Graph Neural Networks Boosted Personalized Tag Recommendation Algorithm

Xuewen Chen*, Yonghong Yu*, Fengyixin Jiang* Li Zhang†, Rong Gao‡, Haiyan Gao*

*Tongda College
Nanjing University of Posts and Telecommunications, China
Email: yuyh@njupt.edu.cn
†Department of Computer and Information Sciences
Northumbria University, Newcastle, UK
Email: li.zhang@northumbria.ac.uk
‡School of Computer Science
Hubei University of Technology, China
Email: gaorong@hbut.edu.cn

Abstract—Personalized tag recommender systems recommend a set of tags for items based on users' historical behaviors, and play an important role in the collaborative tagging systems. However, traditional personalized tag recommendation methods cannot guarantee that the collaborative signal hidden in the interactions among entities is effectively encoded in the process of learning the representations of entities, resulting in insufficient expressive capacity for characterizing the preferences or attributes of entities. In this paper, we proposed a graph neural networks boosted personalized tag recommendation model, which integrates the graph neural networks into the pairwise interaction tensor factorization model. Specifically, we consider two types of interaction graph (i.e. the user-tag interaction graph and the item-tag interaction graph) that is derived from the tag assignments. For each interaction graph, we exploit the graph neural networks to capture the collaborative signal that is encoded in the interaction graph and integrate the collaborative signal into the learning of representations of entities by transmitting and assembling the representations of entity neighbors along the interaction graphs. In this way, we explicitly capture the collaborative signal, resulting in rich and meaningful representations of entities. Experimental results on real world datasets show that our proposed graph neural networks boosted personalized tag recommendation model outperforms the traditional tag recommendation models.

I. INTRODUCTION

With the rapidly increasing of available information on Internet, the problem of information overload has become a big issue that hinders users to quickly find related information from massive data. Recommendation systems [1] have become essential intelligent components in application platforms such as e-commerce, movie websites and online news. Recommendation system mainly mines users' implicit preferences based on historical user data (including browsing, clicking, or buying), providing users with personalized recommendation services. Thereby, recommender systems can effectively alleviate the problem of information overload, and have become a research hotspot in both the academia and industry.

As a branch of the recommendation systems, tag recommendation systems automatically recommend a list of tags for users to annotate an item. Collaborative tagging systems [2], [3] allow users to upload items (e.g. photos, songs, movies and websites) and annotate them with keywords, so-called tags. In the collaborative tagging systems, besides being used to describe the multiple facets of items, tags are beneficial to these systems for efficiently managing and searching related items. Tag recommendation can be roughly divided into non-personalized and personalized tag recommendation according to whether the users' personalized preferences are considered when making tag recommendation. Differ from non-personalized tag recommendation systems [4]–[7] that provide all users with the same tags for a certain item, personalized tag recommendation systems [2], [3], [8], [9] provide personalized tag recommendation for each user by taking users' tagging preferences into account, which makes personalized tag recommendation more challenging than non-personalized tag recommendation. Due to users' unique personality and habits, different users usually assign different tags for a given item. Hence, personalized tag recommendation is more meaningful and practical for real-world tag recommendation scenarios. The most popular personalized tag recommendation models are PITF [9] and NLTF [10]. PITF uses pairwise interactions between users, items, and tags modeling user preferences, and adopting BPR [11] optimization criteria improves the performance of tag recommendation. Different from the linear model PITF, NLTF is a personalized tag recommendation algorithm based on Gaussian kernel for non-linear tensor factor factorization, and uses the Gaussian distribution to extend tag recommendations to a non-linear space.

Recently, deep learning techniques have shown great potential in various fields, such as natural language processing and computer vision. Among them, the graph neural networks (GNNs) [12] is an effective graph representation learning framework, which learns the representations of node or subgraph that preserve the structure of target graphs. In the field of recommendation systems, some researcher incorporate the GNN technique into traditional recommendation models to improve the recommendation performance [13]–[16]. For example, in [13], Qian et al. proposed a news recommendation

model, called IGNN, which integrates a user-item interactions graph and a knowledge graph into the news recommendation model. Fan et al. [14] presented a graph neural network framework, named GraphRec, for social recommendation. In [15], Wu et al. proposed a novel method for session-based recommendation with graph neural networks, called SR-GNN. Wang et al. [16] proposed a recommendation model based on graph neural networks, which exploits the user-item graph structure by propagating embeddings on it. As shown in the above works, graph neural networks could provide great potential to advance the item recommendation models. However, few works have employed the GNNs techniques to boost the personalized tag recommendation. In addition, traditional personalized tag recommendation methods can not guarantee that the collaborative signal hidden in the interaction information, which can be viewed as the behaviorial similarity between interacted entities, is explicitly encoded in the process of learning the representations of entities (i.e., users, items and tags), resulting in insufficient expressive capacity for characterizing the preferences or attributes of entities. Intuitively, it is beneficial for personalized tag recommendation models to integrate the collaborative signal into the process of learning the representations of entities in an explicit manner.

In this paper, inspired by [16], we proposed a graph neural networks boosted personalized tag recommendation model (GNN-PTR), which integrates the graph neural networks into the classic pairwise interaction tensor factorization model. Specifically, we consider two bipartite interactions derived from the user-item-tag assignment information, i.e. the user-tag interactions and item-tag interactions. Then, for each type of interactions, we exploit the graph neural networks to enrich the representations of entities by aggregating the messages of their neighbors, which is propagated over the corresponding interaction graph. In this way, we explicitly injects the collaborative signal that is encoded in the structure of interaction graphs into the process of learning representations of entities. Finally, we adopt the Bayesian personalized ranking (BPR) optimization criterion [11] to optimize the model parameters of GNN-PTR.

