Highlights

A survey for trust-aware recommender systems: A deep learning perspective

Manqing Dong, Feng Yuan, Lina Yao, Xianzhi Wang, Xiwei Xu, Liming Zhu

- A systemic survey of current deep learning-based trust-aware recommendation methods from three aspects: social-awareness, robustness, and explainability.

- A comprehensive overview and comment of the key research issues, and state-of-the-art techniques.

- A summary and discussion of promising future directions to addressing remaining challenges.

A survey for trust-aware recommender systems: A deep learning perspective

Manqing Dong[a], Feng Yuan[a], Lina Yao[a], Xianzhi Wang[b], Xiwei Xu[c], Liming Zhu[c]

[a]*School of Computer Science and Engineering, University of New South Wales, Sydney, NSW 2052, Australia.*
[b]*School of Computer Science, University of Technology Sydney, Sydney NSW 2007, Australia.*
[c]*Data61, Commonwealth Scientific and Industrial Research Organisation, Australia.*

Abstract

A significant remaining challenge for existing recommender systems is that users may not trust the recommender systems for either lack of explanation or inaccurate recommendation results. Thus, it becomes critical to embrace a trustworthy recommender system. This survey provides a systemic summary of three categories of trust-aware recommender systems: social-aware recommender systems that leverage users' social relationships; robust recommender systems that filter untruthful noises (e.g., spammers and fake information) or enhance attack resistance; explainable recommender systems that provide explanations of recommended items. We focus on the work based on deep learning techniques, an emerging area in the recommendation research.

Keywords: Trust, Recommender System, Deep Learning, Survey

1. Introduction

Users face with a significant challenge of information overload as the Web information continuously grows. Recommender systems provide information,

---

Email addresses: `manqing.dong@student.unsw.edu.au` (Manqing Dong), `feng.yuan@student.unsw.edu.au` (Feng Yuan), `lina.yao@unsw.edu.au` (Lina Yao), `xianzhi.wang@uts.edu.au` (Xianzhi Wang), `xiwei.xu@data61.csiro.au` (Xiwei Xu), `liming.zhu@data61.csiro.au` (Liming Zhu)

products, or services to meet users' personalized tastes and preferences to alleviate the above issue. Given those advantages, recommender systems have been popular in various domains and widely deployed in e-commerce applications [1]. For example, when a user looks for a pair of sleep-buds at Amazon[1], the website may recommend a case for the sleep-buds as "Frequently bought together" and a list of products that other customers frequently "buy after viewing this item." According to a report[2], 35% of Amazon.com's profit owes to its recommendation algorithms.

Despite the success of recommender systems, a significant remaining challenge is that users may not trust the recommender systems for either lack of explanation or inaccurate recommendation results. For example, a user may not trust a stranger's taste even though they have similar history records; moreover, the system may recommend an item that is intentionally highly rated by malicious users. All these make a trustworthy recommender system urgent and important. In this work, we define and focus on three aspects of trust in recommender systems:

Social-awareness. With the development of social media, social-aware (sometimes called trust-aware) recommendation has attracted more attention. Recent studies suggest users' ratings area has a positive correlation with the average of their social neighbors for both trust-alike relationships or trust relationships [2]. On the one hand, based on the phenomenon that users' tastes are often influenced by their friends [3], leveraging the trust relationship has great potential to provide a trustworthy recommender system to the user, and also improve the recommendation quality by predicting the user's taste. On the other hand, adding such information alleviates the cold start problem of traditional recommender systems. Generally, social-aware recommender systems include memory-based methods and model-based methods. Memory-based methods usually generate predictions for a user by leveraging the ratings of his/her direct or indirect trusted friends [4]. Thus, the performance of such methods largely depends on trust propagation models. In addition, these methods are usually time-consuming and therefore are not suitable for handling large scale applications because they need to calculate similarities over an entire rating matrix and the whole trust network [4]. Model-based methods (e.g., matrix factorization and deep learning) are

---

[1]https://www.amazon.com/

[2]http://rejoiner.com/resources/amazon-recommendations-secret-selling-online/

widely used [5]. Compared with memory-based methods, which uses trust ties to infer users' neighbors and then promote the accuracy of similarity calculation among users, most model-based methods simultaneously map users and items into low-dimensional feature spaces and then train a prediction model by optimizing some objective functions overrating and trust data [4].

Robustness. While more people are relying on online product reviews to make purchase decisions [6], reviews and social relations are increasingly subject to attacks from spammers and bots. Moreover, social relations have different interpretations in different contexts. For example, two people may reach a consensus on movies but have diverse opinions on purchasing clothes [7]. There are two directions of research in improving the robustness of recommender systems. The first direction is towards filtering out noisy or malicious feedback from the data before executing the recommendation algorithm. Research in this direction aims to find statistical patterns of users/ratings to identify malicious content [8], use supervised classification methods based on feature-engineering [9], or use unsupervised clustering approach to eliminate suspicious users [10]. The second direction aims to develop noise attack-resistant algorithms [11]. Related research mainly constructs noise-sensitive algorithms or introduce auxiliary information (e.g., human-made noise) into the recommendation [12].

Explainability. Users tend to trust a recommendation when provided the appropriate information to understand the recommendation process and results. Explainable recommender systems [13] not only provide users with personalized recommendations but also generate descriptions on why the items are recommended. Therefore, explainability improves both the trustworthiness and transparency of recommendation results. In general, explainable recommender systems can be classified according to two orthogonal criteria: information source and methodology. Existing work utilizes a variety of contents, such as features of items/users [14], textual reviews [15], product images [16], and social connection [17]. And various approaches have been used to generate explanations, e.g., matrix factorization [18], graph-based models [19], topic models [20], deep learning [21], and association rule mining [22]. In addition, much work exists that combining different information sources and methods. In this survey, we limit our scope to only deep learning models and give a thorough analysis of the recent work.

Deep learning techniques have exploded during the recent years. The 2019 Turing Award also recognizes the significant contribution of deep learning to various machine learning tasks. A recent survey on deep learning-based

recommender systems [1] points out the neural architecture's ability of being end-to-end differentiable and providing suitable inductive biases. One significant contribution of deep learning approaches is about the representation learning. For example, many studies use deep learning to learn compact representations from auxiliary data such as content, tag, images, or social graph relationships, and then use the compact representations for prediction or combined with traditional matrix factorization methods [3, 23].

Until now, considerable work has been conducted on applying deep learning into the trust-aware recommendation, including social-aware recommender systems [24], robust recommender systems [25], and explainable recommendation [13]. However, to the best of our knowledge, there is a lack of systemic survey on current deep learning-based trust-aware recommendation methods. This survey aims to review the trust issue in recommender systems from a deep learning perspective to fill the gap. We outline three aspects of trust, i.e. social-awareness, robustness, and explainability. For each aspect, we present the literature review and summarize the related deep learning-based techniques.

The rest of the survey is organized as following. From section 2 to 4, we introduce three categories of trust-aware recommender systems, respectively. In each section, we first provide an overview of the related methods and then introduce the technical details of deep learning-based approaches by the type of algorithms. Then, in section 5, we summarize several challenges of current trust-aware recommendation techniques and provide insights into this field. The following sections assume readers have the basic understanding of deep learning techniques and concepts of recommendation techniques.

2. Social-aware Recommender Systems

2.1. Overview

Social relations have proven helpful in boosting recommendation performance and thus has attracted much attention these years [26]. Social recommendation techniques make use of the user-user trust social links to complement the sparse rating data and thus improve the user preference prediction by considering not only the user rating behavior but also the preference of the user trusted neighbors. Since users usually are interactive with people around them, social relations can greatly help users filter information and alleviate the cold start problem.

2.1.1. Problem Definition

Suppose $\{u|u \in U\}$ is a list of M users, $\{i|i \in I\}$ is a list of N items, and R is the user-item rating matrix, where $r_{ui} \in R$ is the rating by user u for item i. We denote by t_{uv} $(t \in T)$ the trust relationship between user $u \in U$ and user $v \in V$, where T is the user-user trust relationship matrix. Besides, we denote item description and user feedback by D_u and D_i. Each element in D_u and D_i is a feature vector denoted by d_*. Rating-based recommendation tasks aim to predict a rating $\hat{r}_{ui}$ of user u on an unknown item i,

$$\hat{r} = f(R, T, D_u, D_i) \tag{1}$$

where f links the given information to ratings in a fixed range, e.g., [0,5].

In contrast, rank-based tasks aim to provide a user u top-K items with the highest score s_{ui}, which capture users' preference.

$$\hat{s} = f(R, T, D_u, D_i) \tag{2}$$

2.1.2. Traditional Methods

Traditional methods for social-aware recommendation include memory-based methods and model-based methods. The **memory-based** methods deduce ratings of a targeted user via trust propagation based on ratings of its friends [27]. For example, Jamali and Ester [28] combine TrustWalker [29] with neighborhood collaborative filtering. They first run random walks on the trust network and then perform a probabilistic item selection strategy to generate recommendations. Similarly, Zhang et al. [30] extract reliable social information from user feedback and use top-k identified friends to infer the user preferences. Matrix factorization is probably the most widely used technique for **model-based** social-aware recommendation. Wen et al. [31] learn vector representations of social relations via node2vec [32] and then combine them with rating history to conduct matrix factorization. Zhao et al. [33] present a trust-based Bayesian personalized Ranking approach to incorporate trust friendship. They assume the friends preference will affect the users decisions, i.e. the user will give higher ranks to items that preferred by their friends. Guo et al. [2] use a SVD++[34] based methods with considering the user preference and friend's influence. Ahn et al. [35] provide the theoretical support for considering social relationships in recommender systems.

Deep learning-based social-aware recommendation methods diverge in three types: **regularization methods** minimize the distances of latent fea-

tures between trusted users and maximize the latent features' distances between distrusted users to reflect social proximity; **ensemble methods** generate a new rating from the ratings of both a user and its social network; and **co-factorization methods** assume users should share the same user preference in rating and social space [36].

We summarize various techniques for social-aware recommender systems, including autoencoder, recurrent neural network (RNN), graph neural network (GNN), generative models (GM), and hybrid methods, in Table 1.

Table 1: A Summary for Social-aware Recommender Systems

Method	Regularization	Ensemble	Co-factorization	Others
Autoencoder	[37]	[3],[38],[39],[40]	[37]	[41]
RNN	-	[42, 43]	-	-
GNN	-	[26, 43, 44, 27]	[7]	[45]
GM	-	[46],[47]	-	-
Hybrid Methods	-	[48],[49],[50],[51],[52]	-	[23, 33, 53]
Others	-	[54],[55],[56],[20]	-	[57, 58, 31, 59]

2.2. Autoencoder-based Methods

Autoencoder is a type of artificial neural network for learning compressed representations (encodings) for a set of high-dimensional data [40]. Autoencoders can help recommendation with either learning latent factors of users and items (encoder) or reconstructing user's preferences (decoder). For the former one, the learned latent representations are normally further cooperated with other methods for prediction. For example, Deng et al. [40] use the learned latent factors as the initialization of matrix factorization. For the latter one, the decoder of the autoencoder is inferring the potential ratings from the user rating records through a narrow network, where the bottleneck of the network is representing the latent representation of the user rating records [3]. For the social recommendation problem, where we have both user-item rating matrix and user-user trust networks, we also divide the autoencoder based methods into the two aforementioned categories: use **reconstructed input for recommendation** and use **latent representation for recommendation**.

2.2.1. Using Reconstructed Input for Recommendation

This type of method learns from dense representation and reconstructs input as predictions for recommendation. A key problem for this type of

method in social-aware recommendation problems is the way of learning and aggregating representations from the social and rating information.

One way is to learn an ensemble representation of the two types of information for prediction. For example, Pan et al. [3] balance the contributions of the two representations learnt from social relationships and rating history via a weighted layer (Figure 1 (a)). They then use a correlative regularization to exchange information [3] and the unified latent representation to predict user ratings and trust relationships. Wang et al. [38], instead, directly concatenate the two latent representations for recommendation (Figure 1 (b)).

Another idea is to assume that a user's social representation share the same representation with the user's rating representation. For example, Nisha et al. [37] generate a list of trusted users and minimize the distance between their social representation to learn users's social representations. They first use an autoencoder to encode users' rating patterns and item's history rating patterns and then decode the learned representations for recommendation (marked as yellow in Figure 1 (c)). They also use a regularization item to control the distance between the user's social representation and the user's rating representation when training the autoencoder.

2.2.2. Using Latent Representation for Recommendation

Another idea for utilizing autoencoder for social-aware recommendation is to combine the learned latent representation with other methods. For example, Rafailidis et al. [41] learn user latent representations from the social relationships via deep autoencoders and then use the latent factors in matrix factorization (Figure 1 (d)). Liu et al. [50] use stacked denoising autoencoder (SDAE) [60] to learn the social information. The input of SDAE is from K friends of a given user, and each friend is represented by a vector. By aggregating the information from all friends, a condense vector is used for representing the user, see Figure 1 (e). Such representation vector is further combined with other methods for recommendation. Similarly, *Wu et al.*[39] use autoencoder to extract the compact representation of the social network and predict ratings by aggregating user information and item information via several fully connected neural network layers (Figure 1 (f)).

