Noname manuscript No.
(will be inserted by the editor)

Personalized Tag Recommendation via Denoising Auto-Encoder

Weibin Zhao · Lin Shang · Yonghong Yu · Li Zhang · Can Wang · Jiajun Chen

Received: date / Accepted: date

Abstract Personalized tag recommender systems automatically recommend users a set of tags used to annotate items according to users' past tagging information. Learning the representations of involved entities (i.e. users, items and tags) and capturing the complex relationships among them are crucial for personalized tag recommender systems. However, few studies have been conducted to simultaneously achieve these two sub-goals. In this research, we propose a novel personalized tag recommendation model based on the denoising auto-encoder, namely DAE-PTR, which learns the representations of entities and encodes the complex relationships by exploiting the denoising auto-encoder framework. Specifically, for each user, we firstly generate the corrupted version of the respective tagging information by adding the multiplicative mask-out/drop-out noise into the original input. Then, we learn the latent representations from the corrupted input via the auto-encoder frame-

Weibin Zhao
State Key Lab for Novel Software Technology, Nanjing University, P.R.China
E-mail: njzhaowb@gmail.edu.cn

Lin Shang
State Key Lab for Novel Software Technology, Nanjing University, P.R.China
E-mail: shanglin@nju.edu.cn

Yonghong Yu
TongDa College, Nanjing University of Posts and Telecommunications, P.R.China
E-mail: yuyh@njupt.edu.cn

Li Zhang
Department of Computer Science, Royal Holloway, University of London, UK
E-mail: li.zhang@rhul.ac.uk

Can Wang
School of Information and Communication Technology, Griffith University, Australia
E-mail: can.wang@griffith.edu.au

Jiajun Chen
State Key Lab for Novel Software Technology, Nanjing University, P.R.China
E-mail: chenjj@nju.edu.cn

work by using the cross-entropy loss. More importantly, we integrate the latent user and item embeddings into the processing of encoding, which makes the learnt hidden representations of the auto-encoder network encode multiple types of relationships among entities, i.e. the relationships between users and tags, between items and tags, and among tags. Finally, we employ the decoder component to reconstruct the original input based on the learnt latent representations. Experimental results on the real-world datasets show that our proposed DAE-PTR model is superior to the traditional personalized tag recommendation models.

Keywords Recommender Systems · Auto-Encoder · Personalized Tag Recommendation

1 Introduction

In the era of big data, users are faced with serious information overload issue. In order to alleviate the problem of information overload, recommender systems [1] provide users with the personalized information, products and services by revealing the latent preferences of users from their past activities, and have played a significant role in E-commerce, online news and social media sites. Recently, the research focus of recommender systems has shifted from rating prediction [5,38,26,23,37] to top-n recommendation [33,30,19] due to the fact that the task of top-n recommendation is more practical than rating prediction. In addition, with the advances of machine learning, such as active learning [49, 13], representation learning [3,48] and meta learning [27] etc., their application in the field of recommender system has received great attention in industry and academia.

Personalized tag recommender systems [18,20,41,35] is a branch of top-n recommendation, which automatically recommend users a ranked list of tags as comments on items based on users' preferences. For example, collaborative tagging systems [18,20] recommend related keywords (i.e. tags) to users for annotating items (e.g. photos, songs and movies). Personalized tag recommendation are beneficial for both users and web applications. From the users' perspective, personalized tag recommender systems help to improve user experience in the process of tagging. From the web applications' perspective, it help to efficiently manage and search related resources. According to whether user preferences are considered, tag recommendation can be roughly divided into non-personalized tag recommendation and personalized tag recommendation. The non-personalized tag recommendation systems [24,52,42,40] provide all users with the same tags for a certain item, while the personalized tag recommendation systems [18,20,41,35] provide the personalized tag list for each user based on users' preferences. However, the personalized tag recommendation is more practical for the real-world tag recommendation scenarios since different users usually have different habits. Representative personalized tag recommended methods include PITF [35], NLTF [14] and RTF [32] etc., which are basically built on tensor factorization.

Although these classical models advance the research of personalized tag recommendation, most of them focus on learning the latent representations of users, items and tags. Moreover, they ignore the relationship modeling among entities, including relationships between users and tags, relationships between items and tags, and relationships among tags. For example, traditional personalized tag recommendation models cannot guarantee that the relationships among tags are effectively captured, while these relationships are curial for personalized tag recommendation since similar tags have similar semantics and can be used to annotate similar items.

Recently, many studies have employed deep neural network to design recommendation models, because deep learning techniques are able to extract deep and abstract features for users and items, leading to promising improvements in recommendation quality. Among these neural network techniques, the auto-encoder is an effective representation learning technique, which is able to reconstruct inputs in the output layer via a low-dimensional hidden space. In the recommendation field, some auto-encoder based recommendation models have also been proposed, including AutoRec [39], CDAE [51], DCF [25] and RSDAE [45], However, most related studies have only applied the auto-encoder to the item recommendation task, and few studies have been conducted to explore the auto-encoder techniques for personalized tag recommendation task.

In this research, we propose a novel personalized tag recommendation model based on the denoising auto-encoder, namely DAE-PTR, which simultaneously learns the representations of entities and encodes the complex relationships in a unified framework via the denoising auto-encoder technique. Specifically, for each user, we firstly generate the corrupted version of the respective tagging information by adding the multiplicative mask-out/drop-out noise into the original input. Then, we learn the latent representations from the corrupted input via the auto-encoder framework by using the cross-entropy loss. More importantly, we integrate the latent user and item embeddings into the processing of encoding, which makes the learnt hidden representations of the auto-encoder network encode multiple types of relationships among entities, i.e. the relationships between users and tags, between items and tags, and among tags. Then, we employ the decoder component to reconstruct the original input based on the learnt latent representations. The empirical results indicate that the DAE-PTR model consistently outperforms traditional personalized tag recommendation models.

The main contributions of this research are summarized as follows:

- We propose an auto-encoder-based personalized tag recommendation model, which utilizes the denoising auto-encoder framework to simultaneously learns the representations of entities and captures the complex relationships in an unified framework.
- By integrating the latent user and item embeddings into the processing of encoding, we can capture multiple types of relationships among entities within the learnt hidden representations of the auto-encoder network.

– We conduct extensive experiments on real-world datasets to evaluate the effectiveness of our proposed denoising auto-encoder-based personalized tag recommendation model. Experimental results show that our proposed model is superior to existing state-of-the-art personalized tag recommendation models.

The rest of this paper is organized as follows. Section 2 briefly reviews the related work for personalized tag recommendation. Section 3 introduces some preliminary knowledge. Section 4 describes the details of the proposed tag recommendation model. Experimental studies are presented in Section 5. Finally, we conclude this research and provide several directions for future work in Section 6.

2 Related Work

In this section, we briefly review related studies, including personalized tag recommendation models and auto-encoder-based item recommendation models.