The key contributions of our work are summarized as follows:

- We proposed a graph neural networks boosted personalized tag recommendation model, which boosts the classic pairwise interaction tensor factorization model by utilizing the graph neural networks.
- For the task of personalized tag recommendation, we propose to take two types of interactions into account, i.e. the user-tag interactions and the item-tag interactions, and integrate the collaborative signal that is encoded in the entity interaction graphs into the process of learning representations of entities by leveraging the embedding propagation layers.
- We conduct comprehensive experiments on real world datasets to evaluate the effectiveness of our proposed graph-neural-networks-based personalized tag recommendation model. Experimental results show that our proposed method outperforms the state-of-the-art personalized tag recommendation methods.

II. RELATED WORK

In this section, we review the major related work, including the personalized tag recommendation methods and the graph neural-network-based item recommendation algorithms.

A. Personalized Tag Recommendation Methods

Personalized tag recommendation is still an emerging research area and the literature concerning personalized tag recommendation is sparse. Typical representative personalized tag recommendation methods include HOSVD [8], RTF [17], PITF [9] etc.

The tagging information naturally can be represented by a 3-order tensor since the tagging information encodes the ternary relationships between users, items and tags. Hence, most of existing personalized tag recommendation methods are built on tensor factorization techniques, especially the Tucker Decomposition (TD) model. For instance, in [8], Symeonidis et al. developed a unified framework to model three types of entities (i.e. users, items and tags), and applied the Higher Order Singular Value Decomposition (HOSVD) technique [18] to reveal the latent semantic associations between users, items and tags. Cai et al. [19] proposed the lower-order tensor decomposition (LOTD) for tag recommendation. The LOTD utilizes low-order polynomials to enhance statistics among users, items and tags. Both the HOSVD [8] and LOTD [19] basically adopts the point-wise regression method to learn the factorization model from observed tagging data. By contrast, Rendle et al. [17] proposed the Ranking with Tensor Factorization (RTF), which learns personalized ranking of user preferences for tags by optimizing the ranking statistic AUC (area under the ROC-curve) rather than optimizing the square-loss. The computation cost of Tucker Decomposition model used in both HOSVD and RTF makes them infeasible for large-scale personalized tag recommender systems since the model equation of Tucker Decomposition results in a cubic runtime in the factorization dimension. In [9], Rendle et al. proposed the Pairwise Interaction Tensor Factorization (PITF) model, which explicitly models the pairwise interactions between users, items and tags. To increase the capacity of personalized recommendation model, Fang et al. [10] proposed a non-linear tensor factorization method, named NLTF. NLTF also enhances PITF by exploiting the Gaussian radial basis function to capture the complex relations between users, items and tags. In [20], Yuan et al. proposed an attention-based method, called ABNT, which utilizes the multi-layer perceptron to model the non-linearities of interactions between users, items and tags.

B. The Graph Neural-Network-based Item Recommendation Methods

Typical graph-neural-networks-based item recommendation algorithms include GraphRec [14], IGNN [13], SR-GNN [15], NGCF [16] and so on.

In [13], Qian et al. proposed a news recommendation model, called IGNN, which integrates a user-item interaction graph and a knowledge graph into the news recommendation model. Specially, IGNN utilizes the knowledge-aware convolutional neural networks to extract the knowledge-level information from the knowledge graph. Meanwhile, it leverage a graph neural network to fuse the high-order collaborative signals in the process of learning users and news representations. Fan et al. [14] presented a graph neural network framework, named GraphRec, for social recommendation. The GraphRec coherently models the user-user social graph, the user-item interact graph as well as the heterogeneous strengths. In [15], Wu et al. proposed a novel method for session-based recommendation with graph neural networks, called SR-GNN. The SR-GNN models separated session sequences into graph structure data and utilizes graph neural networks to capture complex item transitions. Wang et al. [16] proposed a recommendation model based on graph neural network, which exploits the user-item graph structure by propagating embeddings on it. Ying et al. [21] developed a graph convolutional network algorithm, called PinSage, which combines random walks and graph convolutions to generate embeddings of nodes that incorporate both graph structure as well as node feature information. In [22], Berg et al. proposed a graph auto-encoder framework for matrix completion. The graph auto-encoder products latent features of user and item node through a form of message passing on the bipartite user-item interaction graph. Wu et al. [23] proposed a graph convolutional neural network based social recommendation model, which utilizes the graph convolutional networks to capture how users' preferences are influenced by the social diffusion process in social networks. Despite the considerable progress made by the GNN in the field of item recommendation, few studies have been conducted to exploit the GNNs to advance the personalized tag recommendation. Different from the above existing studies, in this paper, we leverage the GNNs technique to deal with the problem of personalized tag recommendation.

III. PRELIMINARIES

A. Problem Description

Differ from traditional item recommendation systems with two types of entities, i.e. users and items, personalized tag recommender systems usually consists of three types of entities: the set of users U, the set of items I and the set of tags T. The interaction information between user, item and tag is represented as $S \subseteq U \times I \times T$. A ternary $(u, i, t) \in S$ indicates that the user u has annotate the item i with the tag t. In addition, we call a user-item pair (u, i) as a post following the common used scheme in [9], [17]. The set of observed user-item pairs P_S in S is defined as:

$$P_S = \{(u, i) \,|\, \exists t \in T : (u, i, t) \in S\} \quad (1)$$

From the ternary relation set S, personalized tag recommendation methods, especially tensor factorization-based methods,

usually deduce a three-order tensor $Y \in \mathbb{R}^{|U| \times |I| \times |T|}$, whose element $y_{u,i,t}$ is defines as:

$$y_{u,i,t} = \begin{cases} 1, & (u,i,t) \in S \\ 0, & otherwise, \end{cases} \quad (2)$$

The interpretation scheme for Y is similar to the scheme that is used in one-class collaborative filtering [24], [25], i.e., $y_{u,i,t} = 1$ indicates a positive instance, and the remaining data is the mixture of negative instances and missing values.