2.3. RNN-based Methods

Recurrent Neural Network (RNN) has shown its power in dealing with sequential data, e.g., textual [61] and time-series data [62]. RNN-based methods generally target in dynamic user behaviors [63], preferences[64, 65] or the
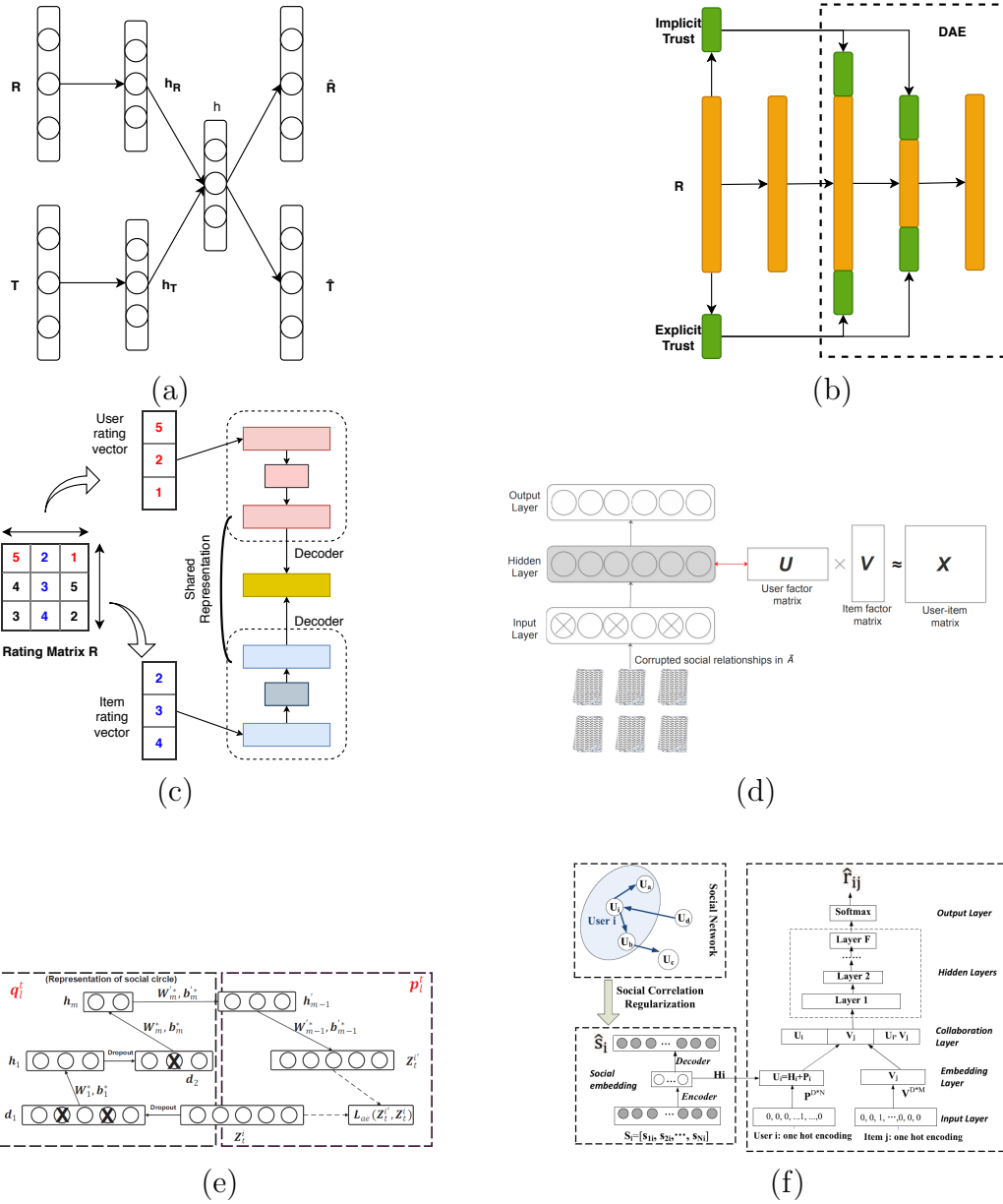
7

Figure 1: Autoencoder for social-aware recommendation. The figures are from (a) Pan et al. [3], (b) Wang et al. [38], (c), Nisha et al. [37], (d) Rafailidis et al. [41], (e) Liu et al. [50], and (f) Wu et al. [39].
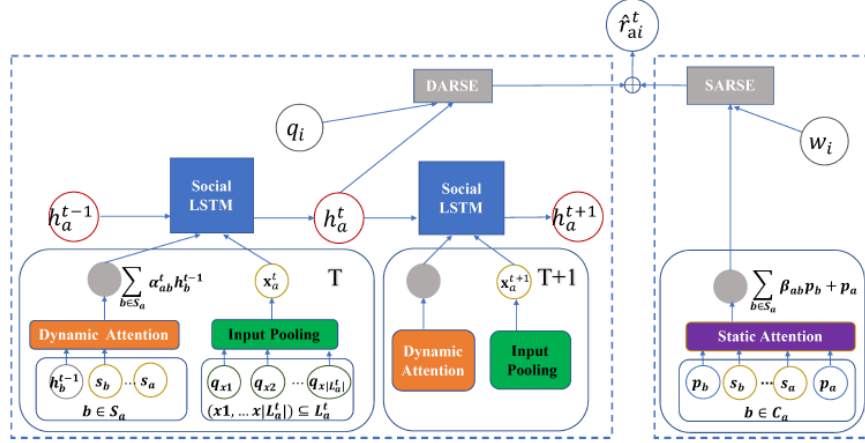
Figure 2: Recurrent neural network for social-aware recommendation. (Sun et al. [42])

side information [66]. Such methods majorly uses RNN to capture the sequential information and then learn a temporal or concrete representation for further uses. For example, Song et al.[43] capture the users' current preferences with RNN based methods and concatenate such information with user's history clicks as the representation for users.

For social-aware recommendation problem, RNN can extract user's temporal preferences and temporal biases of friends. For example, Amy liked painting last year and so as her friend Sarah. She was influenced by Sarah's preferences that she bought a painter with the same brand as Sarah's. This year, Amy starts to learn guitar, then she may infer other friends' preferences who are good at playing the guitar. Sun et al. [42] propose a recurrent network based model with attention for temporal recommendation (see Figure 2). The method includes a static part, which captures the insistent user preference, and a dynamic part that captures the dynamic user preference. For the static part, the static social attention module is applied for selecting the static social relationships for each user, and then aggregates these social relationships together for enriching the user's representation vector. For the dynamic part, a LSTM module is implemented for capturing the complex temporal latent representation of users, i.e. consider the social influences into the temporal preference modeling. Each part will predict a user preference score, and the final rating prediction is the sum of the scores from the two parts.

9

2.4. GNN-based Methods

Graph neural network (GNN) has shown the effectiveness in learning on graphical data by the power of integrating node information and topological structures. As such, for social-aware recommendation problems, GNN has a great potential for mining the social graph structures and user-item graph; where a key is utilizing the GNN to learn the latent factors of users and items [26]. For example, Fan et al. [26] consider to learn the user latent factor and item latent factor via GNNs first, and then concatenate the two latent factors for the final rating prediction (Figure 3 (a)). For the user modeling, the user latent factor is the concatenation of item aggregation and social aggregation. The process of item aggregation is aggregating the user rating histories toward different items (the representation for each item is the combination of item-vector and the item rating) with attention algorithms; and the process of social aggregation is aggregating the user's friends rating histories (for each friend the representation is the reconstruction of user-item vector) with attention algorithms. For the item modeling, the item latent factor is the aggregation of other users' historical ratings toward the target item.

Instead of static modeling of social relationships, the inference by friends may also change along with time. In this regard, Song et al. [43] consider a dynamic situation that users' interests are dynamically influenced by the social relationships – the preferences of the friends may change differently among different periods. They propose a session-based social recommendation algorithm, which models dynamic interests and dynamic social influences. The whole structure is shown in Figure 3 (b). The model captures the user's current preferences by a RNN module, which models the user's historical actions, such as clicks. For modeling the friends' interests, the authors considered both short-term and long-term preferences, where short-term preference is modeled by RNN with capturing the current session's preference and long-term preferences capture the average interests. Then, each friend is represented by a concatenation of the short-term and long-term preferences. To learn social-aware user representation, the authors use an attention algorithm (which is learned by the similarity between the target user and the friends) to leverage the importance of the social relationships and then aggregate them with different weights. Then, they combine the social-aware user representation with the dynamic user interest. The probability distribution of recommending items is the softmax of the similarities between item embeddings and user hybrid representation.
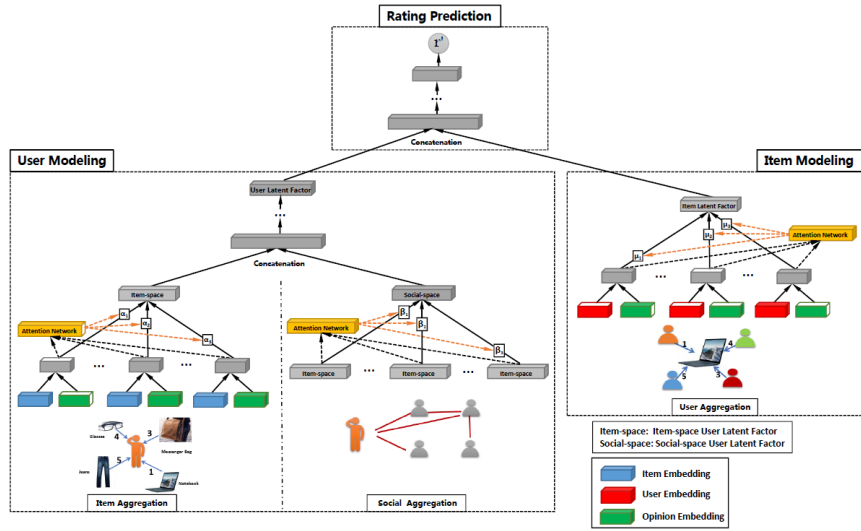
(a)
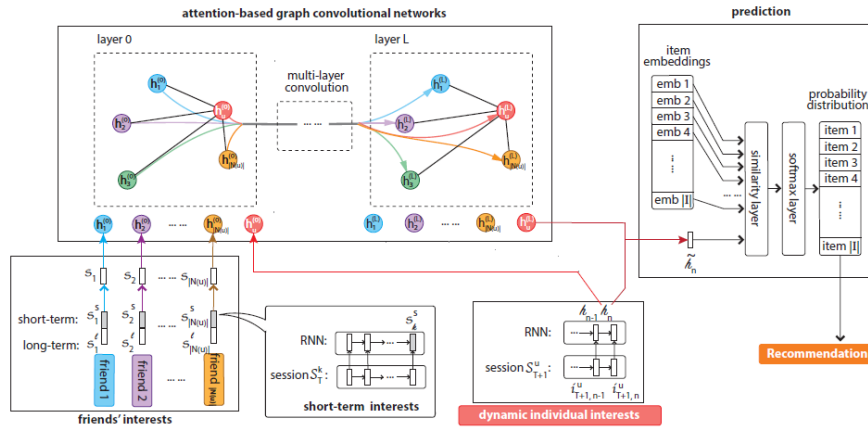
(b)

Figure 3: Graph Neural Network for social-aware recommendation from (a) Fan et al.[26] and (b) Song et al. [43].

As one type of graph neural network, **Graph Convolutional Network (GCN)** has been widely applied in recent social-aware recommendation studies due to the effectiveness in mining social relationships. Different to graph neural network, GCN generates the node embedding in a view of message passing or information diffusion[7], which can encode the graph structure information as low-dimensional representations. Specifically, the embedding for each node is the aggregation of the information from the neighborhoods; the embedding of the neighbors is further learned from the neighbors of the neighbors, and so on. For example, in [45], the authors represent the node in graph with a two-layer graph convolutional neural network (see Figure 4 (a)). In details, the embedding for a node (i.e., an item) is the aggregation of feature information (e.g., visual, textual features) from the node's local graph neighborhood. Each aggregation module learns how to aggregate information from a small graph neighborhood, and by stacking multiple such modules. Then, these embeddings are then used for recommender system candidate generation via nearest neighbor lookup or as features in machine learning systems for ranking the candidates. In [7], the authors assume that the user preference is modeled as the inner product of user and items embeddings. They first initialize the user embedding and item embedding, where each is the combination of descriptive features (such as user profiles and item descriptions) and free basic latent vector. Then, they model the diffusion of the user preferences as in layer-wise diffusion manner. Figure 4 (b) shows the details. Similarly, Wu et al.[27] assume that the representation of a user can be learned from a user-specific latent vector and the user's history rating records, which is called item-based user embedding; and the representation of an item can be learned from an item-specific latent vector and the users who rated it, which is called user-based item embedding (check Figure 4 (c)). For learning the hybrid representation from embeddings with graph structures, the authors tried to use Graph Attention Network (GAT) based approach. Traditional GCN treat each neighbor equally and aggregate the embedding information without balancing them. Thus, the authors implement GAT, which leverages the attention mechanism to balance the significance of the neighbors, enhance the model to focus on important input, and improve the robustness of the model with filtering noise. In details, for an item (a user), the authors first learn its static attribute factor (user static preference factor) from the item embedding (user embedding and user relationships) and then learn a user-context-aware item factor (item-context-aware user factor) as a dynamic attribute factor (dynamic preference factor). And then they fuse

the four factors (the static factor and dynamic factor for user and item) into a synthetic representation by a policy-based fusion layer, and the prediction is made based on it.