2.1 Personalized Tag Recommendation Models

Since the tagging information naturally can be represented by a 3-order tensor, most of existing personalized tag recommendation models are built on tensor factorization techniques, especially the tucker decomposition (TD) model. For instance, Symeonidis et al. [41] developed a unified framework to model three types of entities (i.e. users, items and tags). The 3-dimensional data is represented by a 3-order tensor. Then, they utilized the higher order singular value decomposition (HOSVD) technique [12] to reveal the latent semantic associations among entities. Cai et al. [6] proposed the lower-order tensor decomposition (LOTD) for personalized tag recommendation, which utilizes low-order polynomials to enhance statistics and avoids over-fitting. Unlike other tensor factorization-based methods that optimize the square loss between real values and predicted values, Rendle et al. [32] proposed the ranking with tensor factorization model, namely RTF, which directly optimizes the factorization model for the best personalized ranking. RTF handles missing values and learns user preferences from pairwise ranking constraints. However, the computation cost of TD used in both HOSVD and RTF makes them infeasible for large-scale personalized tag recommender systems since the model equation of TD-based methods is the third power of the factorization dimension. In addition, PITF [35] utilizes the pairwise interactions among users, items, and tags to model user preferences, and adopts the BPR [33] optimization criteria to learn its model parameters. Moreover, PITF has linear runtime both for learning and prediction, which allows PITF to make personalized tag recommendation on large-scale datasets. Different from the linear PITF model, NLTF [14] can be considered as a nonlinear extension, which exploits Gaussian radial basis function to increase model's capacity.

In addition, some researchers proposed the graph-based personalized tag recommendation methods, which exploit the interaction graph to boost the recommendation performance. For example, Guan et al. [15] modeled personalized tag recommendation as a "query and ranking" problem and proposed a novel graph-based ranking algorithm for interrelated multi-type objects. Recently, in order to take the advantages of deep learning techniques, researchers also proposed a few neural-network-based methods to improve the performance of traditional personalized tag recommendation models. In [28], Nguyen et al. proposed a personalized deep learning approach for image tag recommendation that considers both users' preferences and visual information. Based on the assumption that the contents of images (e.g. the objects appearing in the image, colors, shapes or other visual aspects) strongly influence users' tagging decisions, Nguyen et al. [29] proposed a personalize content-aware image tag recommendation approach, which combines historical tagging information and image-based features in a factorization model. In [53], Yuan et al. proposed the ABNT, which utilizes the multi-layer perceptron to model the non-linearities of interactions between users, items and tags, and leverages an attention network to capture the complex pattern of users' tagging sequence. More recently, Erik et al. [31] proposed an end-to-end deep network for preference-aware tag recommendation model, which uses an encoder-decoder network to learn user preferences in an unsupervised way. Moreover, they integrated the visual preference with tagging behavior by jointly training user-preference and visual encoding. In [10], Chen et al. proposed a graph neural networks boosted personalized tag recommendation model, which integrates the graph neural networks into the pairwise interaction tensor factorization model.

2.2 Auto-encoder based Recommendation Models

The auto-encoder neural network [4] is one of core components of deep learning, and has shown great potential in various machine learning fields, such as natural language processing [7,50] and computer vision [47,17]. Essentially, the auto-encoder neural network is an unsupervised learning model that aims at learning effective and compact representations, which are used by the downstream machine learning tasks. Besides the basic auto-encoder model (AE) [4], various variants of AE have also been proposed in recent years, such as the denoising auto-encoder (DAE) [43], the stacked denoising auto-encoder (SDAE) [44] and the contractive auto-encoder [36].

Recently, the auto-encoder models have also received increasing attention in the field of item recommendation. As an example, Sedhain et al. [39] proposed a collaborative filtering model based on the auto-encoder paradigm, namely AutoRec. In addition, in order to take the advantages of DAE [43], several DAE-based item recommendation models were developed. For instance, Li et al. [25] proposed the deep collaborative filtering framework, namely DCF, which integrates the probabilistic matrix factorization [37] and marginalized denoising auto-encoder [9]. Specifically, the DCF learns the latent factors of users and items from the ratings and auxiliary information through marginalized denoising auto-encoder. Wu et al. [51] proposed the Collaborative Denoising Auto-Encoder (CDAE), which formulates the top-N recommendation problem using the auto-encoder framework. Dai et al. [11] proposed a neural network model for context-aware citation recommendation by combining SDAE and Bi-LSTM. Unlike the above models that adopt the auto-encoder frameworks to model users' interaction behaviors, Wang et al. [46] utilizes S-DAE to model the auxiliary information, and proposed a hierarchical Bayesian model, called CDL. The CDL model jointly learns the deep representation of item content by using the SDAE model and performs the probabilistic matrix factorization on the user-item rating matrix. In addition, some researchers have proposed several variational auto-encoder [22] based item recommendation models. For instance, Zheng et al. [54] proposed an implicit trust relation-aware model based on variational auto-encoder for social recommendation. Bahare et al. [2] presented the joint variational auto-encoder model, which jointly learns both users and items representations to predict user preferences.

All aforementioned models focus on the item recommendation task, and few studies have explored the application of the auto-encoder techniques for the tag recommendation, except for the RSDAE [45], which jointly performs deep representation learning and relational learning in principle way under a probabilistic framework. Unlike the RSDAE that tackles the problem of non-personalized tag recommendation by utilizing the SDAE model, in this research, we employ the DAE model for the personalized tag recommendation task. In addition, the work most relevant to our proposed method is CDAE, which formulates the top-N recommendation problem using the Auto-Encoder framework. The key difference between CDAE and DAE-PTR lies in that C-DAE focuses on the item recommendation task, while DAE-PTR focuses on personalized tag recommendation task. And the personalized tag recommendation is more challenging than item recommendation task since it is crucial for personalized tag recommender systems to capture the complex relationships among involved entities, such as the relationships between users and items, the relationships between users and tags as well as the relationships between items and tags.

3 Preliminary Knowledge

In this section, we introduce preliminary knowledge pertaining to the proposed model. We first describe the problem of personalized tag recommendation in Section 3.1. Then, we briefly recall the denoising auto-encoder model in Section 3.2.

3.1 Problem Description

Typically, personalized tag recommender systems include three types of entities: the set of users U, the set of items I and the set of tags T. The interaction information between users, items and tags is represented as $S \subseteq U \times I \times T$. A ternary $(u, i, t) \in S$ indicates that the user u has annotated the item i with the tag t.

From the ternary relation set S, personalized tag recommendation models usually deduce a three-order tensor $Y \in \mathbb{R}^{|U|\times|I|\times|T|}$, whose element $y_{u,i,t}$ is defined as follows:

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

where $y_{u,i,t} = 1$ indicates a positive instance, and the remaining data are the mixture of negative instances and missing values. In addition, the tagging information for the user-item pair (u, i) is defined as $\mathbf{y}_{u,i} = \{y_{u,i,t} | y_{u,i,t}, t \in T\}$.

Personalized tag recommender systems aim at recommending a ranked list of tags to a user for annotating an item. Usually, they design a score function $\widehat{Y} : U \times I \times T \longrightarrow \mathbb{R}$ to model the interaction behaviors among users, items and tags. The entry $\widehat{y}_{u,i,t}$ of $\widehat{Y}$ indicates the probability of the user u annotates the item i with the tag t. After predicting the score $\widehat{y}_{u,i,t}$ for all candidate tag t given a user-item post (u, i), the personalized tag recommender system returns a ranked list of Top-N tags in terms of the obtained scores. Formally, the ranked list of Top-N tags given to the user-item pair (u, i) is defined as follows:

$$Top(u, i, N) = \underset{t \in T}{\overset{N}{argmax}}\ \widehat{y}_{u,i,t}, \tag{2}$$

where N denotes the number of recommended tags.