Personalized tag recommender systems aim at recommending a ranked list of tags to users for annotating an item. Formally, the ranked list of Top-N tags given the user-item pair (u, i) is defined as,

$$Top(u, i, N) = \underset{t \in T}{arg\,max}^{N} \, \widehat{y}_{u,i,t} \quad (3)$$

where N denotes the number of recommended tags. And $\widehat{y}_{u,i,t}$ indicates the probability of the user u annotates the item i with the tag t.

B. Pairwise Interaction Tensor Factorization

Based on the three-order tensor Y, PITF learns latent feature matrices: $\mathbf{U} \in \mathbb{R}^{|U| \times d}, \mathbf{I} \in \mathbb{R}^{|I| \times d}, \mathbf{T}^U \in \mathbb{R}^{|T| \times d}, \mathbf{T}^I \in \mathbb{R}^{|T| \times d}$ (d is the factorization dimension), which corresponds the latent user feature matrix, the latent item feature matrix, the latent user-specific tag feature matrix and the latent item-specific tag feature matrix, respectively. PITF explicitly models the pairwise interactions between users, items and tags by using the following score function $\widehat{y}_{u,i,t}$, formally,

$$\widehat{y}_{u,i,t}^{PITF} = \sum_{f=1}^{d} \mathbf{U}_{u,f} \mathbf{T}_{t,f}^U + \sum_{f=1}^{d} \mathbf{I}_{i,f} \mathbf{T}_{t,f}^I \quad (4)$$

The first part of equation (4) models the interaction between users and tags, and the second part models the item-tag interaction. In addition, PITF assumes that users prefer the observed tag t over the unobserved tag t'. In other words, given a user-item pair (u, i), $\widehat{y}_{u,i,t} > \widehat{y}_{u,i,t'}$ if the user u has annotated the item i with the tag t instead of using the tag t'. In this paper, we use t to represent the positive tag, i.e. the observed tag, and t' to denote the negative tag, i.e. the unobserved tag. Hence, the training set D_S (i.e., the set of quadruple (u, i, t, t')) with the pairwise constraint is defined as:

$$D_S = \{(u, i, t, t') | (u, i, t) \in S \wedge (u, i, t') \notin S\} \quad (5)$$

Then, PITF adopts the Bayesian Personalized Ranking (BPR) optimization criterion [11] to estimate model parameters $\mathbf{\Theta} = \{\mathbf{U}, \mathbf{I}, \mathbf{T}^U, \mathbf{T}^I\}$, and the objective function of PITF is:

$$\mathcal{L}^{PITF} = \min_{\mathbf{U}, \mathbf{I}, \mathbf{T}^U, \mathbf{T}^I} \sum_{(u,i,t,t') \in D_S} -\ln \sigma(\widehat{y}_{u,i,t,t'}) + \lambda_{\Theta} ||\mathbf{\Theta}||_F^2$$

$$(6)$$

where $\widehat{y}_{u,i,t,t'} = \widehat{y}_{u,i,t}^{PITF} - \widehat{y}_{u,i,t'}^{PITF}$ is a real value function that captures the relationship between the user u, the item i, the tags t and t'. $\sigma(x)$ is the sigmoid function $\frac{1}{1+e^{-x}}$. And λ_{Θ} denotes the regularization parameter.
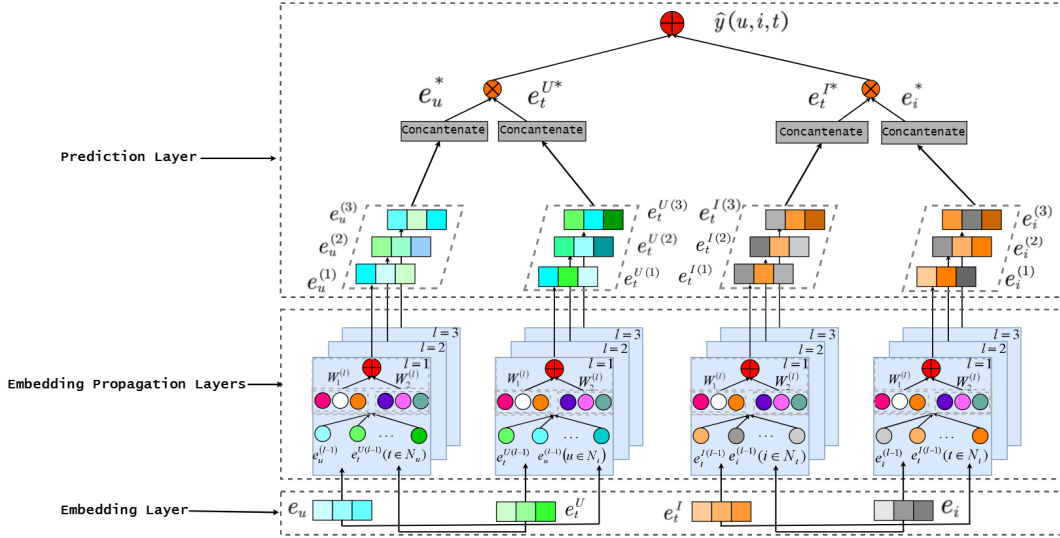
Fig. 1. The framework of our proposed personalized tag recommendation algorithm

IV. THE GRAPH NEURAL NETWORKS BOOSTED PERSONALIZED TAG RECOMMENDATION MODEL

In this section, we present the details of our proposed graph neural networks based personalized tag recommendation model.