2.5. Generative Models

Generative models classically describe models of joint distribution with data and labels. It will be taken to produce new samples with mechanisms sampling from the real data. The two most common types of generative models are generative adversarial nets (GANs)[67] and Variational Autoencoders (VAEs)[68].

2.5.1. GAN-based Methods

Generative Adversarial Network (GAN), which includes a generator and a discriminator to conduct adversarial learning, has shown the effectiveness across various domains due to the ability of learning data probabilistic distribution and generating new samples. The generator attacks the discriminator by generating new samples with similar distribution as real samples; and the discriminator is distinguishing the source of the samples, i.e. whether the sample is coming from real cases or generated cases. Then, a min-max game is played between the two processors, which can promote both of the two models. This is known as adversarial training. When it comes to the recommendation problem, generative models are normally applied for i) predicting missing values and ii) enhancing the representation of items and users. For example, Wang et al.[69] use a generative model to generate simulated user preference distribution of real data. The performance of the generative model is improved by maximizing the classification loss, and the training will be stopped until producing promising generative predicted ratings. For the social-aware recommendation problem, the critical problems are the way of learning information from trust relationships and the way of combining such trust information with rating history. Fan et al. [46] design two adversarial learning modules for enhancing the user representations in the user-item rating part and the social part. In details, for each part, a discriminator is designed for distinguishing the real instances and the generated samples, and a generator is designed for modeling the actual conditional distribution for a given user. For adaptively enhancing the representations in two parts, they utilize a bidirectional mapping between the two parts, where in each iteration, the user's social representation will be updated by a nonlinear mapping
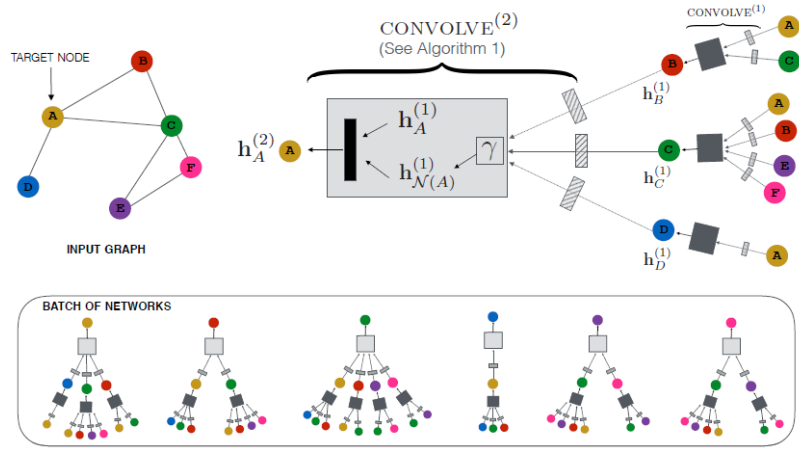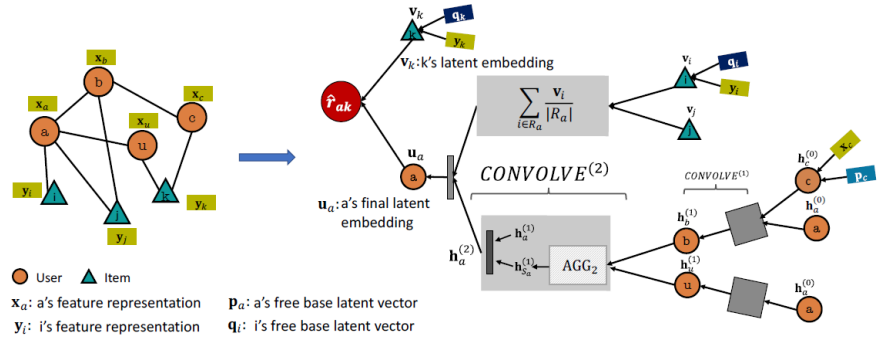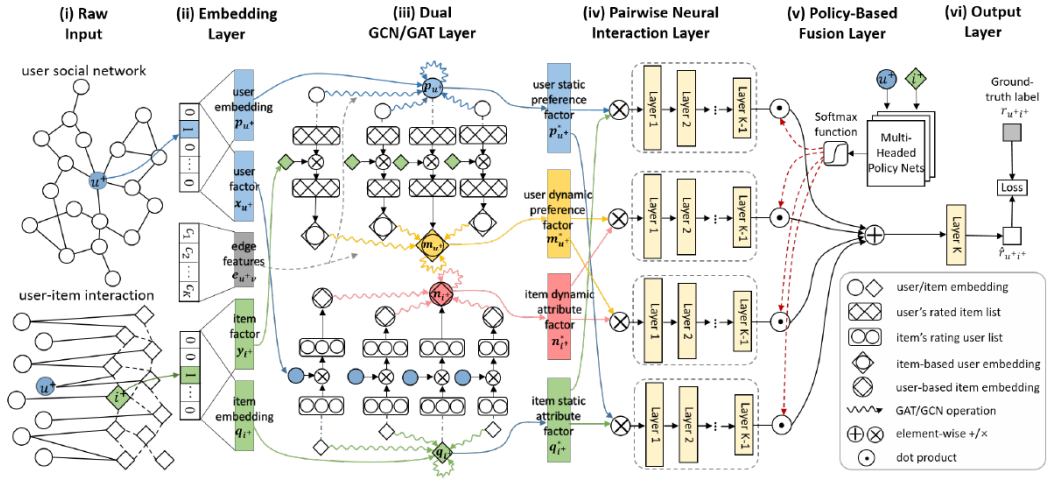
Figure 4: Graph Convolutional Network for social-aware recommendation from (a) Ying et al. [45], (b) Wu et al. [7], and (c) Wu et al. [27].

operation from rating pattern representation, and then updated by the domain adversarial trainer; likewise, the user's rating pattern representation will be updated with the trained social representation, and then be following with the domain-specific training, see Figure 5 (a) for details.

2.5.2. VAE-based Methods

Similar to autoencoder-based methods, variational autoencodesr can be used for predicting missing values or learning comprehensive representations. For the former task, the VAE predicts item ratings/scores by inferring the latent factors [70]. It consists of three parts: a bag-of-items vector i_u by user u is provided as input to the decoder; a latent user vector z_u is sampled from a Gaussian distribution with parameters specified by the encoder; and a new bag-of-items vector $\tilde{i}_u$ see 5 (b). For the latter, the VAE learns representation for users or items. For example, Xiao et al.[47] consider three types of information for recommendation: user-trust relationship, user-item rating history, and item content information. For representing the users, each user is represented by the aggregation of trusted users. For representing the items, the content information for items is considered, where a variational autoencoder model is used to learn the latent patterns for content information. Such latent patterns are used to represent the items. Then, they consider using traditional matrix factorization method for predicting the user-item ratings. The graph structure can be noted in figure 5 (c).

2.6. Hybrid Methods

For bridging the advantages of the above models for enhancing the recommendation performance, several hybrid models are proposed. The hybrid models may adopt hybrid algorithms or multiple types of input for recommendation.

According to the research targets and characteristics of features, multiple algorithms may apply. For example, RNN based methods are widely applied for dealing with session-based recommendation problem, and autoencoder is good at extracting a condense representation of input. Then for solving a session-based trust-aware recommendation problem, such two methods can be applied together. Liu et al. [50] consider that user's preferences are changing over time. They use a stacked denoising autoencoder to learn the user representation, which is the aggregation of the friends' representations, at each time step. Then, such representation is used as the input of the
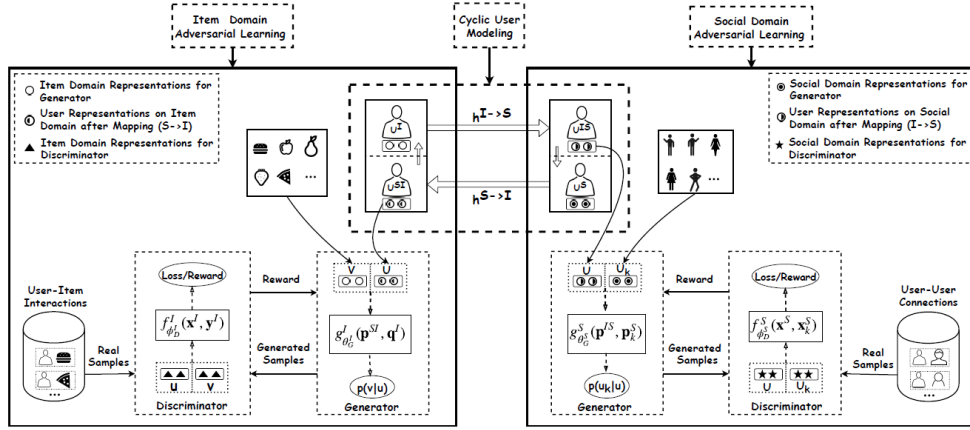
(a)
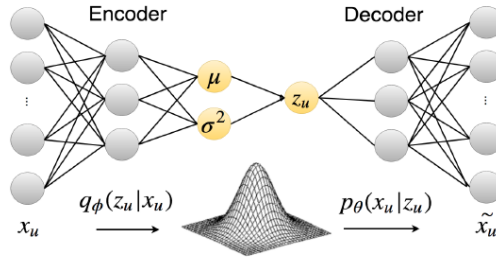
(b)
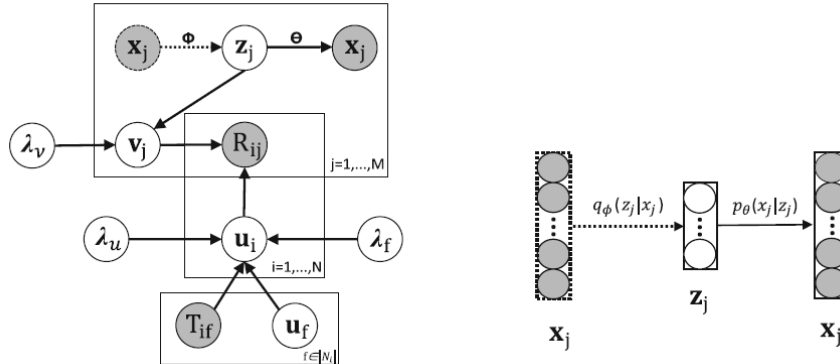
(c)

Figure 5: Generative models for social-aware recommendation from (a) Fan et al.[46], (b) Karamanolakis et al. [70], and (c) Xiao et al. [47]

16

LSTM module for predicting user's current preference. A stacked LSTM for predicting the user's whole time preferences. See Figure 6 (a).