3.2 Denoising Auto-Encoder Model

The basic AE model [4] is designed to reconstruct high-dimensional data using a neural network with a narrow hidden layer, which is able to capture meaningful information about its input. Architecturally, the basic AE model consists of two sub-networks, i.e. the encoder network and the decoder networks, which maps the high-dimensional input into a low-dimensional hidden representation and recovers the input from the hidden representation, respectively.

The underlying learning mechanism of the AE model is as follows: given an input vector $\mathbf{x} \in \mathbb{R}^d$, the encoder network maps it to a hidden representation $\mathbf{z} \in \mathbb{R}^k$ through a deterministic mapping function $f_\theta(\mathbf{x})$, which is usually an affine mapping followed by a non-linearity and is parameterized by $\theta = \{\mathbf{W}, \mathbf{b}\}$. Formally,

$$\mathbf{z} = f_\theta(\mathbf{x}) = \sigma(\mathbf{W}\mathbf{x} + \mathbf{b}), \tag{3}$$

where $\mathbf{W} \in \mathbb{R}^{k \times d}$ is a weight matrix, and $\mathbf{b} \in \mathbb{R}^k$ is a bias vector. σ is an activate function. Then, the resulting hidden representation $\mathbf{z}$ is mapped back a reconstructed d-dimensional vector $\hat{\mathbf{x}}$ in the input space by using the decoder network, i.e. $\hat{\mathbf{x}} = g_\phi(\mathbf{z})$. Typically, the mapping function $g_\phi()$ is also an affine mapping followed by a squashing non-linearity. Formally,

$$\hat{\mathbf{x}} = g_\phi(\mathbf{z}) = \sigma(\mathbf{W}'\mathbf{z} + \mathbf{b}'), \tag{4}$$

where $\phi = \{\mathbf{W}', \mathbf{b}'\}$. $\mathbf{W}' \in \mathbb{R}^{d\times k}$ and $\mathbf{b}' \in \mathbb{R}^d$ denote another weight matrix and a bias vector, respectively.

Usually, the AE model learns its model parameters, i.e. $\{\mathbf{W}, \mathbf{b}, \mathbf{W}', \mathbf{b}'\}$ by minimizing the average reconstruction error:

$$\mathcal{L}_{AE} = \frac{1}{n}\sum_{i=1}^{n} \ell(\mathbf{x}^{(i)} - g_\phi(f_\theta(\mathbf{x}^{(i)}))) \tag{5}$$

where ℓ is a loss function, such as the squared loss or the cross-entropy loss.

Although the hidden representation learnt by the basic AE model retains a certain amount of information about its input, the reconstruction criterion of AE is unable to guarantee that the hidden representation contains useful features. In order to extract useful features, the denoising auto-encoder (DAE) [43] reconstructs a clean "repaired" input $\mathbf{x}$ from a corrupted, partially version $\widetilde{\mathbf{x}}$. First, the input $\mathbf{x}$ is partially corrupted in a stochastic manner, i.e. $\mathbf{x} \sim \mathcal{M}_{\mathcal{D}}(\widetilde{\mathbf{x}}|\mathbf{x})$, which defines the mapping from the true data samples to the corrupted ones. Then, similar to the basic AE model, DAE model utilizes a three-layer feed-forward neural network to recover the original input $\mathbf{x}$. Formally, the objective function of DAE is defined as follows,

$$\mathcal{L}_{DAE} = \frac{1}{n}\sum_{i=1}^{n} \ell(\mathbf{x}^{(i)} - g_\phi(f_\theta(\widetilde{\mathbf{x}}^{(i)}))) \tag{6}$$

Compared to the basic AE model, the key difference is that the hidden representation $\mathbf{z}$ learnt by DAE is a deterministic function of $\widetilde{\mathbf{x}}$ rather than $\mathbf{x}$, which implies that the DAE learns a more efficient mapping that is able to extract useful features for denoising.

4 The Denoising Auto-Encoder Based Personalized Tag Recommendation Model

In this section, we present the details of the proposed DAE-based personalized tag recommendation model. In Section 4.1, we first introduce the framework of the proposed DAE-PTR model and present the process of extracting user latent preferences for tags by utilizing the denoising auto-encoder technique. Then, we explain the process of parameter learning in Section 4.2.

4.1 The Framework of the Denoising Auto-Encoder Based Personalized Tag Recommendation Model

Similar to the standard denoising auto-encoder model, the proposed DAE-based personalized tag recommendation model is also represented as a three-layer feed-forward neural network, consisting of three distinct layers: the input layer, the hidden layer and the output layer. The framework of the proposed DAE-PTR model is presented in Fig. 1. Moreover, our proposed tag recommendation model is made up of three subprocesses: (1) Corrupting the initial input with noise; (2) Encoding the corrupted input into a low-dimensional space; (3) Decoding the low-dimensional representation into a high-dimensional input space.

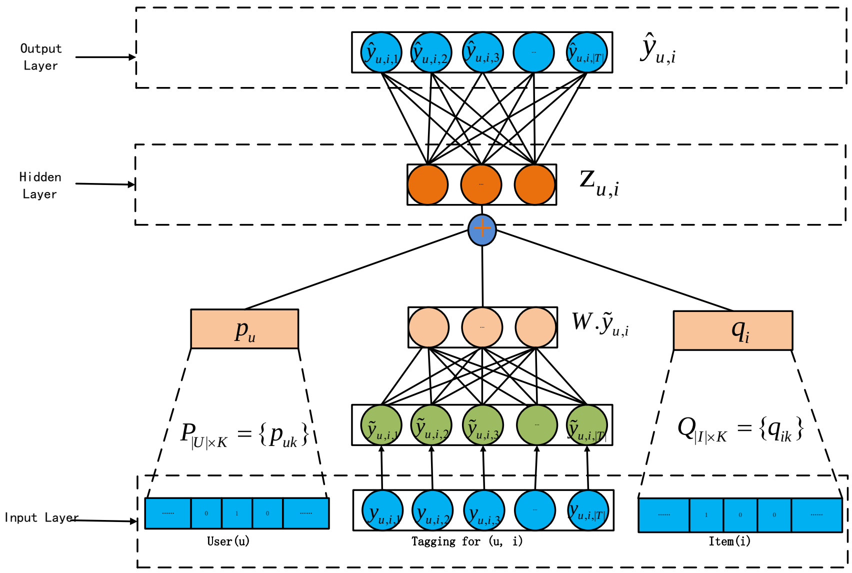


Fig. 1 The framework of our proposed personalized tag recommendation model

4.1.1 Input Layer

The input of the proposed DAE-PTR model is $\mathbf{y}_{u,i}$, whose element $y_{u,i,t} = 1$ indicating that the user u has annotated the item i with the tag t, and vice versa. To obtain the corrupted input $\widetilde{\mathbf{y}}_{u,i}$, we select the multiplicative mask-out/drop-out noise to corrupt the initial input $\mathbf{y}_{u,i}$. In other words, we randomly set each element $y_{u,i,t} = 0$ with a probability c. Formally,

$$\begin{cases} P(\widetilde{y}_{u,i,t} = 0) & = c \\ P(\widetilde{y}_{u,i,t} = y_{u,i,t}) & = 1 - c \end{cases} \tag{7}$$

where c indicates the corruption level. In fact, under the manifold assumption [8], the process of mapping the corrupted input $\widetilde{\mathbf{y}}_{u,i}$ back to the uncorrupted input $\mathbf{y}_{u,i}$ can be viewed as a mechanism of learning a manifold, in which the hidden representation can be interpreted as a coordinate system for points on the manifold as well as capturing the main variation in the inputs along the manifold.