A. The Framework of Personalized Tag Recommendation Method Based on GNNs

Figure 1 presents the architecture of our proposed model, which mainly consists of three layers: the embedding layer, the embedding propagation layer and the prediction layer. The main function of each layer described as follows: (1) the embedding layer obtains the embedding representations of users, items and tags based on their IDs; (2) the embedding propagation layer implements messages propagation and messages aggregation; (3) the prediction layer ensembles multiple representations for each type of entities and outputs the predicted score for a (user, item, tag) triplet. In the following sections, we describe the details of each component.

1) Embedding Layer: In the embedding layer, we project users, items and tags into a low-dimensional space according to their IDs. Specifically, a training instance is a quadruple (u, i, t, t'), where u and i denote the indexes of user u and item i, respectively. t and t' are the corresponding positive and negative tag indexes with respect to the post (u, i), respectively. We get the embedded representations of the user u, the item i, the positive tag t and the negative tag t' by the lookup operation over the embedding matrices. Formally,

$$
\begin{aligned}
\mathbf{e}_u &= \mathbf{U}.onehot(u), & \mathbf{e}_i &= \mathbf{I}.onehot(i), \\
\mathbf{e}_t^U &= \mathbf{T}^U.onehot(t), & \mathbf{e}_{t'}^U &= \mathbf{T}^U.onehot(t'), \\
\mathbf{e}_t^I &= \mathbf{T}^I.onehot(t), & \mathbf{e}_{t'}^I &= \mathbf{T}^I.onehot(t'),
\end{aligned}
\tag{7}
$$

where $onehot(.)$ denotes the one-hot encoding operation.

2) Embedding Propagation Layers: The goal of embedding propagation layers is to capture the collaborative signal and enrich the representations of users, items and tags. The collaborative signal is not explicit, which is latent in the interaction among users, items and tags. In the embedding propagation layers, we mainly exploit the GNNs to explicitly capture the collaborative signal among interacted entities.

Generally, in personalized tag recommendation systems, there are three types of interactions, i.e. user-tag interactions, item-tag interactions and user-item interactions. Similar to [9], in this paper, we only consider user-tag interactions and item-tag interactions. For each type of interactions, we employ the message-passing mechanism to capture collaborative signal along the corresponding bipartite, which is derived from their interaction information. In each type of interactions, there are two types of messages that transmit along the interaction graph. Take the user-tag interaction as an example, the propagated messages include the information that propagates from tag node to user node as well as information that propagates from user node to tag node.

Given a user-tag pair (u, t), the propagated messages between the user u and the tag t are defined as follows:

$$
\begin{aligned}
m_{u \leftarrow t} &= p_{ut} \left(W_1 e_t^U + W_2 \left(e_u \odot e_t^U \right) \right) \\
m_{t \leftarrow u} &= p_{tu} \left(W_1 e_u + W_2 \left(e_t^U \odot e_u \right) \right)
\end{aligned}
\tag{8}
$$

where $m_{u \leftarrow t}$ and $m_{t \leftarrow u}$ indicate the messages that transmit from the tag t to the user u and from the user u to the tag t, respectively. And $\odot$ indicates the element-wise product. The p_{ut} and p_{tu} are decay factors that are used to control each message propagation. Formally, p_{ut} and p_{tu} are defined as the Laplacian norm $\frac{1}{\sqrt{|N_u||N_t|}}$, where N_u and N_t represent the first-hop neighbors of the user u and tag t, respectively. The $W_1, W_2 \in \mathbb{R}^{d' \times d}$ are training weight matrices, where d' is the transformation size.

Given the definition of propagation messages as well as the neighborhood structure of one node, we can aggregate the messages to form a new representation for nodes, which explicitly encodes the first-order connectivity between interacted entities. Formally, by assembling the messages that are transmitted by the direct neighbors, the assembled representations for the user u and the tag t are as follows:

$$e_u^{(1)} = LeakyReLU\left(m_{u \leftarrow u} + \sum_{t \in N_u} m_{u \leftarrow t}\right)$$
$$e_t^{U(1)} = LeakyReLU\left(m_{t \leftarrow t} + \sum_{u \in N_t} m_{t \leftarrow u}\right) \quad (9)$$

where $LeakyReLU$ is an activation function [26], which non-linearly transforms the propagated messages. And the $m_{u \leftarrow u}$ and $m_{t \leftarrow t}$ consider the self-connections of the user u and the tag t, respectively.