For recommendation problems such as image recommendation and movie recommendation, the auxiliary information will be considered. Wu et al. [48] target an image recommendation problem. For the image information, they use CNNs for learning the embedded representation. Besides for the images, they consider three auxiliary information: user upload history, social influence, and creator admiration. They use an attention model to leverage such three aspects. The user's preference is represented by the aggregation of all the information. And the rating prediction is based on the product of item embedding and user's preference vector, see Figure 6 (b). Zhao et al. [33] design a heterogeneous social-aware movie recommender system by exploiting multi-modal movie contents (i.e. images and corresponding descriptions), users' social relations and their relative preference feedback. The goal of the model is providing top-K movies for recommendation, where the rank for movies is a ranking score of the given user and item, which is calculated on the representations for the user and item. For learning a sharing item representation with both movie images and descriptions, they use a multi-modal learning approach: a deep convolutional neural network for the images and a deep recurrent neural network for the descriptions. The representations for users, which is the aggregation of friend relationships, are learned via DeepWalk[71]. Monti et al. [23] propose two different ways of predicting the user-item rating matrix. The first one is Recurrent Multi-Graph CNN (RMGCNN) architecture, see the top figure of Figure6 (c), which operates on the user-item matrix and operates simultaneously on the rows and columns. Both of the users and items are learned via Multi-Graph CNNs: the users are modeled by their relationships, and the items are modeled with the images. Then, the whole rating matrix X is learned by RNN model step by step, until provides stable predictions. For the second method, which is called Separable Recurrent MGCNN (sRMGCNN), operates separately on the rows and columns of the matrix, see the bottom figure of Figure6 (c). Gao et al. [51] consider a video recommendation problem. They propose a dynamic RNN to capture the user dynamic preference by considering the video information, user interest, and user social relationships. The video semantic embedding includes visual features and textual features, which are learned by pre-trained deep models. User interest modeling is based on the user view history, which is learned by topic modeling. As for the user social relationship mining, see Figure 6 (d).
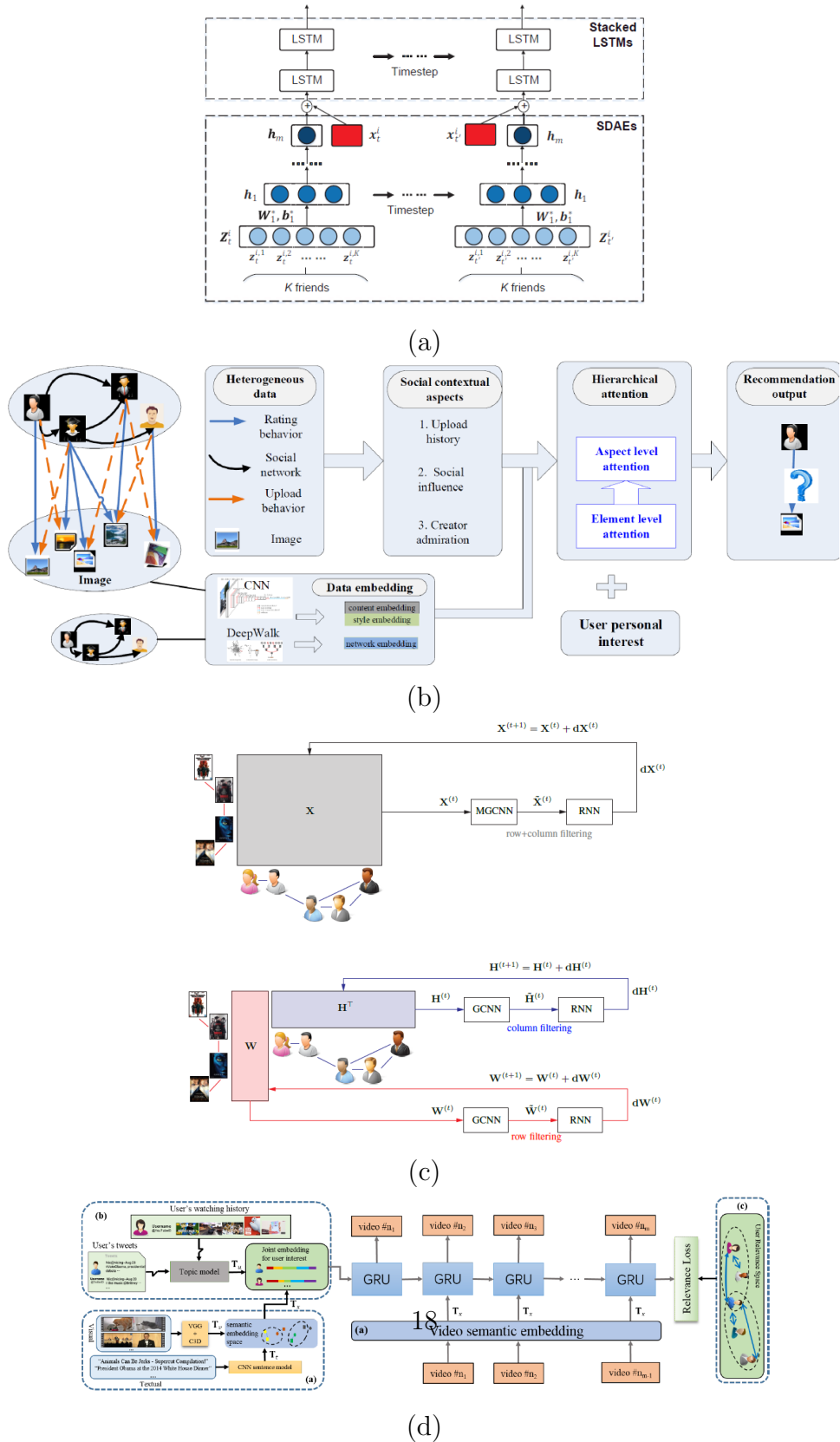
(a)

(b)

(c)

18

(d)

Figure 6: Hybrid methods for social-aware recommendation from (a) Liu et al. [50], (b) Wu et al. [48], (c) Monti et al. [23], and (d) Gao et al. [51].

2.7. Others

2.7.1. Attention-based Methods

Informally, a neural attention mechanism enables the neural network focus on a subset of its input (or features), i.e. assigns different weights to the input. For example, for the machine translation problem, it allows the machine translator to look over all the information the original sentence holds, then generate the proper word according to current word it works on and the context[72]. Now the attention mechanism is popular in many other areas, such as object recognition and image caption[73]. There is limited work only emphasises on discussing the effectiveness of applying an attention mechanism into recommender systems; but on the contrast, as an enhancing module, the attention can work well when incorporated with other models for recommendation[1].

For the social-aware recommender system, the attention mechanism could be used for balancing the friends influences. For example, *Chen et al.* [54] consider the problem that the influence of user's friends should be different and dynamic. For different items, the user may infer different friends preferences. Thus, they propose a hierarchical attention module for the recommendation. First, each friend's representation is built on the user embedding and friend embedding, and is learned by attention mechanism. Then, for different friends, the authors also applied an attention mechanism for balancing friends' influences to get a final user representation. This user representation is then multiplied with item representation as a score for ranking. *Rafailidis and Weiss* [55] propose a similar structure that considers a subset of friends and uses an attention mechanism for social collaborative filtering.

2.7.2. Combining with Matrix Factorization

Traditional matrix factorization methods is predicting the ratings/scores by multiplying the latent representations of users h_u and items h_i, i.e. $\hat{r}_{ui} = h_u^\top \cdot h_i$. The difference between deep matrix factorization methods and traditional ones is that the latent representation learning is implemented with deep learning techniques. For tackling social-aware recommendation problem, a common way is combining the social influences into the user representation. Fan et al. [57] learn the social embeddings via node2vec [32], and then use multilayer perceptron for learning the embeddings. Each user is represented by the social embeddings and an initialized latent factors. Such user representation is further be used in probabilistic matrix factorization [74]. Bao et al. [49] use an attentive way for learning social influences. They first use

19

autoencoder to learn compact representation of neighbors, where each neighbor is represented by h_v. Then the social influences for user u is modeled by $h_T = \Sigma_{v \in V} \alpha_v \cdot h_v$; α is the attention value. Then the user latent representation is given by $\hat{r}_{ui} = (\beta h_u + (1 - \beta)h_T)^\top \cdot h_i$, where β is a self-defined hyper-parameter. In [58], the authors argue that the matrix factorization method could be represented with a shallow neural network model. They consider a cross-domain recommendation problem[75], where they are trying to recommend a top-K items list from information-domain to users in social-domain. Specifically, they use a deep collaborative filtering model to predict the user preference $\hat{s}_{ui}$, which is calculated by the latent user representation h_u and latent item representation h_i. Both of the latent representations are learned from the initial embedding vector and attribute vectors (i.e. the hashtags). Then, the prediction of user-item interaction is further enhanced by integrating the social relationships: the intuition that users with strong connections are more likely to share similar tastes on items. The processing is minimizing the user latent representation gap between strongly connected users.

2.7.3. Others

Xiao et al. [56] propose a SVD++[34] based model for recommendation. The inputs for the network are user's representation h_u, item's representation h_i and user's social representation h_t. They incorporate the social relationships by considering both its latent representation and the interaction with items. The following layer is represented as the concatenation of the above information, i.e. $[h_u, h_i, h_t, f(h_u, h_i), f(h_i, h_t)]$, where f is representing several neural networks. The prediction is made after a few fully connected layers.

2.8. Summary

To summarize, deep learning based social-aware recommendation algorithms have shown their effectiveness in different tasks. Different to traditional methods, deep learning based techniques need less manual extracted features and have the advances in grasping complex latent feature interactions. It is a trend to incorporate traditional recommendation methods with deep learning methods, e.g., graph neural network, which can leverage both advantages. Although effective, current deep learning based social-aware recommendation algorithms have the following challenges.

Table 2: Commonly used datasets for social-aware recommendation

Dataset	Ciao	Epinions	FilmTrust	Flixster	Douban
# of Users	7,317	18,088	1508	147,612	129,490
# of Items	104,975	261,649	2071	48,794	58,541
# of Ratings	283,319	764,352	35,497	8,196,077	16,830,839
Density (ratings)	0.0368%	0.0161%	1.1366%	0.1138%	0.2220%
# of Social relations	111,781	355,813	1,853	7,058,819	1,711,780
Density (social relations)	0.2087%	0.1087%	0.0815%	0.0324%	0.0102%

- The quality and the quantity of the social links. For example, in most of the recommender systems, it is hard to get explicit and reliable links since a few amount of users indicate their social relationships.

- Most of the current works model the trust relationships with shallow model and ignore the high-order interactions among each users' friends. It's possible for a user to take all the opinions of his friends into account and then come out his own thinking rather than linearly combine all of them.

- The assumption that user shares similar tastes with friends may mislead the recommendation. For example, a user can connect with people who have different shopping preferences.

- Most of existing approaches ignore that users have different knowledge in different domains.

Table 2 shows some widely used datasets for the social-aware recommender systems[76], which are taken from popular social networking websites[3].

3. Robustness of Recommender Systems

3.1. Overview

The recommender systems promote the efficiency and benefits for both customers and merchants. Although effective, the recommendation schemes are vulnerable to shilling attacks or noise. For example, merchants may

---

[3]The links for each dataset can be found in the Librec website https://librec.net/datasets.html

hire a group of spammers to insert their profiles and fake ratings into the systems, which will affect the performance of the recommendation[25] and also the customer's trust on recommender systems. Detecting such attacks and designing a robust recommender system are very important. Generally, researches in this field include shilling attack detection techniques and robust recommender systems.

3.1.1. Attack Types

The shilling attacks can be classified by the type of attackers, the intent for the attack, the knowledge of the attack, etc. For example, according to the intention, the attacks can be categorized into push attack, nuke attack, and random attack[11]. The first intends to increase the popularity of the items while the second intends to decrease the popularity. According to the knowledge-cost, the attacks diverge into high-knowledge attack, i.e., the attackers get some knowledge about other normal users, and low-knowledge attack.

The attack profile consists of the history rated items and generally includes four parts: the target items, selected items, filler items and unrated items. The target items I_T, which will be either "push" or "nuke" ratings, are rated with a rating function σ_T. The selected items I_S are rated by the attacker with particular intentions, e.g. the group attacks. The filler items I_F include randomly chosen items that to make the profile look normal and harder to detect. Also, We denote the unrated items by I_U[11]. Different parts may have different generative functions for getting the ratings.

I_T			I_S			I_F			I_U		
i_T^1	...	$i_T^{N_T}$	i_S^1	...	$i_S^{N_S}$	i_F^1	...	$i_F^{N_F}$	i_U^1	...	$i_U^{N_U}$
$\sigma_T(i_T^1)$	...	$\sigma_T(i_T^{N_T})$	$\sigma_S(i_S^1)$	...	$\sigma_S(i_S^{N_S})$	$\sigma_F(i_F^1)$	...	$\sigma_F(i_F^{N_F})$	null	null	null

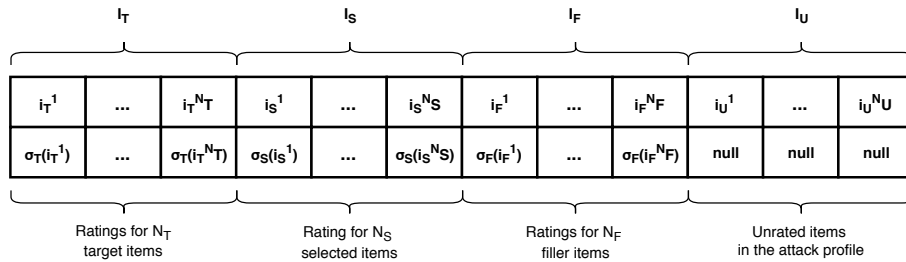| Ratings for N_T target items	Rating for N_S selected items	Ratings for N_F filler items	Unrated items in the attack profile

Figure 7: The general components of an attacker profile

3.1.2. Research Fields

The two directions to reduce the effects of shilling attacks on recommender systems include shilling attack detection techniques and robust algorithms.

The former is first detecting the attacks, filtering the attack profiles, and then constructing the recommender systems. The latter refers to construct attack-resistant recommender systems, i.e., robust recommendation methods[11].

Shilling Attack Detection Algorithms are mainly discussing the way of detecting malicious user files. According to the research targets, the shilling attack detection algorithms can be used for detecting point (personal) attacks or collaborative (group) attacks. The point attacks may represent an irregularity or deviation that happens randomly and may have no particular interpretation. Also, according to the nature of input data, we may utilize sequential (e.g., textual information and time series) or non-sequential data ((e.g., images, user profiles). Techniques used for shilling attack detection can be roughly classified into statistical methods, supervised classification methods, semi-supervised methods, and unsupervised clustering methods. The *statistical methods* are focusing on detecting the outlier items. For example, statistical testing is used for identifying the differences between the sample distribution. Zou et al. [8] introduce a probabilistic inference network and the Belief Propagation (BP) algorithm [77] to perform inference efficiently. For the *supervised classification methods*, most of the work conducts feature engineering first and then design the algorithms. Features such as rating deviation, the similarity with top neighbors are considered. For example, Yang et al. [9] propose three new features, i.e., the filler size with maximum, minimum and average ratings, on filler or selected items to identify the attack profiles. The features are analyzed with statistical tests and classified by a variant of AdaBoost method. *Unsupervised clustering approach* is normally clustering the users into groups and then eliminate the suspicious users. For example, Bhaumik et al. [10] apply k-means clustering on user profiles and identify the small clusters as attacker groups.