In the input layer, besides the $|T|$ tag nodes, i.e. $\widetilde{\mathbf{y}}_{u,i} = \{\widetilde{y}_{u,i,1}, \widetilde{y}_{u,i,2}, ..., \widetilde{y}_{u,i,|T|}\}$, there are another two nodes, i.e. the user-specific node u and the item-specific node i. In this way, we connect the corrupted tagging information $\widetilde{\mathbf{y}}_{u,i}$ with its corresponding user id u and item id i, and force the middle hidden representation captures the complex interaction patterns among users, items and tags.

4.1.2 Hidden Layer

In the process of encoding the corrupted input $\widetilde{\mathbf{y}}_{u,i}$ into a low-dimensional space, we first obtain the latent user and item representations $\mathbf{p}_u$, $\mathbf{q}_i$ based on the corresponding user id u and item id i by using the embedding table lookup operation. Specifically,

$$\mathbf{P}_u = \mathbf{P}.onehot(u), \qquad \mathbf{Q}_i = \mathbf{Q}.onehot(i), \tag{8}$$

where $\mathbf{P} \in \mathbb{R}^{|U| \times K}$ and $\mathbf{Q} \in \mathbb{R}^{|I| \times K}$ denote the latent user feature matrix and the latent item feature matrix, respectively. And K ($K << |T|$) is the dimension of the hidden layer, which is much smaller than the size of the corrupted input $\widetilde{\mathbf{y}}_{u,i}$. The *onehot*() method indicates the result of one-hot encoding for the user id u or the item id i.

Then, we apply the affine mapping function on the corrupted input $\widetilde{\mathbf{y}}_{u,i}$ followed by a sum operation with other three elements, i.e. the latent user feature vector $\mathbf{p}_u$, the latent item feature vector $\mathbf{q}_i$ and the bias vector $\mathbf{b}$. Finally, we use a non-linear activate function to constrain each element of the above intermediate result, which endows the hidden representation with non-linear model capability. Formally, the encoded result of the corrupted input $\widetilde{\mathbf{y}}_{u,i}$, i.e. the hidden representation of our proposed tag recommendation model, is defined as,

$$\mathbf{z}_{u,i} = g\left(\mathbf{W} \cdot \widetilde{\mathbf{y}}_{u,i} + \mathbf{p}_u + \mathbf{q}_i + \mathbf{b}\right), \tag{9}$$

where $\mathbf{W} \in \mathbb{R}^{K \times |T|}$ indicates the weight matrix between the tag nodes with the nodes in the hidden layer. $g(.)$ is an element-wise activate function, such as the sigmoid function $g(x) = \sigma(x) = 1/(1 + \exp^{-x})$. Note that, unlike the basic DAE model that only extracts hidden representation based on the corrupted input, we integrate the latent user and item feature vectors in the process of encoding the corrupted input, which makes the mapped hidden representation $\mathbf{z}_{u,i}$ encode both the user and item information related the input tagging information $\mathbf{y}_{u,i}$. In other words, this design scheme forces the proposed model to capture the complex interaction behavior patterns among users, items and tags, which are beneficial for boosting the performance of personalized tag recommendation.

4.1.3 Output Layer

In the process of decoding the low-dimensional hidden representation, we recover the input by applying a decoder network to map the hidden representation $\mathbf{z}_{u,i}$ into the high-dimensional input space, whose dimension is equivalent to the size of the input $\mathbf{y}_{u,i}$, i.e. $|T|$. The result of the decoding process is $\widehat{\mathbf{y}}_{u,i}$, which is a reconstructed tagging information encoding the user u' s preference to all the tags T for the item i. Formally,

$$\begin{aligned} \widehat{\mathbf{y}}_{u,i} &= f\left(\mathbf{W}' \cdot \mathbf{z}_{u,i} + \mathbf{b}'\right) \\ &= f\left(\mathbf{W}' \cdot g\left(\mathbf{W} \cdot \widetilde{\mathbf{y}}_{u,i} + \mathbf{p}_u + \mathbf{q}_i + \mathbf{b}\right) + \mathbf{b}'\right) \end{aligned} \tag{10}$$

where $\mathbf{W}' \in \mathbb{R}^{|T| \times K}$ and $\mathbf{b}' \in \mathbb{R}^{|T|}$ denote the weight matrix and the bias vector in the decoder network. $f(.)$ is an activation function in the output layer. As shown in Eq. (10), we recover the input tagging information with the corrupted input $\widetilde{\mathbf{y}}_{u,i}$, instead of the original input $\mathbf{y}_{u,i}$, which forces the proposed model extract more useful and robust features for personalized tag recommendation. In addition, the output $\widehat{\mathbf{y}}_{u,i}$ is a fine-grained preference vector, whose element $\widehat{y}_{u,i,t}$ denotes the possibility of the user u annotating the item i with the tag i.

4.2 Model Learning

In the general auto-encoder framework, there are several forms of parameterized mapping functions that are able to be used for the encoder and decoder networks, such as the encoder that combines affine mapping with sigmoid function, and the decoder that consists of affine mapping and ReLU function. In the proposed DAE-based personalized tag recommendation model, both the encoder and decoder networks are affine mapping followed by the sigmoid function, i.e. both the $f(.)$ and $g(.)$ are the sigmoid functions. One reason is that the reconstruction values that denote the probabilities of users annotate items with some tags should be limited within the range of $[0, 1]$. With the non-linear sigmoid function, the proposed tag recommendation model is able to effectively model the nonlinear interaction behaviors among users, items and tags, i.e. endowing the proposed model with the nonlinear modeling ability. In addition, as reported in [44], the auto-encoder framework trained with the cross-entropy loss matches better with the encoder and decoder networks that consist of affine mapping and sigmoid function, resulting in the extraction of useful features from the corrupted input. Hence, we train the parameters of the proposed tag recommendation by minimizing the cross-entropy loss. Formally, the objective function of the proposed DAE-based personalized tag recommendation model is formalized as:

$$\begin{aligned}\ell_{DAE-PTR} = &- \sum_{\mathbf{y}_{u,i} \in \Omega} \sum_{t=1}^{|T|} (y_{u,i,t} log(\widehat{y}_{u,i,t}) + (1 - y_{u,i,t}) log(1 - \widehat{y}_{u,i,t})) \\ &+ \mathcal{R}(\mathbf{W}, \mathbf{W}', \mathbf{b}, \mathbf{b}', \mathbf{P}, \mathbf{Q}),\end{aligned} \tag{11}$$

where Ω indicates the set of training instances. And $\mathcal{R}(\mathbf{W}, \mathbf{W}', \mathbf{b}, \mathbf{b}', \mathbf{P}, \mathbf{Q})$ is the regularization term, which is used to prevent overfitting, formalized as follows:

$$\begin{aligned}\mathcal{R}(\mathbf{W}, \mathbf{W}', \mathbf{b}, \mathbf{b}', \mathbf{P}, \mathbf{Q}) = &\frac{\lambda}{2}(\| \mathbf{W} \|_F^2 + \| \mathbf{W}' \|_F^2 + \| \mathbf{b} \|_F^2 \\ &+ \| \mathbf{b}' \|_F^2 + \| \mathbf{P} \|_F^2 + \| \mathbf{Q} \|_F^2),\end{aligned} \tag{12}$$

where λ indicates the regularization coefficient. $\| \,.\, \|_F^2$ is the Frobenius norm.