By assembling the messages propagated from the direct neighbors, the assembled representations $e_u^{(1)}$ and $e_t^{U(1)}$ explicitly consider the first-order connectivity information. In order to further enrich the representations, we inject the high-order connectivity information into the embedded representations of nodes by stacking more embedding propagation layers. In other words, we assemble the messages from high-hop neighbors to generate the representations of users, items and tags. Specifically, with l embedding propagation layers, the assembled representations of the user u and the tag t are formulated as:

$$e_u^{(l)} = LeakyReLU\left(m_{u \leftarrow u}^{(l)} + \sum_{t \in N_u} m_{u \leftarrow t}^{(l)}\right)$$
$$e_t^{U(l)} = LeakyReLU\left(m_{t \leftarrow t}^{(l)} + \sum_{u \in N_t} m_{t \leftarrow u}^{(l)}\right) \quad (10)$$

where $m_{* \leftarrow \star}^{(l)}$ denotes the message that is propagated from their corresponding l-hop neighbors. Formally,

$$\begin{cases} m_{u \leftarrow t}^{(l)} = p_{ut}\left(W_1^{(l)} e_t^{U(l-1)} + W_2^{(l)}\left(e_u^{(l-1)} \odot e_t^{U(l-1)}\right)\right) \\ m_{u \leftarrow u}^{(l)} = W_1^{(l)} e_u^{(l-1)} \end{cases}$$
$$\begin{cases} m_{t \leftarrow u}^{(l)} = p_{ut}\left(W_1^{(l)} e_t^{U(l-1)} + W_2^{(l)}\left(e_u^{(l-1)} \odot e_t^{U(l-1)}\right)\right) \\ m_{t \leftarrow t}^{(l)} = W_1^{(l)} e_t^{U(l-1)} \end{cases}$$
$$(11)$$

where $W_1^{(l)}, W_2^{(l)} \in R^{d_l \times d_{l-1}}$ are the weight transformation matrices, and the d_l is transformation size. The $e_u^{(l-1)}$ and $e_t^{U(l-1)}$ are the embedded representations that are obtained at the $(l-1)$th embedding propagation layer.

So far, we have described how to stack multiple embedding propagation layers to capture the collaborative signal between users and tags. Similarly, we adopt the similar architecture to deal with the item-tag interaction information, and capture the collaborative signal between items and tags by propagating and assembling embedded representations of neighbors of items or tags. In this way, we enrich the representations of items and tags by exploiting the connectivity information encoded in the item-tag interactions.

3) Prediction Layer: By stacking multiple embedding propagation layers, we obtain the set of embedded representations of users, items and tags:

$$\begin{aligned} &\left\{e_u^{(1)}, e_u^{(2)}, \cdots, e_u^{(l)}\right\} \\ &\left\{e_i^{(1)}, e_i^{(2)}, \cdots, e_i^{(l)}\right\} \\ &\left\{e_t^{U(1)}, e_t^{U(2)}, \cdots, e_t^{U(l)}\right\} \\ &\left\{e_t^{I(1)}, e_t^{I(2)}, \cdots, e_t^{I(l)}\right\} \end{aligned} \quad (12)$$

For each entity, the element $e_*^{(l)}$ is the output of embedding propagation layer that assembles messages propagated from the l-hop neighbors. Hence, different element of one set focuses on different order of connectivity information, and characterizes different aspect of users' preferences, items' and tags' characteristics. For each entity, since each element has contributions to the embedded representations of the entity, we concatenate all elements to get the final representation for the entity,

$$\begin{aligned} e_u^* &= e_u^{(1)} || e_u^{(2)} || \cdots || e_u^{(l-1)} || e_u^{(l)} \\ e_i^* &= e_i^{(1)} || e_i^{(2)} || \cdots || e_i^{(l-1)} || e_i^{(l)} \\ e_t^{U*} &= e_t^{U(1)} || e_t^{U(2)} || \cdots || e_t^{U(l-1)} || e_t^{U(l)} \\ e_t^{I*} &= e_t^{I(1)} || e_t^{I(2)} || \cdots || e_t^{I(l-1)} || e_t^{I(l)} \end{aligned} \quad (13)$$

where $||$ is the concatenation operation.

In the way, the final representations of entities is endowed with rich semantics, which include both low-order and high-order connectivity information and capture the collaborative signal among interacted entities. Hence, the final representation scheme could increase the expressiveness of entity embeddings.

Based on the final representations of users, items and tags, we also explicitly model the pairwise interaction between users, items and tags, which is similar to the PITF [9]. Given a triplet (u, i, t), the predicted score $\hat{y}_{u,i,t}$ is computed as:

$$\hat{y}_{u,i,t} = \sum_k e_{u,f}^* \cdot e_{t,f}^{U*} + \sum_k e_{i,f}^* \cdot e_{t,f}^{I*} \quad (14)$$

where k is the dimension of the final representations of entities.

B. Model Parameters Learning

We adopt the widely used ranking optimization criterion, i.e. the bayesian personalized ranking [11], to learn the model parameters of our proposed graph neural networks boosted tag recommendation model. The objective function of our proposed method is defined as follows:

$$\mathcal{L} = \min_{\Phi} \sum_{(u,i,t,t') \in D_S} -\ln \sigma(\hat{y}_{u,i,t} - \hat{y}_{u,i,t'}) + \lambda_\Phi ||\Phi||_F^2 \quad (15)$$

where (u, i, t, t') is the training data, which include two instances, i.e. a positive instance (u, i, t) and a negative instance (u, i, t'). And $\Phi = \{\mathbf{U}, \mathbf{I}, \mathbf{T}^U, \mathbf{T}^I, W_1^{(i)}, W_2^{(i)}, i = 1, 2, ..., l\}$ is the model parameters. λ_Φ denotes regularization coefficient that controls the effect of the regularization terms. In addition, we adopt the mini-batch Adam optimizer to optimize the objective function $\mathcal{L}$.

V. EMPIRICAL ANALYSIS

In this section, we conduct several groups of experiments on two real-world datasets to compare the performance of our proposed personalized tag recommendation method with other state-of-the-art methods.

A. Dataset

In our experiments, we choose two public available datasets, i.e. Last.fm and ML10M [1], to evaluate the performance of our proposed tag recommendation algorithm. Similar to [9], [17], we preprocess each dataset to get their corresponding p-core, which is the largest subset with the property that every user, every item and every tag has to occur at least p times. In our experiments, all datasets are 5-core and 10-core. The general statistics of datasets are summarized in Table I.