Robustness of Recommender Systems are focusing on developing attack-resistant systems, which is trying to reduce the influence of shilling attacks. Current robust recommendation algorithms mainly lie on two aspects: constructing robust algorithms or taking trust relationships into consideration. We have discussed the latter one in the previous section. Thus, here, we majorly introduce the former one: robust algorithms for recommendation. Some work adopts matrix factorization for the methods. For example, Alonso et al. [78] use matrix factorization based method to identify the shilling attacks. According to the observation that fake ratings are occurred during a short interval of time, they assume that malicious profiles will affect the reliability of the model in an anomalous way. They use two

matrix factorization models to obtain the real prediction errors and the estimated prediction errors; the second error is used for evaluating the prediction reliability. Zhang et al. [79] incorporate the R1-norm into the loss function to improve the robustness. They claim that the squared error function is sensitive to large residuals. Yu et al. [80] design a robust matrix factorization model with kernel mapping and kernel distance. He et al. [81] consider to improve the robustness of the recommender system by adding an adversarial module to the training.

3.2. Deep Learning-based Shilling Attack Detection Algorithms

For shilling attack detection problem, a key point is to evaluate the suspiciousness of users, posts and reviews. One advantage of using deep learning for detecting the shilling attacks is its capability of capturing complex structures in the data. Also, there is a need for large-scale detection techniques as the volume of data increases in the real world cases. Different from traditional methods, deep learning-based approaches need less manually defined features and thus solve the problem in an end-to-end trainable way[82].

Some use convolutional neural networks for mining the local features. Convolutional neural networks work as feature extractor; it can learn from local representations and map these representations into higher or lower dimensional representations for further uses[83]. Many studies mine textual information for detecting the suspicious ratings/reviews[84]. For example, Zhang et al. [85] propose a deep model to identify the review spam. They assume the fraudulent users lack real experience, while the normal users are with real experience; then, the textual information should indicate different patterns between the fraudulent users and normal users. Li et al. [86] use word vector to represent the textual features, and use CNN to learn the semantic representation. Some work also considers the user's behaviors [87]. Wang et al. [88] consider the cold-start problem for new-coming users. They use CNN to learn the embedding from both textual and behavioral information.

Some use RNN-based methods for targeting the sequence input. Recurrent Neural Network (RNN), which has the function of memory, has shown its efficiency for processing sequential information. The following work, such as Gated Recurrent Unit (GRU)[89] and Long Short-Term Memory (LSTM)[90] are further designed for solving the gradient vanishing problems. As for detecting malicious ratings or reviews, the RNN based methods are used for learning sequential patterns such as texts and time series. For example, Ren

24

et al. [91] use CNN to learn from words and use a bidirectional-GRU for learning the sentences. The learned representation is further be used for prediction. Similarly, Wang et al. [92] use LSTM to learn from texts.

Some consider hybrid methods or multiple inputs. For example, Wang et al. [93] propose a hybrid method that learns from both the review content and product information. They use a tensor factorization algorithm to learn the latent representations from reviews and products. Then, the learned representations are further combined with deep learning-based classifiers. Dong et al. [94] use autoencoder to extract latent representations from textual information and user behavioral patterns. Aghakhani et al. [95] improve model performance by adding adversarial noises.

3.3. Deep Learning for Robustness of Recommender Systems

3.3.1. Introducing noises to recommender systems

An intuition way for enhancing the robustness of recommender systems is adding noises in the recommendation training process. By doing so, the model is forced to learn robust parameters to improve the denoising capability.

Randomly generalized noises. Some may add man-made noise into the input. For example, the model may be added with extra bias terms; the inputs are corrupted before feeding in the models. By doing so, the model is forced to learn the most informative and robust parameters to improve the robustness of the recommender systems. One example is the denoising auto-encoder (DAE) [12], which corrupts the inputs with adding noises. Wu et al. [96] propose the collaborative denoising auto-encoder (CDAE) that has similar ideas of DAE. They first corrupt the inputs, i.e. the ratings, with Gaussian noises, feed the inputs into neural nets and get a dense representation of such corrupted inputs. The decoder of the model tries to recover the original values of the dense representation and thus predicts the ratings for recommendation. Strub et al. [97] also corrupt inputs by stacked denoising autoencoders[60]. Besides, they incorporate the side information, such as the user profile (age and gender) and the movie category, to enhance the robustness of the model. Wang et al.[98] propose collaborative recurrent autoencoder that integrates RNNs and denoising autoencoder for recommendation. They design a robust recurrent network to process the item textual information and overcome the shortage of using man-crafted features. In details, the recurrent network is designed in the autoencoder way, i.e. the layers

in the encoder and decoder are recurrent networks. By doing so, the recurrent autoencoder can learn both the sequential information and the dense representation of inputs. The learned dense representation is regarded as the item representation and further combined with user representation for rating prediction. Furthermore, in case of over-fitting, they design a denoising and a beta-pooling approach.

Adversarial noises. In recent work, some attempt to add adversarial noise to the model to improve the robustness. For example, He et al. [81] consider to solve the problem by adversarial training based on Bayesian Personalized Ranking. In details, they corrupt the model parameters with adversarial noises; the adversarial personalized ranking is made by minimizing the training loss and maximizing the adversarial loss, i.e. to identify the worst case of the corruption. The model is optimized via stochastic gradient descent. Similarly, Yuan et al.[99] propose an adversarial training framework for recommendation. The model is designed based on the collaborative denoising autoencoder. Different to traditional CDAE method that corrupts the inputs, they insert a noise mixing layer into the autoencoders. The addressed adversarial training strategy includes: a training step to obtain optimal parameters; and a re-training step to minimize the training loss while maximize the adversarial noise loss. Wang et al. [100] consider a session-based recommendation problem and design a memory network for storing the long term and short term user preferences. They use generative adversarial nets to generate negative samples to improve the model parameter inference. Wang et al.[69] propose a generative adversarial model for recommendation. Similar to generative adversarial network, the model includes two modules: the generative model simulates the real user profiles by capturing the patterns from the raw datasets; while the discriminative model tries to identify such generated samples from the real ones. In such case, the generator do similar work as the malicious users, and the discriminator identifies the malicious content; thus, they promote the both performance and improve the robustness of the recommender system.

3.3.2. Other Methods

The attention mechanism is able to filter the uninformative information from the input and thus prevent the side effects of noise. Many works consider to mine informative patterns in user history records. Jhamb et al. [101] examine from user preferences that they propose Attentive Contextual Denoising Autoencoder and use attention mechanism for encoding contextual

attributes of user preferences. Zhou et al. [102] consider from the heterogeneous user behaviors; they use self-attention algorithm for predicting the user preferences by aggregating the different contributions of the user behaviors. Loyala et al. [103] study the user transitions in different sessions by RNN-based method and use attention module for learning the more expressive portions of the sequences. Ying et al. [104] also use attention mechanism for a dynamic situation of user preferences. Two attention layers are utilized to learn user's long-term preferences and learn from both long-term and short-term preferences, separately. Liu et al. [105] provide similar ideas for incorporating long-term and short-term preferences. Some works consider from other patterns. Seo et al. [106] mine the potential benefits from textual features. They build vector representations of user and item using attention-based CNNs, where the attention mechanism is used for extracting keywords before the CNN modules; such vector representations will be further used to predict the ratings. Tay et al. [107] focus on the user-item relationships and use an attention module that can visualize the model and enhance the model performance with capturing the significant patterns. Chen et al. [108] propose two attention modules that one for selecting informative components of multimedia items and one for scoring the item preferences.

Besides, incorporating auxiliary knowledge from other domains, such as social relationships (we have discussed in section2), can also help improve the robustness of the recommendation.

3.4. Summary

Shilling attacks detection has been the traditional research field to tackle the robustness of recommender systems, by filtering or removing malicious profiles. Some other work designs robust machine learning methods to neutralize the malicious profiles impact, e.g. adding man-made noise in the system to improve the robustness. Both ways show the improvement in the recommendation performance. However, there are still several challenges in this area.

- It is hard to unify such two methods in a trainable end-to-end model for leveraging both capabilities. And most of the deep learning methods are sensitive to the data resource and the cross-domain conditions.

- Most of the current work does not consider the dynamic conditions, i.e., anomalous behavior may change over time.

- The anomalies are rare entities in real life. Thus, it is challenging to obtain labels.

4. Explainable Recommender System

4.1. Overview

As another approach to enable trust-aware recommendation, explainable recommender systems address the problem from a different perspective[13]. Unlike other personalized recommendation algorithms, explainable recommender systems offer reasons on why the systems provide users with such recommendations and also give guidance to system designers to improve the recommendation results. It not only improves the effectiveness and user satisfaction of recommendation systems but also enables the systems to generate trustworthy recommendations. Recently, a host of explainable recommendation approaches have been proposed, including but not limited to matrix factorization, deep learning, association rule mining, topic modeling, knowledge-graph models. Despite such variety, these methods can be divided into two groups in general. *Post-hoc* [109, 110] methods do not modify the recommendation algorithm itself, but attempt to explain the results, such as "this item is the most popular" and "people like the same item as you do also bought." These methods often can not explain the recommendation mechanism, and the diversity of explanations is limited. On the contrary, *Embedded methods* [111, 112, 108] design explanation-oriented recommendation models so that the recommendation process itself can automatically generate explanations which are normally selected from the side information, e.g., texts or images. In this section, we narrow down our focus on only deep learning models, which belongs to the family of embedded explanation methods. Deep learning has recently become very successful in recommendation tasks[1]. Similar to other embedded methods, we find that the majority of recent work based on deep learning leverages text or image information, e.g., user reviews, product photos, and movie posters for explanation generation. Most of them are proposed to explain a specific recommendation model, but, recently, some work addresses the recommendation explainability from a model-agnostic perspective [113, 114]. For instance, Wang et.al.[113] employs reinforcement learning to explain any recommendation model. Therefore, in this section, we classify the previous work into five categories:

1. traditional explanation based on collaborative filtering;

28

Table 3: Summary for Deep Learning-Based Explainable Recommender Systems

Method \ Data Type	RBM	Attentive CNN	Attentive RNN	Generative RNN	GAN	Memory Networks	Others
Ratings Only	[115]						
Textual Reviews		[21], [111], [116], [117]	[118]	[119], [120], [121], [122], [123], [124]			[113], [125]
Images		[108], [112], [126]			[127], [128]		[129], [16], [48]
Temporal Data		[120], [130]				[131]	

2. explanation using a piece of textual sentence;

3. explanation via visual contents;

4. explanation via temporal dynamics;

5. other deep learning-based explainable recommendation models.

4.2. Explanation on Collaborative Filtering

In the very early days of recommendation explanation research, collaborative filtering (CF) serves as the fundamental method for personalized recommendation. CF leverages the users' implicit or explicit feedback from which explanations can sometimes be generated in a very straightforward way. For example, in user-based CF, the system decides whether or not to provide a certain user with an item according to the ratings from his/her neighbors, which can be considered as a form of explanation. Similarly, in item-based CF, explanations are generated for a target user based on whether the rating given to an item is similar to the other already-given ratings. However, this is not the case for deep learning-based CF. Recently, with a number of deep learning-based recommendation models being proposed, the state-of-the-art performance in various recommendation tasks, such as rating prediction, top-N recommendation, and sequential recommendation, has been dominated by deep approaches, but most of them lack the ability to explain their recommendation results because what the deep neural networks have learned is normally hard to interpret. Abdollahi and Nasraoui [115] focuses on the interpretability of Restricted Boltzmann Machines (RBM) based CF recommendations without relying on any auxiliary data, such as item content or user attributes. In this paper, similar to the idea of explainable user-based CF, for each target user, the authors introduce the concept of "explainability score" which is calculated from the rating distribution inside his neighbors that are determined using the cosine similarity. This score is ranges from zero to one. Only when a score is greater than zero, the item is explainable for
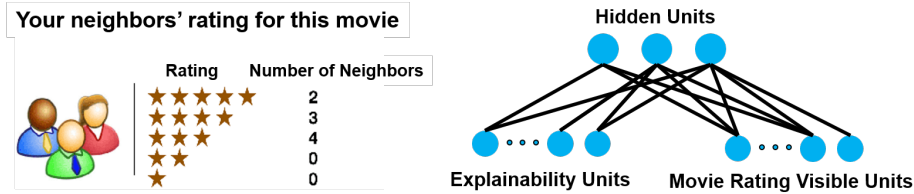
29

Figure 8: Explainable Restricted Boltzmann Machines: explanations with relevant users (left), conditional RBM for explainability (right).(Abdollahi and Nasraoui[115])

a user. The higher the score, the more explainability is achieved. Then, the authors employ the conditional RBM model with an additional visible layer that has exactly the same number of hidden units as the number of items (see figure 8). The output value of each hidden unit in this layer is limited within 0 and 1, representing the above explainability score. In this way, the conditional RBM model tends to recommend items that are explainable. In essence, this approach provides explanations via user-based neighborhoods.

