 995–1000, 2010.

[28] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. *arXiv:1205.2618*, 2012.

[29] S. Sedhain, A. Menon, S. Sanner, L. Xie, and D. Braziunas. Low-rank linear cold-start recommendation from social data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2017.

[30] K. Song, F. Nie, J. Han, and X. Li. Parameter free large margin nearest neighbor for distance metric learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2017.

[31] J. Sun, Z. Cheng, S. Zuberi, F. Pérez, and M. Volkovs. Hgcf: Hyperbolic graph convolution networks for collaborative filtering. In *Proceedings of the Web Conference 2021*, pages 593–601, 2021.

[32] Y. Tan, C. Yang, X. Wei, C. Chen, L. Li, and X. Zheng. Enhancing recommendation with automated tag taxonomy construction in

hyperbolic space. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pages 1180–1192. IEEE, 2022.
[33] Y. Tan, C. Yang, X. Wei, Y. Ma, and X. Zheng. Multi-facet recommender networks with spherical optimization. In *2021 IEEE 37th International Conference on Data Engineering*, pages 1524–1535, 2021.
[34] Y. Tay, L. Anh Tuan, and S. C. Hui. Latent relational metric learning via memory-based attention for collaborative ranking. In *Proceedings of the 2018 World Wide Web Conference*, pages 729–739, 2018.
[35] 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 *Proceedings of the 13th International Conference on Web Search and Data Mining*, pages 609–617, 2020.
[36] X. Wang, X. He, Y. Cao, M. Liu, and T.-S. Chua. Kgat: Knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 950–958, 2019.
[37] X. Wang, X. He, F. Feng, L. Nie, and T.-S. Chua. Tem: Tree-enhanced embedding model for explainable recommendation. In *Proceedings of the 2018 World Wide Web Conference*, pages 1543–1552, 2018.
[38] 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 *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 679–688, 2020.
[39] Y. Yang, L. Wu, R. Hong, K. Zhang, and M. Wang. Enhanced graph learning for collaborative filtering via mutual information maximization. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 71–80, 2021.
[40] F. Yu, L. Cui, W. Guo, X. Lu, Q. Li, and H. Lu. A category-aware deep model for successive poi recommendation on sparse check-in data. In *Proceedings of the Web Conference 2020*, pages 1264–1274, 2020.
[41] K. Zhang, H. Gao, X. Han, Z. Cai, and J. Li. Modeling and computing probabilistic skyline on incomplete data. *IEEE Transactions on Knowledge and Data Engineering*, 32(7):1405–1418, 2019.
[42] F. Zhao and Y. Guo. Learning discriminative recommendation systems with side information. In *IJCAI*, pages 3469–3475, 2017.
[43] X. Zheng, M. Wang, C. Chen, Y. Wang, and Z. Cheng. Explore: Explainable item-tag co-recommendation. *Information Sciences*, 474:170–186, 2019.
[44] J. Zhou, S. Gou, R. Hu, D. Zhang, J. Xu, A. Jiang, Y. Li, and H. Xiong. A collaborative learning framework to tag refinement for points of interest. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1752–1761, 2019.
[45] T. Zhu, L. Sun, and G. Chen. Embedding disentanglement in graph convolutional networks for recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 2021.

Xiaolin Zheng is PhD, Professor, PhD supervisor, and the deputy director of the Institute of Artificial Intelligence, Zhejiang University. Senior member of IEEE, and Distinguished Member of China Computer Federation, a Committee Member in Service Computing of China Computer Federation. His main research interests include Recommender Systems, Privacy-Preserving Computing, and Intelligent Finance. He has published more than 100 referenced papers in TKDE, NeurIPS, WWW, KDD, and so on.

Yanchao Tan obtained her PhD degree in the College of Computer Science, Zhejiang University, China, in 2022. She is currently a lecturer in the College of Computer and Data Science, Fujian Key Laboratory of Network Computing and Intelligent Information Processing, Fuzhou University. Her research interests include data mining, recommender system, and healthcare.

Yan Wang is currently a Full Professor in the School of Computing, Macquarie University, Australia. He received his PhD from Harbin Institute of Technology (HIT), P. R. China. Prior to joining Macquarie University in 2003, he was a Postdoctoral Fellow/Research Fellow in the Department of Computer Science, School of Computing, National University of Singapore (NUS). He has published a number of research papers in international conferences including AAAI, AAMAS, ICDE, IJCAI, WWW, and journals including CSUR, TIST, TKDE, TSC and TWEB. His research interests cover trust management, recommender systems, social computing and service computing.

Xiangyu Wei received the BS degree in software engineering from Zhejiang University, China. He obtained a master's degree in computer science from Zhejiang University in 2022. His research interests include recommendation systems and data mining.

Shengjia Zhang received the BS degree in computer science from Jilin University, China, in 2020. He is currently working toward the Master degree in software enginering, Zhejiang University, China. His research interests include recommender system and data mining.

Chaochao Chen obtained his PhD degree in computer science from Zhejiang University, China, in 2016, and he was a visiting scholar in UIUC, during 2014-2015. He is currently a Distinguished Research Fellow at Zhejiang University. Before that, he was a Staff Algorithm Engineer at Ant Group. His research mainly focuses on recommender system, privacy preserving machine learning, and graph machine learning. He has published more than 50 papers in peer reviewed journals and conferences.

Longfei Li is currently a Staff Algorithm Engineer at Ant Financial. His research mainly focuses on machine learning, AutoML and Causal Infernece. He has published more than 10 papers in top-tier machine learning and data mining conferences.

Carl Yang is an Assistant Professor in Emory University. He received his Ph.D. in Computer Science at University of Illinois, Urbana-Champaign in 2020, and B.Eng. in Computer Science and Engineering at Zhejiang University in 2014. His research interests span graph data mining, applied machine learning, knowledge graphs and federated learning, with applications in recommender systems, social networks, neuroscience and healthcare. Carl's research results have been published in 70+ peer-reviewed papers in top venues including TKDE, KDD, WWW, NeurIPS, ICML, ICLR, ICDE, SIGIR and ICDM. He is also a recipient of the Dissertation Completion Fellowship of UIUC in 2020, the Best Paper Award of ICDM in 2020, the Dissertation Award Finalist of KDD in 2021, the Amazon Research Award in 2022, the Best Paper Award of KDD Health Day in 2022 and multiple Emory internal research awards.