TABLE I
DESCRIPTION OF DATASETS

Dataset	#Users	#Items	#Tags
lastfm-core5	1348	6927	2132
lastfm-core10	966	3870	1024
ml-10m-core5	990	3247	2566
ml-10m-core10	469	1524	1017

B. Evaluation Metrics

We adopt the common evaluation protocol, which is widely used in [9], [17]. Specifically, for each user, we randomly select one post and remove the triples that related to the selected post from S to S_{test}. The remaining observed user-item-tag triples are the training set $S_{train} := S \backslash S_{test}$. Similar to the classic item recommendation problem, the personalized tag recommendation provides a top-N highest ranked list of tags for a (user, item) pair. Hence, we employ two widely used ranking metrics to measure the tag recommendation performance of all compared methods, i.e., Precision@N and Recall@N, where N denotes the length of ranked tag recommendation list. Formally,

$$Prec@N := \frac{1}{|P_{S_{test}}|} \sum_{(u,i) \in P_{S_{test}}} \frac{|Top(u, i, N) \cap \{t|(u, i, t) \in S_{test}\}|}{N}$$

$$Rec@N := \frac{1}{|P_{S_{test}}|} \sum_{(u,i) \in P_{S_{test}}} \frac{|Top(u, i, N) \cap \{t|(u, i, t) \in S_{test}\}|}{|\{t|(u, i, t) \in S_{test}\}|}$$

where $|P_{S_{test}}|$ is the number of posts then are included in the test set S_{test}. For both metrics, we set $N = 3, 5, 10, 20$ to evaluate the performance in our experiments.

[1] Two datasets can be found in https://grouplens.org/datasets/hetrec-2011/

C. Experimental Settings

We choose the following traditional tag recommendation algorithms as baselines:

- PITF: PITF [9] was proposed by Rendle and Steffen. It explicitly models the pairwise interaction between users, items and tags, and is a strong competitor in the field of personalized tag recommendation.
- NLTF: NLTF [10] was proposed by Fang et al. It is a non-linear tensor factorization model, which enhances the PITF by exploiting the Gaussian radial basis function to capture the non-linear interaction relations among users, items and tags.
- ABNT: ABNT [20] was proposed by Yuan et al. It utilizes the multi-layer perceptron to model the non-linearities of interactions between users, items and tags.

To make a fair comparison, we set the parameters of each model based on respective references or based on our experiments, such that the recommendation performance of the compared models is optimal under these parameters. For all compared methods, the dimension of latent factor vector d is tuned amongst $\{8, 16, 32, 64, 128, 256\}$. The mini-batch size is selected from $\{512, 1024, 2048\}$ and the learning rate is tuned amongst $\{0.001, 0.005, 0.01\}$. The regularization coefficient is chosen from $\{0.001, 0.005, 0.01, 0.05\}$. All latent factor vectors and parameters are randomly initialized using the Gaussian distribution with mean of 0 and standard deviation of 0.01. For most datasets and baselines, we empirically set the dimension of latent factor vector d with 64, the number of batch is 512, the learning rate is set to 0.001, and the regularization coefficient is 0.01. For the ABNT, the structure of multi-layer perceptron follows the tower structure, i.e. the dimension of hidden layer is half of that of the previous hidden layer. For our proposed method, we set the number of embedding propagation layers $l = 3$.

D. Performance Comparison

Tables II, III, IV, V report the tag recommendation quality of all compared methods on four datasets.

TABLE II
PERFORMANCE COMPARISONS ON LASTFM-CORE5

Method	PITF	NLTF	ABNT	**GNN-PTR**
Pre@3	0.21266	0.19486	0.15628	**0.23244**
Pre@5	0.17893	0.16780	0.13531	**0.19125**
Pre@10	0.12737	0.11907	0.10178	**0.13272**
Pre@20	0.08323	0.07986	0.06996	**0.08468**
Rec@3	0.25711	0.22753	0.15691	**0.32444**
Rec@5	0.34786	0.32389	0.21940	**0.41697**
Rec@10	0.48138	0.45230	0.32984	**0.54541**
Rec@20	0.60074	0.57657	0.44266	**0.65544**

From Table II to Table V, we have the following observations: (1) On four datasets, PITF achieves a better performance than NTLF and ABNT, which demonstrates the strong competitiveness of PITF model. On the other hand, the observation also indicates that integrating the multi-layer

perceptron into PITF framework cannot guarantee improvements of tag recommendation quality, although ABNT is built upon the PITF. One possible reason is that the ABNT involves more trainable parameters, whereas train data available is insufficient for learning its model parameters. (2) For each compared method, its recommendation performance is better on the core-10 datasets than that on the corresponding core-5 datasets. This observation indicates that increasing the density of datasets could boost the tag recommendation performance. (3) Our proposed graph neural networks based personalized tag recommendation method consistently outperforms other methods, which demonstrates the effectiveness of our proposed method. In terms of precision@3, our proposed GNN-PTR model improves the PITF by 9.3% and 4.1% on last.fm-core5 and ml-10m-core5, respectively. In terms of precision@5, the improvements of GNN-PTR over PITF are 2.7% and 18.6% on last.fm-core10 and ml-10-core10, respectively. To some extent, the improvements are considerable. Hence, this observation confirms that integrating the collaborative signal into the learning of embeddings in an explicitly manner is beneficial for the personalized tag recommendation model.