We implement the proposed tag recommendation model based on the Tensorflow computing framework. Specifically, to train model parameters, we first shuffle the training instances, and then we continually feed a mini-batch of instances into the proposed denoising auto-encoder based personalized tag recommendation model until the proposed model converges. In addition, we employ the Adam optimizer [21] to update model parameter since the Adam optimizer tunes the learning rate based on the adaptive schemes and thus yields fast convergence.

5 Experiments

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

5.1 Dataset

In our experiments, we choose two public available datasets, i.e. Last.fm and ML10M [1], to evaluate the performance of all compared models. The Last.fm dataset is collected from the Last.fm online music system, and contains social network, tagging and music listening information from the set of around 2K users. Each user has a list of most listened artists, tag assignments, i.e. tuples (user, artist, tag), and friend relations within the social network. The ML10M dataset is an extension of Movielens10M dataset, published by GroupLeans research group. This dataset contains rating and tagging information assigned to

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

movies by users. Similar to [35, 32], we preprocess each dataset to obtain their corresponding p-core, which is the largest subset where 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 1.

Table 1 Statistics of Datasets

Dataset	#Users	#Items	#Tags	#Tag Assignments
Lastfm	1892	12523	9749	186479
ML10M	4009	7601	16529	95580

5.2 Evaluation Metrics

To evaluate the recommendation performance of all compared models, we adopt the leave-one-out evaluation protocol, which has been widely used in the literature [16, 33]. Specifically, for each post (u, i), we select the last triple (u, i, t) according to the tagging time and remove it from S to S_{test}. The remaining observed user-item-tag triples are the training set $S_{train} = S - S_{test}$. Similar to the classic item recommendation problem, the personalized tag recommendation provides a top-N highest ranked list of tags for a post (u, i). Hence, we employ two widely used ranking metrics to measure the tag recommendation performance of all the models, i.e., $Hit@N$ (Hit Ratio) and $NDCG@N$ (Normalized Discounted Cumulative Gain), where N denotes the length of ranked tag recommendation list. For both metrics, we set $N = \{1, 3, 5, 10\}$ to evaluate the performance in our experiments.

5.3 Experiment Settings

We choose the following traditional personalized tag recommendation models as baselines:

- PITF: PITF [35] explicitly models the pairwise interactions among users, items and tags, and is a strong competitor in the field of personalized tag recommendation.
- NLTF: NLTF [14] is a non-linear tensor factorization model, which enhances PITF by exploiting the Gaussian radial basis function to capture the nonlinear interaction relations among users, items and tags.
- ABNT: ABNT [53] utilizes the multi-layer perceptron to model the nonlinearities of the interactions among users, items and tags.

To make a fair comparison, we set the parameters of each model based on respective references or our experiments, such that the recommendation performance of the compared models is optimal under these parameter settings. For all compared models, the dimension of latent factor vector K is tuned amongst $\{8, 16, 32, 64, 128, 256, 512, 1024, 2048\}$. The mini-batch size is selected from $\{512, 1024, 2048\}$ and the learning rate is tuned amongst $\{0.0001, 0.0005, 0.001, 0.005, 0.01\}$. The regularization coefficient is chosen from $\{0.001, 0.005, 0.01, 0.05\}$. For most datasets and baselines, we empirically set the dimension of latent feature vector $K = 64$, the learning rate $\eta = 0.01$, the regularization coefficient $\lambda = 0.01$, and the training batch size is 1024. In addition, for the ABNT model, the number of hidden layers is set to 2. For our proposed model, the corruption level c is tuned amongst $\{0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$.

5.4 Performance Comparison

Tables 2-5 present the tag recommendation quality of all compared models on the selected four datasets.

Table 2 Recommendation Quality Comparisons on Lastfm-core5

Model	PITF	NLTF	ABNT	DAE-PTR
Hit@1	0.29047	0.22463	0.21697	0.43859
Hit@3	0.52438	0.44148	0.41574	0.67259
Hit@5	0.64211	0.55883	0.51221	0.76481
Hit@10	0.78172	0.71615	0.62180	0.86153
NDCG@1	0.29047	0.22463	0.21697	0.43859
NDCG@3	0.42603	0.34982	0.33227	0.57561
NDCG@5	0.47450	0.39811	0.37202	0.61370
NDCG@10	0.51984	0.44914	0.40773	0.64519

Table 3 Recommendation Quality Comparisons on Lastfm-core10

Model	PITF	NLTF	ABNT	DAE-PTR
Hit@1	0.40150	0.27471	0.24563	0.46717
Hit@3	0.66351	0.52172	0.45966	0.70947
Hit@5	0.77137	0.63990	0.55305	0.79857
Hit@10	0.88501	0.78490	0.66621	0.88934
NDCG@1	0.40150	0.27471	0.24563	0.46717
NDCG@3	0.55380	0.41805	0.36989	0.60891
NDCG@5	0.59833	0.46667	0.40839	0.64569
NDCG@10	0.63529	0.51370	0.44524	0.67528

From Tables 2 to Table 5, we have the following observations: (1) Among all compared models, ABNT achieves the worst performance on four datasets, which indicates that the MLP-based personalized recommendation scheme cannot guarantee good recommendation quality. This observation is similar to the results reported in [34], which illustrate that the MLP is not a better choice for combining embeddings in traditional item recommendation. Al-

Table 4 Recommendation Quality Comparisons on ML10M-core5

Model	PITF	NLTF	ABNT	DAE-PTR
Hit@1	0.31237	0.19736	0.10501	0.35483
Hit@3	0.50425	0.36421	0.21081	0.52840
Hit@5	0.58218	0.44135	0.26849	0.60226
Hit@10	0.67604	0.55317	0.37049	0.68100
NDCG@1	0.31237	0.19736	0.10501	0.35483
NDCG@3	0.42445	0.29422	0.16550	0.45636
NDCG@5	0.45657	0.32601	0.18918	0.48691
NDCG@10	0.48697	0.36221	0.22221	0.51244

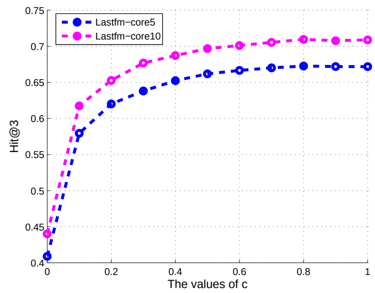
Table 5 Recommendation Quality Comparisons on ML10M-core10

Model	PITF	NLTF	ABNT	DAE-PTR
Hit@1	0.38229	0.30864	0.13688	0.43612
Hit@3	0.59007	0.49631	0.28592	0.62263
Hit@5	0.67226	0.57314	0.35624	0.68876
Hit@10	0.76168	0.67429	0.47142	0.77268
NDCG@1	0.38229	0.30864	0.13688	0.43612
NDCG@3	0.50312	0.41754	0.22231	0.54464
NDCG@5	0.53707	0.44919	0.25119	0.57190
NDCG@10	0.56615	0.48222	0.28848	0.59920