4.3. Explanation on Textual Data

In recommender systems, textual contents are a common and major source of auxiliary information such as user reviews and product descriptions. Numerous deep learning techniques have been adopted to exploit textual data, such as CNN [21, 111, 117], RNN [122, 123, 16], and attention mechanism [108, 116, 118]. Among the above methods, CNNs are often employed for deep feature extraction from text and are combined with an attention mechanism to generate explanations[21, 111], while RNNs are often used for textual explanation generation[124, 123]. Here, we review these deep learning-based explainable recommendation models that exploit textual side information, and analyze their strengths and weaknesses.

The first group of methods combines CNN with attention mechanism to analyze the textual data. Seo et al. [21] aggregates all the review texts given by a user and an item respectively to form two sets of representations from which the abstract features of different users and items are learned via convolutional neural networks with dual attention mechanism (i.e., global and local attention), as shown in figure 9(a). The predicted ratings are then generated from the learned features similar to that of matrix factorization. In the meantime, the dual attention networks enable word focusing in the review texts. Different from the architecture in [21], Chen et al. [111] proposed a neural attentional regression model with review-level explanations

(NARRE) which employs the DeepCoNN network to process the reviews [132], shown in figure 9(b). Although NARRE uses only a single attention layer on the output of DeepCoNN network, it is not only capable of generating highly-accurate prediction ratings, but can also choose useful reviews that offer a form of review-level explanations to the target users. The above two methods use review texts as inputs only. In other words, they ignore the user-item interactions, and thus fail to completely model the users' rating behaviors. Wu et al. [116] proposed a context-aware user-item representation learning model (CARL) to overcome such shortcomings. CARL fuses two different networks, one for review feature extraction and the other for user-item interaction feature extraction. To process reviews, CARL employs an attentive CNN neural network, and to model user-item rating interactions, it adopts a matrix factorization-like approach to learn user/item latent representations. The final predicted ratings are fused by a dynamic weighting scheme between the outputs of the two networks. Unlike the aforementioned static methods with attention applied only on textual data, Chen et al. [117] built a dynamic explainable recommender (DER) by combining a gated recurrent unit (GRU)-based network that models user dynamic ratings with a sentence-level CNN to profile an item by its reviews. DER applies attention on the mixture of a user's time-varying preference at a certain time and the sentence-level features of the item reviews, merging sentence embeddings under "user-aware" attention weights. Therefore, DER can provide explanations in a dynamic, personalized manner. Despite the model differences, in all the above approaches, the recommendation explanations are produced in the form of a group of words with high attention weights to help the user understand the recommendations, shown in figure 9(c)(d).

The second group of methods exploits RNN [133], a very effective family of deep neural networks for natural language processing. Some approaches introduce attention mechanism into RNNs for similar reasons as the above attentive CNN-based models to select highly-relevant words from the review texts as explanations [118]. Most existing models take advantage of the generative ability of RNNs to produce user/item review explanations [120, 121, 124, 119, 122, 123, 16]. Cong et al. [118] proposed a hierarchical attention-based network(HANN) that generates explanations by considering the contribution of reviews to the overall ratings at two levels, the word level and the review level, shown in figure 10. HANN is similar to Seo et al. [21] in that HANN replaces the CNN by GRU-RNN. HANN also splits the textual data into user reviews and item reviews that are further fed into two sepa-
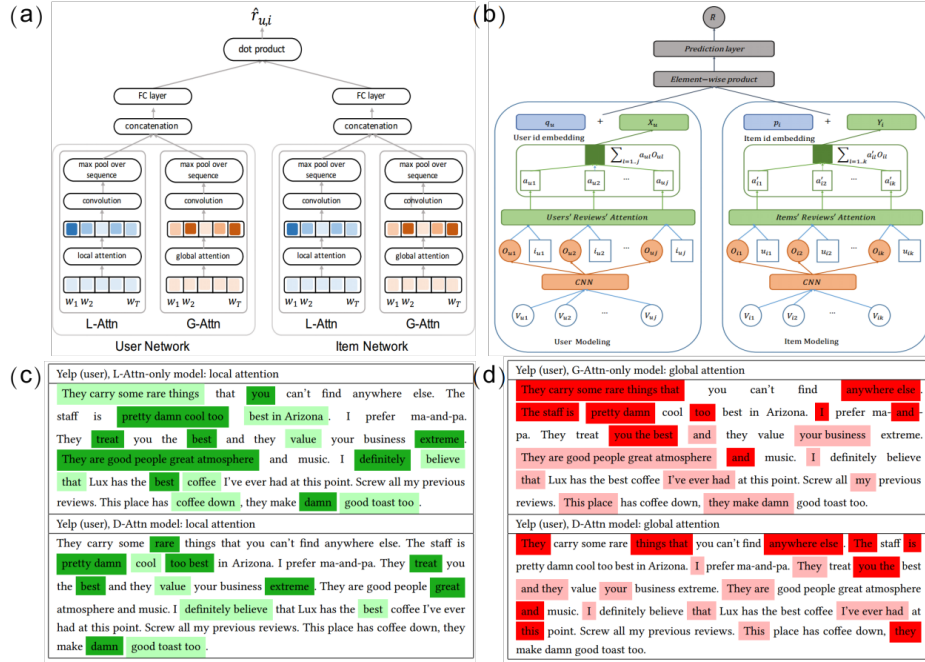
31

Figure 9: Examples of attentive CNN networks for explainable recommendation. (a)Dual-attention neural networks (Seo et al. [21]); (b)Neural attentional regression networks (Zheng et al. [111]); (c) and (d) Words in a user's review text annotated according to the learned attention scores(Seo et al. [21]).

rate GRU-based deep neural networks, known as the user net and the item net. Dual attention is adopted, one at word level for intra-review attention and the other at review level for inter-review attention. Both nets are fused together by fully-connected layers to predict the ratings. In figure 10(b), the explanations are generated using the attention scores at both levels. The darker the pink color, the higher the attention score reaches at the review level. Word level attention scores are similarly denoted by the green color. In this way, HANN extracts useful words from the reviews to form the explanation and meanwhile, globally distinguishes the effectiveness of reviews on the final predicted rating scores.

Due to the powerful text generation ability of RNNs, a host of RNN-based explainable recommender systems are proposed to leverage this feature. Costa et al. [119] designed a character-level generative concatenative network based on LSTM cells [90], where the ground-truth ratings serve as an auxiliary information and are concatenated into the input layer. There-
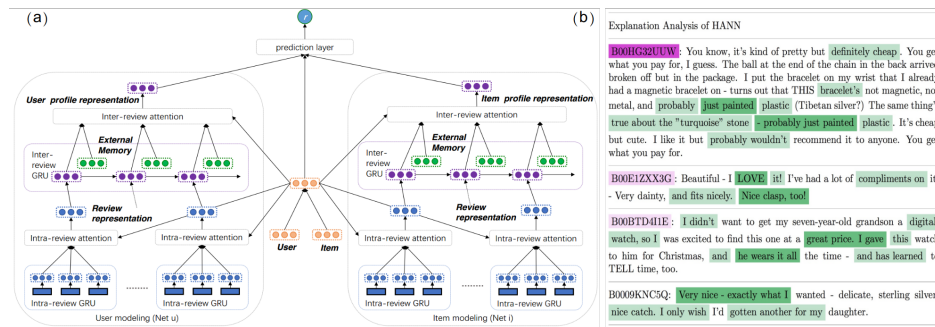
Figure 10: An Example of attentive RNN networks for explainable recommendation. (a)Hierarchical attention-based network (HANN); (b)Explanation analysis of HANN.(Cong et al. [118])

fore, the model is able to generate reviews following the directions pointed by the rating scores (figure 11(a)). By adjusting the hyper-parameters, the model is able to provide very natural explanations for human readers shown in figure 11(e). Instead of generating reviews, Li et al. [120] proposed a multi-task learning model, i.e. the neural rating and tips generation network (NRT). NRT takes ratings and reviews as context and produces abstract tips, as shown in figure 11(b). User-item rating pairs are first used to learn the user/item latent factors via multi-layer perceptrons (MLP) that forms the rating regression network. These latent factors are then fed into a standard MLP-based review text generation network whose output layer, together with the predict ratings, serves as the context of a GRU-RNN based tip generation network. The multi-task objective is then composed of the rating regression loss, the review generation loss, and the tip generation loss. The generated tips are both concise and vivid enough to predict users' possible experience and feelings. Another multi-task recommendation model is proposed by Lu et al. [121]. The authors utilize the adversarial sequence to sequence learning technique. The reviews are first encoded into latent feature vectors using a bidirectional GRU-RNN network and then decoded by a single GRU-RNN network, creating a review autoencoder structure. This autoencoder is adversarially trained with a CNN-based review discriminator to identify if a piece of given review is written by user i on item j. Unlike NRT that considers ratings as the context of the explanation generation process, Lu et al. the latent textual features are fed into a matrix factorization rating prediction algorithm as the context. Both models are jointly trained using the alter-

nating least squares (ALS) technique [134] to perform rating prediction and explanation generation. Furthermore, the RNN-generated reviews not only provide explanations but can also act as inputs for the recommender system. To test whether the generated reviews are more effective in recommendation than human-written reviews, using DeepCoNN [132] as the recommender, Ouyang et al. [122] compare human-written reviews with synthetic reviews that are produced at both character and word levels by popular review generation models (figure 11(d)). Results show that synthetic reviews can carry more consistent information appropriate to the demands of a recommender system than human-written reviews, justifying the feasibility and rationality of using generated reviews to explain recommendations. Instead of using reviews or tips as inputs, Zhao et al. [123] feed user/item side information (e.g., user/item tags, item title, user gender, etc.) into a recurrent attention generation network to produce reasons for the explainable recommendation in conversation applications. Similarly, Suzuki et al. [124] adopts a MLP network to encode multicriteria evaluation ratings (e.g., overall rating, location rating, service quality rating, price rating, etc.) into a latent vector which is then decoded by an attentive LSTM-RNN network into reviews so that personalized explanations are generated for the predicted ratings.

4.4. Explanation on Visual Data

Compared with textual data, visual contents often contain more information that can be exploited for recommendation explanations. Most previous image-based recommendation approaches transform images into latent representation vectors to be incorporated into recommendation algorithms [135, 136, 137, 138]. However, such approaches are hardly useful in ex planing why a particular item is recommended. Recently, some initial steps have been taken towards visual explainability of recommendations via exploiting the power of deep learning. Most existing recommendation models using visual data adopt CNN as the building block, given its popularity and success in processing visual data. Other deep learning techniques employed to enable explainable recommendations include the attention mechanism [108, 112, 126] and generative adversarial network (GAN) [128, 127]. Apart from generating explanations from the visual data itself, some approaches regard images as a source of auxiliary information to help explain the recommendations [16, 129, 48]. Unlike the textual data based explainable recommendation models, all the above approaches explain the recommendation results in a straightforward manner.
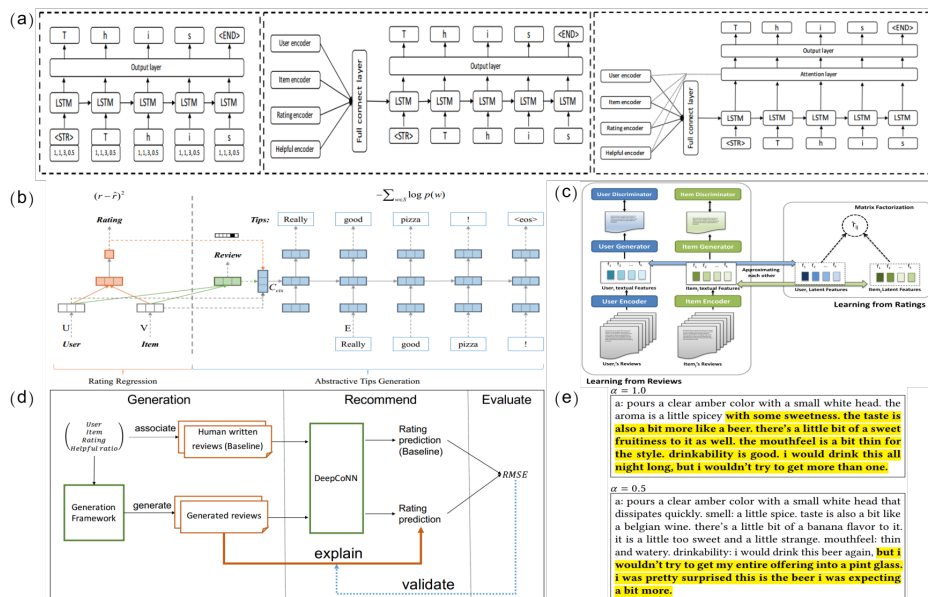
Figure 11: Examples of generative RNNs for explainable recommendation. (a) Generative concatenative networks, context-aware generation model and attention enhanced generating model (Ouyang et al. [122]); (b) Neural rating and tips generation (NRT) network (Li et al. [120]); (c) Multi-task learning model for simultaneous rating prediction and review generation (Lu et al. [121]); (d) Validation setup of recommendation performance of generated reviews (Ouyang et al. [122]); (e) Generated textual reviews for explanations. With different parameters, the model generates different explanations. (Costa et al. [119])