TABLE III
PERFORMANCE COMPARISONS ON LASTFM-CORE10

Method	PITF	NLTF	ABNT	GNN-PTR
Pre@3	0.25132	0.24431	0.16406	**0.26467**
Pre@5	0.20875	0.20624	0.13665	**0.21429**
Pre@10	0.14577	0.12493	0.09413	**0.14617**
Pre@20	0.08931	0.08205	0.06796	**0.09224**
Rec@3	0.32035	0.28448	0.15792	**0.34791**
Rec@5	0.41583	0.40170	0.21895	**0.45288**
Rec@10	0.56539	0.55412	0.30336	**0.58738**
Rec@20	0.69311	0.68562	0.45190	**0.71441**

TABLE IV
PERFORMANCE COMPARISONS ON ML-10M-CORE5

Method	PITF	NLTF	ABNT	GNN-PTR
Pre@3	0.13976	0.13232	0.08215	**0.14545**
Pre@5	0.10206	0.09717	0.06283	**0.10545**
Pre@10	0.06414	0.05960	0.04000	**0.06717**
Pre@20	0.03768	0.03667	0.02470	**0.04046**
Rec@3	0.32077	0.29738	0.20888	**0.33312**
Rec@5	0.39096	0.35602	0.25378	**0.39653**
Rec@10	0.46230	0.42697	0.30388	**0.48516**
Rec@20	0.52332	0.51305	0.36596	**0.57213**

TABLE V
PERFORMANCE COMPARISONS ON ML-10M-CORE10

Method	PITF	NLTF	ABNT	GNN-PTR
Pre@3	0.16986	0.14357	0.08955	**0.19332**
Pre@5	0.11725	0.11429	0.07591	**0.13902**
Pre@10	0.07443	0.07143	0.05011	**0.08422**
Pre@20	0.04479	0.04382	0.03369	**0.04989**
Rec@3	0.37704	0.33881	0.22100	**0.46023**
Rec@5	0.45230	0.43344	0.30147	**0.54606**
Rec@10	0.52050	0.53408	0.38579	**0.63980**
Rec@20	0.60167	0.63966	0.50586	**0.73557**

E. Impact of The Number of Embeddings Propagation Layers

In our proposed method, the number of embedding propagation layers l is an important parameter that affects the tag recommendation performance of our proposed model. In this section, we conduct a group of experiments to explore the effect of l on tag recommendation performance by varying the value of l from 1 to 4. Other parameters keep the same settings as described in Section V-C. The experimental results in terms of precision@10 on lastfm-core10 and ml-10-core10 are shown in Figure 2.
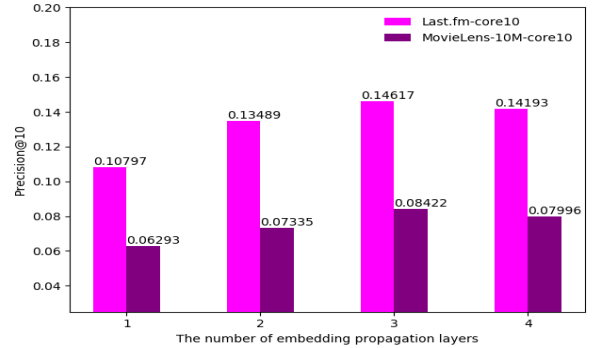

Fig. 2. Impact of the number of embedding propagation layers

As shown in Fig. 2, our proposed tag recommendation model is sensitive to the value of l. With the number of embedding propagation layers increases, the $Precision@10$ of GNN-PTR firstly increases. Then, if the number of embedding propagation layers continues to increase and surpasses a threshold value, the performance of the proposed model begins to degrade. The possible reason is that: a large value of l makes our proposed method leverage the collaborative signal that is propagated from the relative distant neighbors. Intuitively, the collaborative signal of the distant neighbors may not be helpful for enriching the representation of target entities since the correlations between entity and their distant neighbors are weak. When the number of embedding propagation layers $l = 3$, our proposed personalized tag recommendation method achieves the best performance.

F. Impact of The Dimension of Latent Feature Vectors

In this section, we vary the dimension of the hidden feature vectors d in [16, 32, 64, 128, 256] , and investigate the impact of parameter d on tag recommendation quality. Other parameters remain unchanged. We only plot the precision@10 of GNN-PTR on lastfm-core10 and ml-10m-core10 in Fig. 3. Other metrics show similar trends.

As we can see, the dimension of latent feature vectors d also affects the recommendation performance of GNN-PTR. In the early stage, the recommendation performance of GNN-PTR is constantly improving as the value of d increases. Then, when the value of d reaches to 128, the curve of precision@10 remains stable and the tag recommendation performance does not further improve as we further increase the value of d. This is because that if the latent feature vectors can capture the
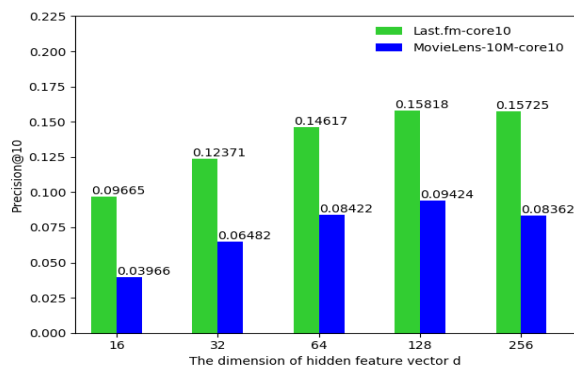
Fig. 3. Impact of parameter d.

interacted entities' preferences or characteristics effectively, further increasing the value of d could not enhance the representation capacity of our proposed model. Our proposed recommendation method achieves its best performance when d is equal to 128.