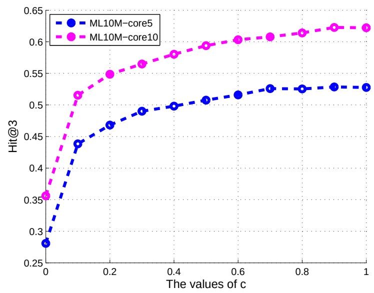
though MLP is a well-known universal function approximator, one possible reason for the poor recommendation quality is that it would require more data and may encounter difficulty to learn the target function since the MLP-based model involves more parameters for representing the target function. (2) PITF is superior to NTLF and ABNT on all datasets and metrics, which demonstrates that the PITF is a strong competitor in the field of personalized tag recommendation. In fact, the dot product is inherently used to model the interactions among users, items and tags in PITF model, while the simple dot product is proved to be a powerful embedding combiner [34]. (3) Our proposed denoising auto-encoder based personalized tag recommendation model exhibits the best recommendation quality over all evaluation metrics and datasets, which demonstrates the superiority of the proposed model over all the baseline models. Compared against the most competitive baseline, i.e. PITF, our proposed model improves the Hit@3 of PITF by 28.2%, 6.9%, 4.8% and 5.5% on Lastfm-core5, Lastfm-core10, ML10M-core5 and ML10M-core10, respectively. In terms of NDCG@5, the improvements of our proposed model over PITF are 29.3%, 7.9%, 6.6% and 6.5% on the above four datasets, respectively. This observation confirms that the adopted denoising auto-encoder framework is able to boost personalized tag recommendation performance. We argue that the improvements is mainly attributed to the adopted denoising auto-encoder framework, whose hidden representations capture the complex relationships among users, items and tags in an implicit manner. (4) each compared model performs better on the core-10 dataset than on the corresponding core-5 dataset. The observation follows up the general principle, i.e., increasing the density of the evaluation dataset is able to improve the recommendation quality.

5.5 Impact of Parameter c

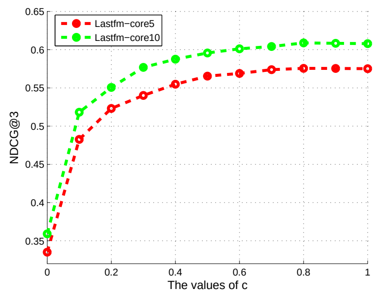
In our proposed personalized tag recommendation model, we force the autoencoder framework to learn robust hidden representations from partially corrupted inputs, which are generated by using the multiplicative mash-out/drop-out noise. In the process of input corruption, the corruption level c is an important parameter, which controls the degree of input corruption. A large value of c indicates that we drop out more elements of the original input $\mathbf{y}_{u,i}$, which makes our proposed model discover more robust and effective hidden representations to capture the complex relationships among users, items and tags. In this section, we perform a group of experiments to investigate the impact of parameter c on the personalized tag recommendation by changing the values of c from 0 to 1. Other parameters are set as follows: the learning rate is 0.001, the regularization coefficient is 0.01. Meanwhile, we set the dimension of latent factor vector K to 512 on the Lastfm dataset and 256 on the ML10M dataset. Fig. 2 presents the impact of c on Hit@3 and NDCG@3 for four datasets. Other evaluation metrics show similar trends.



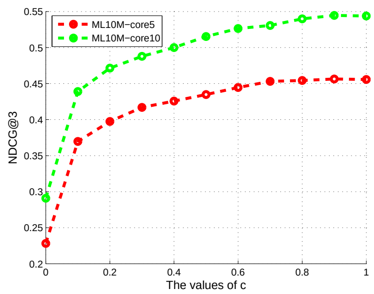
(a) Hit@3 on Lastfm



(b) Hit@3 on ML10M



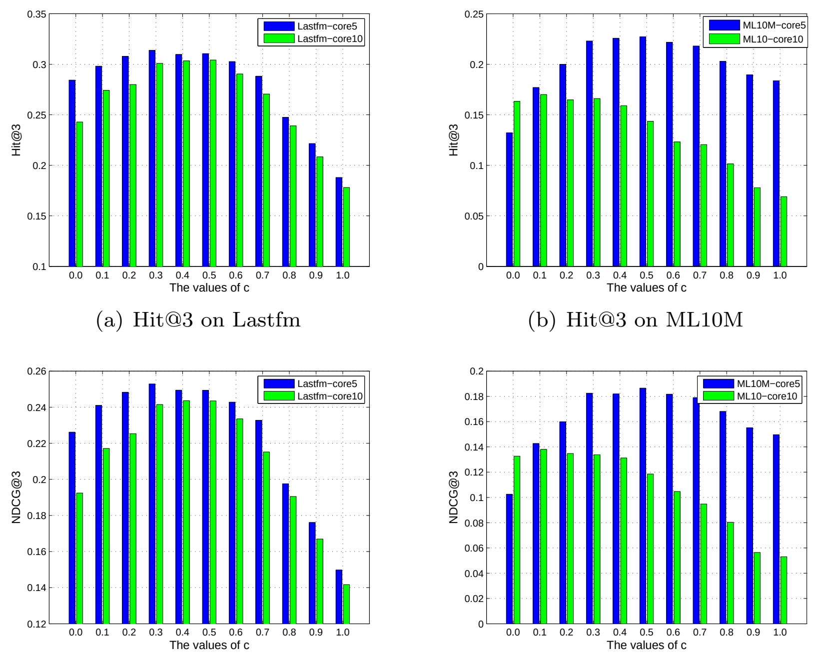
(c) NDCG@3 on Lastfm



(d) NDCG@3 on ML10M

Fig. 2 Impact of c on Lastfm and ML10M

As shown in Fig. 2, the values of c has a significant impact on the personalized tag recommendation quality. In addition, the experimental results indicate that the denoising technique is able to greatly improve the recommendation performance since our proposed personalized tag recommendation model achieves the worst performance when the corruption level is equivalent to 0. Moreover, a large value of c is more beneficial to our proposed recommendation model than that with a small value of c. One possible reason is that adding relatively higher level of noise can force the proposed denoising auto-encoder-based tag recommendation to learn more robust hidden presentations. Furthermore, our proposed personalized tag recommendation model gains the best recommendation quality when c is set as 0.8 and 0.9 on Lastfm, and ML10M, respectively.