Similar to those models that leverage attention mechanism for textual data, the first type of visually-explainable recommender systems applies attentive deep neural networks to select a group of "physical regions" [108, 112] or "semantic regions" [126] from the images as the explanations. The earliest attempt we can find is from Chen et al. [108], in which work the attentive, collaborative filtering (ACF) model is proposed via hierarchical attention at both component level and item level, as shown in figure 12(a). ACF combines the latent factor model with an attentive neural network that processes the features from items to provide top-N recommendations using implicit feedback. The item features are extracted by a CNN-based deep network, ResNet-152 [139], from images or video frames. After processed by the dual attention network, these features are merged with user latent factors via element-wise addition to reflect the users' detailed preferences. Bayesian Personalized Ranking (BPR) [140] is adopted as the last step to generate the

final recommendations. Figure 12(b) shows an example from the ACF's results. The explanations are given by the attention weights where the higher the weights, the more probable the user will like the entire images or the regions of images. Chen et al. [112] exploits both item images and user's textual reviews whose features are extracted via VGG-19 and GRU-RNN, respectively. The VGG-19 produced image features are split into a number of regions which are then passed through an attention layer for explanation and merged with the item latent factors to represent the items. The user and item latent factors are combined with the item representation vectors to serve as the inputs of the GRU-RNN network for review generation. This model is named review-enhanced visually explainable collaborative filtering (Re-VECF), shown in figure 12(c). Unlike ACF that aims at user representations, Re-VECF focuses on item representations and uses a single attention layer with the adoption of element-wise multiplication to merge the image features and item latent factors. Figure 12(d) illustrates some explainable recommendations, where the bolded italic words (e.g., sleeve) mean that the word generated by Re-VECF was also mentioned in the true review, and the word is aligned to the boxed area of the image learned by the attention mechanism. Different from the above two approaches that exert attention on "physical regions" of the images, Hou et al. [126] proposed the Semantic Attribute Explainable Recommender System (SAERS) to understand users' semantic preferences via integration of the Fine-grained Preferences Attention (FPA) mechanism and the Semantic Extraction Network (SEN) for fashion recommendation. SAERS converts each attribute extracted from a particular region of the clothing images into one dimension in a semantic attribute visual space. As shown in figure 13(a), SEN consists of the CNN-based ResNet-50 [139] network for semantic attribute classification and the Gradient-weighted Attribute Activation Maps (Grad-AAM) [141] for location and extraction of attribute representations in a weakly-supervised manner. The authors then adopt FPA to align the user latent factors with the semantic attribute visual space. Each user latent factor is concatenated with one transferred semantic attribute representation vector, upon which the attention mechanism is applied to learn the user's preferences over different semantic attributes. Finally, BPR is used for recommendation, where the ratings are predicted via the inner product of user and item latent factors. Figure 13(b) shows several visual explanation examples from SAERS. The learned attention weights are demonstrated in the red boxes. The weights indicate how much the user prefers a particular attribute. For instance, when a dress is recommended to
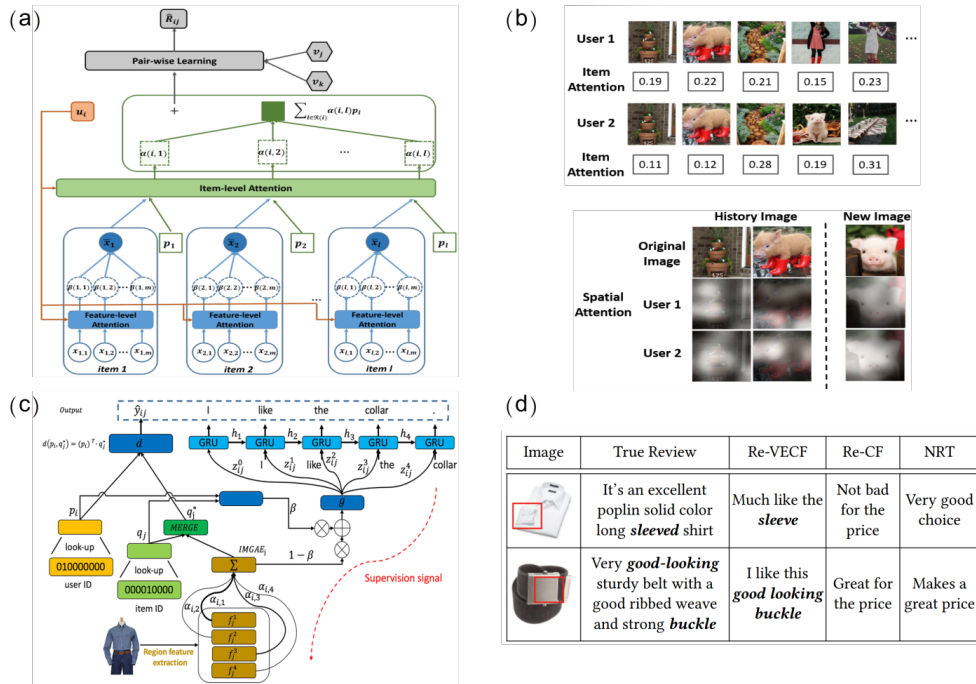
36

Figure 12: Examples of attentive CNN networks for explainable recommendation. (a) Attentive collaborative filtering (ACF) (Chen et al. [108]); (b) Visualization results of the dual level attention given by ACF. At the item-level, the value under each item represents the attention weight of the item, while for the component-level, a heat map is used to represent the attention value, in which the darker the color is, the lower its represented attention value is (Chen et al. [108]); (c) Review-enhanced visually explainable collaborative filtering (Re-VECF) [112]; (d) Generated reviews from Re-VECF compared with the true reviews [112].

user C, the model explains to her that this dress has a V-shaped neckline, which is reasonable because according to user C's purchase history, she has bought three V neckline dresses before. Therefore, the recommendations are visually explained, improving the trust of the system.

The second type of visual explainable recommendation approaches exploits GAN. Similar to SAERS, Kang et al. [127] addresses the fashion recommendation by employing the Siamese CNNs [142] to extract "fashion-aware" image features to give the notations of "style." Although this is enough to provide explanations from a certain aspect, the authors further adopt conditional GANs [143] to generate images leveraging semantic inputs, where the product's top-level category is chosen as the condition. Thus, this approach
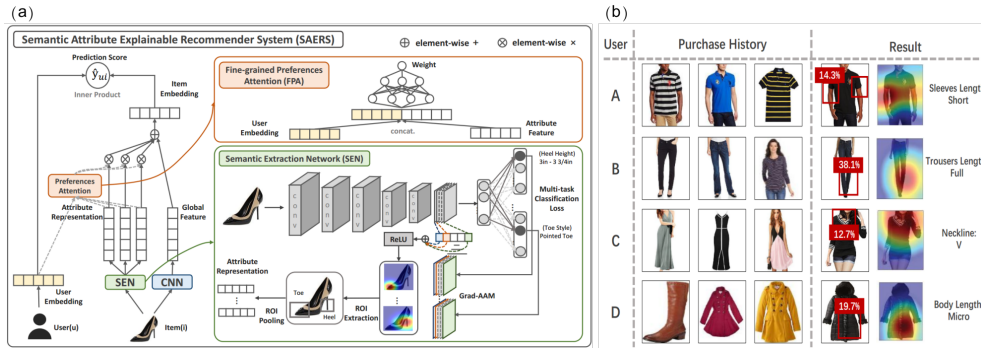
Figure 13: The semantic attribute explainable recommender system (SAERS). (a) The architecture of SAERS; (b) Examples of the visual explanations from SAERS. (Hou et al. [126])

can generate novel items that is likely to satisfy the users and are not in the training dataset so that the explanations for a series of recommended items can be summarized into such generated images. Kumar et al. [128] tackles the pairing problem in fashion recommendation via an enhanced conditional GAN model called c^+GAN. Given one piece of clothing image, the model recommends a set of items that best match the given clothing in a generative manner. c^+GAN modifies the generator with a classical mean squared error (MSE) loss and also a simplified perceptual loss using discrete cosine transform (DCT) coefficients of the generated as well as the target images. A simplified lensing technique [144] to the discriminator is applied to stabilize the generator training. Equipped with these techniques, c^+GAN is able to generate very meaningful fashion items as recommendation explanations.

Bharadhwaj [129] adopts the content-based similarity approach to recommend images. The authors modify the VGG-16 network by layer-wise relevance propagation [145], which enables relevance conservation at each layer in a pixel-wise manner. Given the purchase history of the target user (query items), the model can generate a list of other items that most resemble the query items. The explanations are given by the highlighted pixels that are the most informative for inferring which items go along well with the query item, shown in figure 14 (a) and (b).

Other explainable recommender systems consider the visual features extracted via deep neural networks as the auxiliary information to generate non-visual explanations. Lin et al. [16] applied multi-task learning to recommendation and proposed the neural outfit recommendation (NOR) model.

Figure 14: Examples of some other image-based recommendations. (a) and (b) Recommendation via layer-wise relevance propagation. The left-most image is the input query item and the rest are recommendations. On the original images, the generated heatmaps are shown (Bharadhwaj [129]); (c) Overview of the neural outfit recommendation (NOR) model; (d) A few examples of recommendations and the corresponding generated comments for explanation from NOR (Lin et al. [16]).

NOR recommends outfits to users with abstractive comments generated as explanations. To achieve both tasks, as shown in figure 14 (c), NOR adopts two neural networks, i.e., the outfit matching network and the comment generation network. Equipped with the mutual attention mechanism, the outfit matching network utilizes CNNs for visual feature extraction. These abstract visual features are further transformed into rating scores to predict the most matched outfit. To generate textual explanations from the aforementioned visual features, the authors exploit the cross-modality attention over the above CNN network and a GRU-RNN network, leading to the comment generation network. Figure 14 (d) shows some examples from the recommendations produced by NOR. The recommendation explanations are given in the form of generated comments which are mostly focused on the general opinions on the matched pair of outfits. Wu et al. [48] exploited the hierarchical attention mechanism in the field of image recommendation in social networks. The proposed model leverages heterogeneous data, e.g., users' rating behaviors, social network, upload behaviors, images. And from such complex relationships between users and images, the model represents these contextual factors as different sets of embeddings. A hierarchical at-

tention network is then applied to attend differently to various embeddings. Images are represented using their content vectors, extracted by VGG-19, and their style vectors, generated by a CNN-based synthesis method [146]. However, this model does not generate explanations straightforwardly, but the recommendations can be intuitively interpreted by the learned attention weights of different aspects.

4.5. Explanation on Temporal Data

Sequential recommendation takes advantage of the temporal characteristics that exist in user dynamic behaviors to improve recommendation effectiveness [147]. The temporal aspect provides another dimension to generate explanations for the recommendations. A user's history is no longer a collection of unordered items, but a sequence of time-aware items. The order itself can provide certain explanations. For instance, if item i and j are complementary, a user bought item j at a certain time might be explained by the fact that this use had bought item i sometime earlier. With such observation, Li et al. [120] proposed the Neural Attentive Recommendation Machine (NARM) to learn the user's primary intention in the current session (figure 15(a)). NARM employs GRU-RNN as the basic building block. NARM contains a global encoder that interprets the last hidden state in the RNN as the user's behavior feature, and a local encoder that interprets all the hidden states in the current session as the user's primary purpose feature. Attention mechanism is applied to the local encoder to learn different weights for the hidden states so that the model can tell which past items contribute more to the future items. The two encoders are then combined as inputs of the decoder, which predicts the recommendation possibility of each candidate item. In figure 15(b), we can clearly see the effect of the attention mechanism. In a particular session, the importance of items are reflected by the depth of the colors. This can give certain explanations on the next recommended items. To be specific, the users' decision on the next clicked items are more influenced by those near the end of the session than those at the start, which is consistent with people's purchasing or browsing behavior we have noticed in reality. Tang et al. [130] proposed a Convolutional Sequence Embedding Recommendation Model (Caser) as another solution to the sequential pattern extraction problem. As shown in figure 15(c), Caser embeds a set of recent items into a two dimensional matrix whose dimensions represent the time and latent feature respectively. Two convolutional filters, one vertical and the other horizontal, are then applied onto this matrix to

learn sequential patterns that are expressed as local features. The two filters capture patterns at different levels. Horizontal filters aim for union-level patterns via unifying the data into multiple sizes, while vertical filters aim for point-level sequential patterns by calculating the weighted sums using the previous items' latent representations. For clarity, visualization of vertical filters is shown in figure 14(d) that reflects the importance of different past items. Union-level sequential patterns can be effectively extracted by the horizontal filters (figure 14(e)), where the recommended $\hat{R}_3$ (ground truth) is generated by the union of S_3, S_4 and S_5 due to the same genre they belong to. If any of S_3, S_4 and S_5 is masked in the horizontal filters, the ranking position of $\hat{R}_3$ is largely reduced. Chen et al. [131] took advantage of the memory mechanism for long-term memory and integrated collaborative filtering into a memory-augmented neural network (MANN). MANN stores and updates users' historical records explicitly and is also able to extract the intuitive patterns of how users' future actions are influenced by their previous decisions and behaviors. Shown in figure 16, MANN can capture two different types of sequential patterns. "One-to-one" behavior pattern generates sequences where the next action is only influenced by the most recent action. "One-to-multiple" behavior pattern generates sequences where a set of continuous behaviors are influenced by the same previous action. Both patterns can be widely observed in practice. For example, when browsing web pages, one may keep following the related links on each page and form a "one-to-one" pattern, and when searching key words through a search engine, one may browse multiple pages related to the same key words, leading to a "one-to-multiple" pattern. The discovery of such patterns by MANN can act as the explanations for why a particular user will buy a certain item in the future.