VI. CONCLUSION

Traditional personalized tag recommendation methods ignore the collaborative signal in the process of learning representation of entities, leading to the lack of expressive ability for characterizing the preferences or attributes of entities. In this paper, we proposed a graph neural networks boosted personalized tag recommendation model, which integrates the graph neural networks into the pairwise interaction tensor factorization model. Based on the user-item-tag interaction triples, we consider two types of interactions, i.e. the user-tag interactions and the item-tag interactions. We exploit the graph neural networks to capture the collaborative signal between interacted entities as well as integrate the collaborative signal into the learning of representations of entities by performing messages propagation over the entity interaction graphs. Experimental results show that our proposed method outperforms the state-of-the-art personalized tag recommendation methods.

REFERENCES

[1] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *TKDE*, no. 6, pp. 734–749, 2005.

[2] A. Hotho, R. Jäschke, C. Schmitz, and G. Stumme, "Information retrieval in folksonomies: Search and ranking," in *European Semantic Web Conference*. Springer, 2006, pp. 411–426.

[3] R. Jäschke, L. Marinho, A. Hotho, L. Schmidt-Thieme, and G. Stumme, "Tag recommendations in folksonomies," in *European Conference on Principles of Data Mining and Knowledge Discovery*. Springer, 2007, pp. 506–514.

[4] R. Krestel, P. Fankhauser, and W. Nejdl, "Latent dirichlet allocation for tag recommendation," in *RecSys*. ACM, 2009, pp. 61–68.

[5] Y. Wu, Y. Yao, F. Xu, H. Tong, and J. Lu, "Tag2word: Using tags to generate words for content based tag recommendation," in *CIKM*. ACM, 2016, pp. 2287–2292.

[6] S. Tang, Y. Yao, S. Zhang, F. Xu, T. Gu, H. Tong, X. Yan, and J. Lu, "An integral tag recommendation model for textual content," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 5109–5116.

[7] B. Sun, Y. Zhu, Y. Xiao, R. Xiao, and Y. Wei, "Automatic question tagging with deep neural networks," *IEEE Transactions on Learning Technologies*, vol. 12, no. 1, pp. 29–43, 2018.

[8] P. Symeonidis, A. Nanopoulos, and Y. Manolopoulos, "Tag recommendations based on tensor dimensionality reduction," in *RecSys*. ACM, 2008, pp. 43–50.

[9] S. Rendle and L. Schmidt-Thieme, "Pairwise interaction tensor factorization for personalized tag recommendation," in *WSDM*. ACM, 2010, pp. 81–90.

[10] X. Fang, R. Pan, G. Cao, X. He, and W. Dai, "Personalized tag recommendation through nonlinear tensor factorization using gaussian kernel," in *AAAI*, 2015.

[11] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "Bpr: Bayesian personalized ranking from implicit feedback," in *UAI*, 2009, pp. 452–461.

[12] J. Zhou, G. Cui, Z. Zhang, C. Yang, Z. Liu, and M. Sun, "Graph neural networks: A review of methods and applications," *ArXiv*, 2018.

[13] Y. Qian, P. Zhao, Z. Li, J. Fang, L. Zhao, V. S. Sheng, and Z. Cui, "Interaction graph neural network for news recommendation," in *WISE*. Springer, 2019, pp. 599–614.

[14] W. Fan, Y. Ma, Q. Li, Y. He, E. Zhao, J. Tang, and D. Yin, "Graph neural networks for social recommendation," in *WWW*. ACM, 2019, pp. 417–426.

[15] S. Wu, Y. Tang, Y. Zhu, L. Wang, X. Xie, and T. Tan, "Session-based recommendation with graph neural networks," in *AAAI*, vol. 33, 2019, pp. 346–353.

[16] X. Wang, X. He, M. Wang, F. Feng, and T.-S. Chua, "Neural graph collaborative filtering," in *SIGIR*, 2019, pp. 165–174.

[17] S. Rendle, L. Balby Marinho, A. Nanopoulos, and L. Schmidt-Thieme, "Learning optimal ranking with tensor factorization for tag recommendation," in *KDD*. ACM, 2009, pp. 727–736.

[18] L. De Lathauwer, B. De Moor, and J. Vandewalle, "A multilinear singular value decomposition," *SIAM journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1253–1278, 2000.

[19] Y. Cai, M. Zhang, D. Luo, C. Ding, and S. Chakravarthy, "Low-order tensor decompositions for social tagging recommendation," in *WSDM*. ACM, 2011, pp. 695–704.

[20] J. Yuan, Y. Jin, W. Liu, and X. Wang, "Attention-based neural tag recommendation," in *International Conference on Database Systems for Advanced Applications*. Springer, 2019, pp. 350–365.

[21] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, "Graph convolutional neural networks for web-scale recommender systems," in *KDD*. ACM, 2018, pp. 974–983.

[22] R. van den Berg, T. Kipf, and M. Welling, "Graph convolutional matrix completion," *ArXiv*, 2017.

[23] L. Wu, P. Sun, R. Hong, Y. Fu, X. Wang, and M. Wang, "Socialgcn: An efficient graph convolutional network based model for social recommendation," *ArXiv*, 2018.

[24] R. Pan, Y. Zhou, B. Cao, N. N. Liu, R. Lukose, M. Scholz, and Q. Yang, "One-class collaborative filtering," in *ICDM*. IEEE, 2008, pp. 502–511.

[25] Y. Hu, Y. Koren, and C. Volinsky, "Collaborative filtering for implicit feedback datasets," in *ICDM*. IEEE, 2008, pp. 263–272.

[26] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," *ArXiv*, 2017.