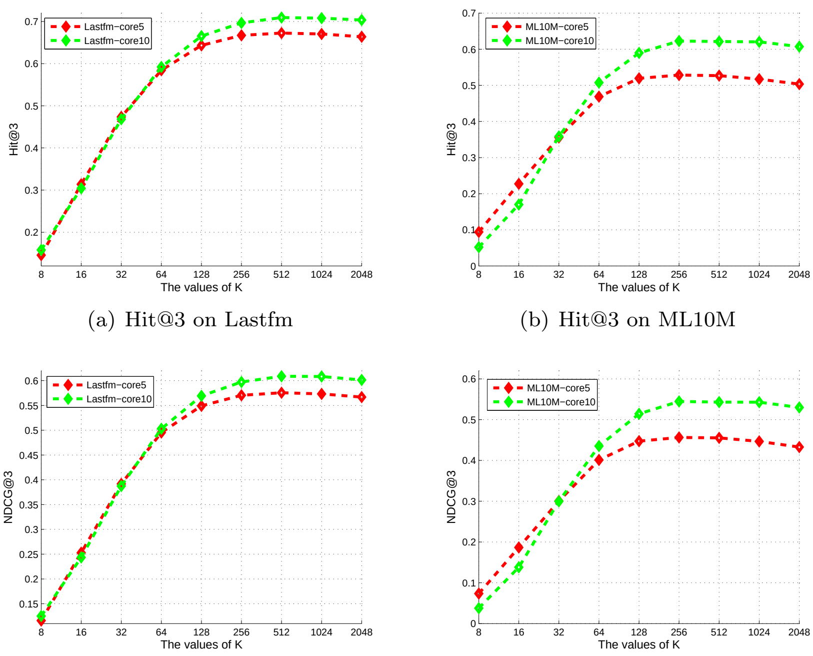


(a) Hit@3 on Lastfm (b) Hit@3 on ML10M

(c) NDCG@3 on Lastfm (d) NDCG@3 on ML10M

Fig. 3 Impact of c on Lastfm and ML10M with $K = 16$

In fact, we also conduct another group of experiments to analyze how the corrupt level c affects the recommendation quality of the proposed personalized recommendation model when the dimension of latent factor vector K is relatively small. We plot the Hit@3 and NDCG@3 of our proposed recommendation model with $K = 16$ in Fig. 3. In this group of experiments, other parameters remain unchanged. As shown in Fig. 3, the proposed model generally performs the best when the corrupt level c is relatively small. For example, with the dimension of latent factor vector $K = 16$, our proposed model achieves the best performance on Lastfm-core5 when $c = 0.3$, while it performs comparatively more efficiently when $c = 0.8$ if $K = 512$. This observation suggests that we should inject small amounts of noise into the denoising auto-encoder-based personalized tag recommendation model when the dimension of latent factor vector K is set to a relatively small value.



(a) Hit@3 on Lastfm (b) Hit@3 on ML10M

(c) NDCG@3 on Lastfm (d) NDCG@3 on ML10M

Fig. 4 Impact of K on Lastfm and ML10M

5.6 Impact of Parameter K

In our proposed personalized tag recommendation model, the dimension of latent factor vector K is another important parameter since it controls the representation capacity of the proposed DAE-based tag recommendation model. In this section, we conduct a group of experiments to study the sensitivity of K to the recommendation quality by changing K within {32,64,128,256,512,1024}. Except for the corrupt level c, other parameters remain the same settings as described in Section 5.3. For each value of K, we tune the value of c ranging from 0.0 to 1.0, and report the best result for each value of K. It is worth noting that our proposed personalized tag recommendation model does not consistently achieve its best performance at the same corruption level for different values of K. The experimental results in terms of Hit@3 and NDCG@3 are plotted in Fig. 4.

As indicated in Fig. 4, the curves of Hit@3 and NDCG@3 show similar changing trends on four datasets. As the dimension of the latent factor vector K increases, the values of Hit@3 or NDCG@3 firstly move upwards, where the recommendation quality improves. After K reaches a certain threshold, both Hit@3 and NDCG@3 keep relatively stable, which indicates that continually increasing the value of K cannot guarantee the continuous improvement of recommendation performance. One possible reason is that the DAE framework used in our proposed model inherently learns the preferences of users or the characteristics of items as well as encodes the complex relationships among users, items and tags via low dimension and compact hidden representations. When the value of K arrives at a specific threshold, the proposed model has enough expression ability, and continuous enlarging the dimension of latent representations may lead to over-fitting. Finally, our proposed DAE-based personalized tag recommendation model achieves its optimal performance when K is equivalent to 512 and 256, respectively.

6 Conclusion

Traditional personalized tag recommendation models mainly focus on learning the latent representations of users, items and tags, but ignore the relationship modeling among them, which leads to sub-optimal tag recommendation quality. In this research, we propose a novel personalized tag recommendation model based on the denoising auto-encoder network, which learns the representations of entities and captures the complex relationships in an unified framework. First, for each user, we add the multiplicative mask-out/drop-out noise into the original input to obtain the corrupted input. Then, we learn the latent representations from the corrupted input via the auto-encoder framework by using the cross-entropy loss. More importantly, we integrate the latent user and item embeddings into the processing of encoding. As a result, the learnt hidden representations of the auto-encoder network are able to encode multiple types of relationships among entities, which are used to boost the performance of the proposed DAE-based personalized tag recommendation model. Finally, we employ the decoder component to reconstruct the original input based on the learnt latent representations. The empirical results on the real-world datasets indicate that our proposed DAE-PTR model is superior to the traditional personalized tag recommendation models.

Recently, some variants of basic auto-encoder network model have been proposed, e.g. stacked denoising auto-encoder [44], contractive auto-encoder [36] and variational auto-encoder [22], and have shown great potential in computer vision and natural language processing. In the future work, we will investigate the incorporation of these variants of the basic auto-encoder network with the proposed personalized tag recommendation model to further enhance its performance.

Acknowledgements The authors would like to acknowledge the support for this work from the Natural Science Foundation of the Higher Education Institutions of Jiangsu Province (17KJB520028) and Tongda College of Nanjing University of Posts and Telecommunications (XK203XZ21001).

Conflict of interest

The authors declare that they have no conflict of interest.

References

1. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. Knowledge and Data Engineering, IEEE Transactions on **17**(6), 734–749 (2005)
2. Askari, B., Szlichta, J., Salehi-Abari, A.: Variational autoencoders for top-k recommendation with implicit feedback. In: SIGIR, pp. 2061–2065 (2021)
3. Bengio, Y., Courville, A., Vincent, P.: Representation learning: A review and new perspectives. IEEE Transactions on Pattern Analysis and Machine Intelligence **35**(8), 1798–1828 (2013)
4. Bengio, Y., Lamblin, P., Popovici, D., Larochelle, H.: Greedy layer-wise training of deep networks. In: NIPS, pp. 153–160 (2006)
5. Breese, J.S., Heckerman, D., Kadie, C.: Empirical analysis of predictive algorithms for collaborative filtering. In: UAI, pp. 43–52 (1998)
6. Cai, Y., Zhang, M., Luo, D., Ding, C., Chakravarthy, S.: Low-order tensor decompositions for social tagging recommendation. In: WSDM, pp. 695–704 (2011)
7. Chandar, A.P.S., Lauly, S., Larochelle, H., Khapra, M.M., Ravindran, B., Raykar, V., Saha, A.: An autoencoder approach to learning bilingual word representations. In: NIPS, pp. 1853–1861 (2014)
8. Chapelle, O., Scholkopf, B., Zien, Eds., A.: Semi-supervised learning (chapelle, o. et al., eds.; 2006) [book reviews]. IEEE Transactions on Neural Networks **20**(3), 542–542 (2009)
9. Chen, M., Xu, Z., Weinberger, K.Q., Sha, F.: Marginalized denoising autoencoders for domain adaptation. In: ICML, pp. 1627–1634 (2012)
10. Chen, X., Yu, Y., Jiang, F., Zhang, L., Gao, R., Gao, H.: Graph neural networks boosted personalized tag recommendation algorithm. In: IJCNN, pp. 1–8 (2020)
11. Dai, T., Zhu, L., Wang, Y., Carley, K.M.: Attentive stacked denoising autoencoder with bi-lstm for personalized context-aware citation recommendation. IEEE/ACM Transactions on Audio, Speech and Language Processing **28**, 553–568 (2020)
12. De Lathauwer, L., De Moor, B., Vandewalle, J.: A multilinear singular value decomposition. SIAM journal on Matrix Analysis and Applications **21**(4), 1253–1278 (2000)
13. Elhamifar, E., Sapiro, G., Yang, A., Sasrty, S.S.: A convex optimization framework for active learning. In: 2013 IEEE International Conference on Computer Vision, pp. 209–216 (2013)
14. Fang, X., Pan, R., Cao, G., He, X., Dai, W.: Personalized tag recommendation through nonlinear tensor factorization using gaussian kernel. In: AAAI, pp. 439–445 (2015)
15. Guan, Z., Bu, J., Mei, Q., Chen, C., Wang, C.: Personalized tag recommendation using graph-based ranking on multi-type interrelated objects. In: SIGIR, pp. 540–547 (2009)
16. He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T.S.: Neural collaborative filtering. In: WWW, pp. 173–182 (2017)

17. Hong, C., Yu, J., Wan, J., Tao, D., Wang, M.: Multimodal deep autoencoder for human pose recovery. IEEE Transactions on Image Processing **24**(12), 5659–5670 (2015)
18. Hotho, A., Jäschke, R., Schmitz, C., Stumme, G.: Information retrieval in folksonomies: Search and ranking. In: European semantic web conference, pp. 411–426 (2006)
19. Hu, Y., Koren, Y., Volinsky, C.: Collaborative filtering for implicit feedback datasets. In: ICDM, pp. 263–272 (2008)
20. Jäschke, R., Marinho, L., Hotho, A., Schmidt-Thieme, L., Stumme, G.: Tag recommendations in folksonomies. In: European Conference on Principles of Data Mining and Knowledge Discovery, pp. 506–514 (2007)
21. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: ICLR, pp. 1–15 (2014)
22. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: ICLR (2014)
23. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. Computer **42**(8), 30–37 (2009)
24. Krestel, R., Fankhauser, P., Nejdl, W.: Latent dirichlet allocation for tag recommendation. In: RecSys, pp. 61–68 (2009)
25. Li, S., Kawale, J., Fu, Y.: Deep collaborative filtering via marginalized denoising autoencoder. In: CIKM, pp. 811–820 (2015)
26. Linden, G., Smith, B., York, J.: Amazon.com recommendations: item-to-item collaborative filtering. Internet Computing, IEEE **7**(1), 76–80 (2003)
27. Mike, H., Jan N., v.R., Aske, P.: A survey of deep meta-learning. Artificial Intelligence Review **54**, 4483–4541 (2021)
28. Nguyen, H.T., Wistuba, M., Grabocka, J., Drumond, L.R., Schmidt-Thieme, L.: Personalized deep learning for tag recommendation. In: PAKDD, pp. 186–197 (2017)
29. Nguyen, H.T., Wistuba, M., Schmidt-Thieme, L.: Personalized tag recommendation for images using deep transfer learning. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pp. 705–720 (2017)
30. Pan, R., Zhou, Y., Cao, B., Liu, N.N., Lukose, R., Scholz, M., Yang, Q.: One-class collaborative filtering. In: ICDM, pp. 502–511 (2008)
31. Quintanilla, E., Rawat, Y., Sakryukin, A., Shah, M., Kankanhalli, M.: Adversarial learning for personalized tag recommendation. IEEE Transactions on Multimedia **23**, 1083–1094 (2021)
32. Rendle, S., Balby Marinho, L., Nanopoulos, A., Schmidt-Thieme, L.: Learning optimal ranking with tensor factorization for tag recommendation. In: SIGKDD, pp. 727–736 (2009)
33. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: Bpr: Bayesian personalized ranking from implicit feedback. In: UAI, pp. 452–461 (2009)
34. Rendle, S., Krichene, W., Zhang, L., Anderson, J.: Neural collaborative filtering vs. matrix factorization revisited. In: RecSys, pp. 240–248 (2020)
35. Rendle, S., Schmidt-Thieme, L.: Pairwise interaction tensor factorization for personalized tag recommendation. In: WSDM, pp. 81–90 (2010)
36. Rifai, S., Vincent, P., Muller, X., Glorot, X., Bengio, Y.: Contractive auto-encoders: Explicit invariance during feature extraction. In: ICML, pp. 833–840 (2011)
37. Salakhutdinov, R., Mnih, A.: Probabilistic matrix factorization. In: NIPS, pp. 1257–1264 (2007)
38. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Item-based collaborative filtering recommendation algorithms. In: WWW, pp. 285–295 (2001)
39. Sedhain, S., Menon, A.K., Sanner, S., Xie, L.: Autorec: Autoencoders meet collaborative filtering. In: WWW, pp. 111–112 (2015)
40. Sun, B., Zhu, Y., Xiao, Y., Xiao, R., Wei, Y.: Automatic question tagging with deep neural networks. IEEE Transactions on Learning Technologies **12**(1), 29–43 (2018)
41. Symeonidis, P., Nanopoulos, A., Manolopoulos, Y.: Tag recommendations based on tensor dimensionality reduction. In: RecSys, pp. 43–50 (2008)
42. Tang, S., Yao, Y., Zhang, S., Xu, F., Gu, T., Tong, H., Yan, X., Lu, J.: An integral tag recommendation model for textual content pp. 5109–5116 (2019)
43. Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P.A.: Extracting and composing robust features with denoising autoencoders. In: ICML, pp. 1096–1103 (2008)

44. Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P.A.: Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. JMLR **11**(11), 3371–3408 (2010)
45. Wang, H., Shi, X., Yeung, D.Y.: Relational stacked denoising autoencoder for tag recommendation. In: AAAI, pp. 3052–3058 (2015)
46. Wang, H., Wang, N., Yeung, D.Y.: Collaborative deep learning for recommender systems. In: KDD, pp. 1235–1244 (2015)
47. Wang, R., Tao, D.: Non-local auto-encoder with collaborative stabilization for image restoration. IEEE Transactions on Image Processing **25**, 2117–2129 (2016)
48. Wang, Z., Du, B., Guo, Y.: Domain adaptation with neural embedding matching. IEEE Transactions on Neural Networks and Learning Systems **31**(7), 2387–2397 (2020)
49. Wang, Z., Du, B., Tu, W., Zhang, L., Tao, D.: Incorporating distribution matching into uncertainty for multiple kernel active learning. IEEE Transactions on Knowledge and Data Engineering **33**(1), 128–142 (2021)
50. Wei, L.C., Deng, Z.H.: A variational autoencoding approach for inducing cross-lingual word embeddings. In: IJCAI, pp. 4165–4171 (2017)
51. Wu, Y., DuBois, C., Zheng, A.X., Ester, M.: Collaborative denoising auto-encoders for top-n recommender systems. In: WSDM, pp. 153–162 (2016)
52. Wu, Y., Yao, Y., Xu, F., Tong, H., Lu, J.: Tag2word: Using tags to generate words for content based tag recommendation. In: CIKM, pp. 2287–2292 (2016)
53. Yuan, J., Jin, Y., Liu, W., Wang, X.: Attention-based neural tag recommendation. In: DASFAA, pp. 350–365 (2019)
54. Zheng, Q., Liu, G., Liu, A., Li, Z., Zheng, K., Zhao, L., Zhou, X.: Implicit relation-aware social recommendation with variational auto-encoder. World Wide Web (2021)