4.6. Other Approaches

Wang et al. [113] proposed a reinforcement learning framework for explainable recommendation, which is quite universal because instead of integrating a certain explanation mechanism into a recommendation model, it has no restrictions on the details of the model to be explained. The authors consider users, items, side information, and a recommendation model to be explained as the environment. Two couple agents are employed, one for explanation generation and the other for explanation discrimination. At each state, the generator agent gives a piece of explanation taking the user-item pairs as inputs, while the discriminator agent takes the generated explana-
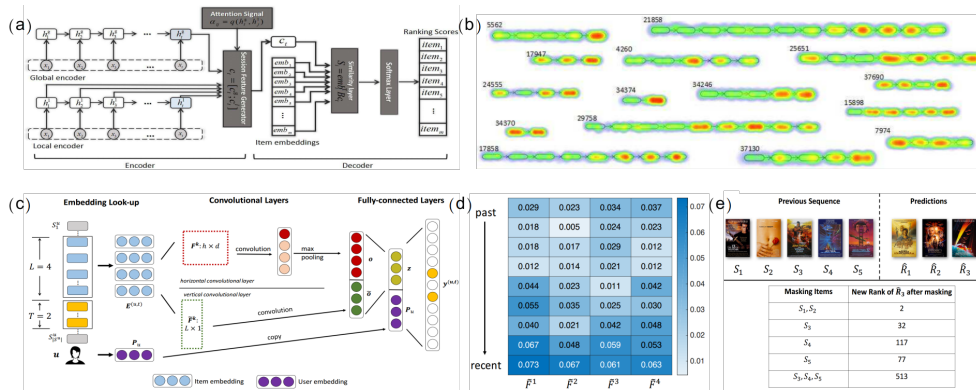
Figure 15: Examples of sequential explainable recommender systems. (a) The architecture of neural attentive recommendation machine (NARM) (Li et al. [120]); (b) Heatmap visualization of items weights by NARM (Li et al. [120]); (c) The network architecture of convolutional sequence embedding recommendation (Caser)(Tang et al. [130]); (d) Illustration of the vertical convolutional filters in the Caser model (Tang et al. [130]); (e) Illustration of the union-level pattern extraction ability of the horizontal convolutional filters in the Caser model (Tang et al. [130]).

tion as input to predict rating scores. The reward of the agents is calculated by measuring how similar the agent-predicted scores are to the recommendation model-predicted scores and the presentation quality (e.g., readability and consistency) of the generated explanation. By taking the textual sentences as interpretable components, the authors then adopt a personalized attention-based neural network as an instantiation of the proposed framework and show that it can well explain the recommendations via sentence-level explanations. Lin et al. [125] integrates the rating score prediction task and the explainable word generation task into a unified neural network. In this model, neural collaborative filtering (NCF) [148] is applied to the user-POI rating matrix to predict the rating scores. The reviews are transformed into syntax relations by utilizing the spaCy CNN dependency parsing model [149, 150], which are further organized into pairs ¡opinion, aspect¿. The learned user embeddings from NCF are then clustered based on cosine similarity. The textual explanations are then extracted from the pre-processed word pairs in the top-K users' reviews.

4.7. Summary

In this section, we reviewed deep learning-based explainable approaches for trust-aware recommendation. After introducing the general techniques
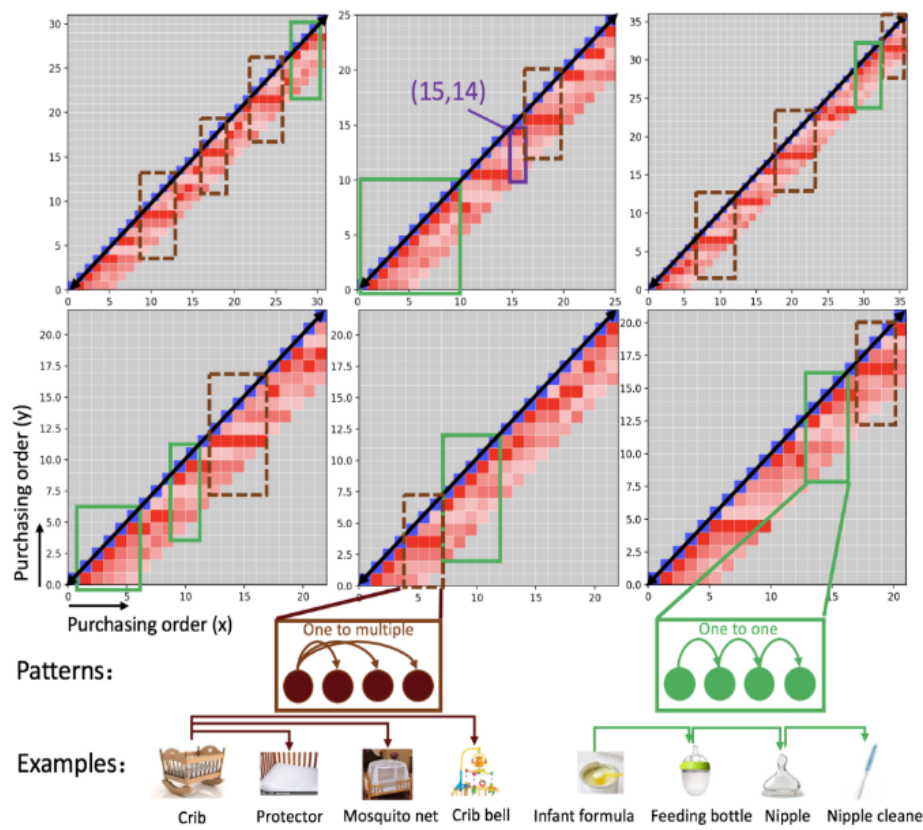
42

Figure 16: Illustration of sequential item-to-item transitions using MANN (Chen et al. [131]).

for explainable recommendation, we focused on deep learning methods that leverage collaborative filtering, textual data, visual data, and temporal data. Generally, deep learning aims to explain the recommendation results from the mechanism of how the recommendation process works. Most models leverage textual reviews, item images, and temporal information of the user-item interactions to deal with limited user-item ratings. Considerable work combine attention mechanism with deep neural networks (e.g., RNNs, CNNs and memory neural networks) and generate explanations from auxiliary data (i.e., reviews, images, sequential patterns). Another category of models adopts generative methods (e.g., RNN and GAN), to provide novel textual/image explanations. Other models include traditional content-based similarity methods, hybrid methods that exploit both texts and images, and reinforcement

learning that controls the quality of explanations. Overall, deep learning has demonstrated as a promising approach to explainability for trustworthy recommendation.

5. Discussion

Though effective, current researches face the challenges such as relying on sufficient labels, requiring manually tuning, and inflexibility for multi-tasks. We discuss potential solutions to some of the issues as follows.

5.1. Dynamic Trust in Recommender Systems

In the real world cases, the trust information is changing over time. For example, trust friend relationships may change; we may have a different group of friends during different periods. Also, the preference for friendship may change over time[117]. Another example is for malicious users. We know that the malicious reviews or ratings may affect common users, but not all the reviews written by malicious reviewers are for sure the fake reviews[11]. Some fake reviewers may mystify the detection system by writing some common reviews but write fake reviews to the target products. Most work assumes that the reviews/ratings made by malicious users are fake samples; however, limited work considers solving the problem case by case because it is hard to get all the labels. Another challenge is that the dynamic system may be time-consuming for updating the whole system over time.

5.2. Embedding for Recommender System

Embedding methods, which include node embedding, sequential embedding, and graph embedding, are widely applied into recommender systems. For example, several graph embedding methods learn social relationship as a type of graph information. Such embedding methods include node2vec[32], Euclidean embedding[151], UniWalk (explainable) [152], deepWalk[71], and the recent model, graph neural networks[153]. Effective embedding has achieved significant improvement in recommender systems, including various application domains[154]. The learned embedding vector is normally used for representing users/items or further combined with other representations. Some may construct the graph for items to model the user behaviors: each item is represented by a node in the graph, and the co-occurrence of items is denoted by edge. Then, the graph embedding methods are applied to learn the embedding[154]. Some may represent the user by their trust

relationships, and then combine such representation with user rating behaviors for recommendation [7, 27]. Most related work has the limitation of ignoring the inner interactions between different types of information. Intuitively, a user's decision is affected by many factors. Friendship, product reviews (especially malicious reviews), and product description can all affect our decisions. Thus, the latent representation for the user should be better not simply the concatenation of representations in different domains, but a unified factor.

5.3. Deep Meta-learning for Recommender Systems

For recommendation problem, there is no standard base model for dealing with different tasks. Like the aforementioned three trust-aware tasks, limited work combines different ideas, i.e., cover all the bases, for recommendation problems. Applying meta-learning or learning to learn may cover this limitation. Meta-learning, or learning to learn, is the science of observing the performance of different model configurations on various tasks, and then learning from the history observation, to provide guidance to the new tasks. This will not only improve the efficiency of the new task modeling but also enable the model with automatically learning capabilities. This is an inspiring area since most current works are based on hand-engineered way[155]. Most deep models only perform well on one task or a single dataset. This means we may cost a lot of manual efforts on designing the models instead of solving the problems. Thus, it is meaningful to design a module to let the machine learns itself with supporting the related information.

5.4. Blockchain for Decentralized Trust Management

Current recommender systems are built upon the data from web users – which contain both normal users and malicious users – and thus become vulnerable to the real world frauds. For example, with the increasing number of fraudulent ratings or feedbacks, the truth for a recommender system will deviate from the genuine truth. A major reason for fraudulent behaviors is the easiness of getting publicly available information. With learning from normal users, fraudulent users can hide their intention and cheat the detection techniques, which will affect the robustness of recommender systems. One potential solution is leveraging the blockchain idea for trust management in recommender systems. Blockchain[156, 157] is a shared ledger technology, and each participant shares a common view of truth. It uses a decentralized peer-to-peer network to manage the data, which can eliminate the potential

risks of centrally stored data, and all validated activities are permanently recorded. Each participant can get access to their own data, and even a system administrator cannot delete the records. This secures each transaction and thus eliminates the human error or fraud. Besides, most of the deep learning-based recommendation techniques are data-hungry and are centralized in computing. A recent idea is distributing the learning tasks of a deep learning method on blockchain, which can improve both efficiency and privacy [158].

6. Conclusion

In this survey, we investigate three aspects of trust in recommender systems: social-awareness, robustness, and explainability, with a focus on deep learning-based on recommender systems. We describe how deep learning methods work for the trust-aware recommendation in representation learning, predictive learning, and generative learning. We notice that the growing research in deep learning have brought about a significant improvement in the performance of recommender systems in various tasks. Meanwhile, current research still faces severe challenges in adapting to labeled data, reducing tuning efforts, and enhance the flexibility in handling multiple tasks. We hope this survey could give readers a comprehensive understanding of state-of-the-art studies in deep learning-based recommendation and inspire more insights and contributions to this vibrant research domain.

References

[1] S. Zhang, L. Yao, A. Sun, Y. Tay, Deep learning based recommender system: A survey and new perspectives, ACM Computing Surveys (CSUR) 52 (1) (2019) 5.

[2] G. Guo, J. Zhang, N. Yorke-Smith, Trustsvd: collaborative filtering with both the explicit and implicit influence of user trust and of item ratings, in: Twenty-Ninth AAAI Conference on Artificial Intelligence, AAAI, 2015.

[3] Y. Pan, F. He, H. Yu, Trust-aware collaborative denoising auto-encoder for top-n recommendation, arXiv preprint arXiv:1703.01760 (2017).

[4] B. Yang, Y. Lei, J. Liu, W. Li, Social collaborative filtering by trust, IEEE transactions on pattern analysis and machine intelligence 39 (8) (2017) 1633–1647.

[5] X. Ma, H. Lu, Z. Gan, J. Zeng, An explicit trust and distrust clustering based collaborative filtering recommendation approach, Electronic Commerce Research and Applications 25 (2017) 29–39.

[6] G. Fei, A. Mukherjee, B. Liu, M. Hsu, M. Castellanos, R. Ghosh, Exploiting burstiness in reviews for review spammer detection, in: Seventh international AAAI conference on weblogs and social media, 2013.

[7] L. Wu, P. Sun, R. Hong, Y. Fu, X. Wang, M. Wang, Socialgcn: An efficient graph convolutional network based model for social recommendation, arXiv preprint arXiv:1811.02815 (2018).

[8] J. Zou, F. Fekri, A belief propagation approach for detecting shilling attacks in collaborative filtering, in: Proceedings of the 22nd ACM international conference on Information & Knowledge Management, ACM, 2013, pp. 1837–1840.

[9] Z. Yang, L. Xu, Z. Cai, Z. Xu, Re-scale adaboost for attack detection in collaborative filtering recommender systems, Knowledge-Based Systems 100 (2016) 74–88.

[10] R. Bhaumik, B. Mobasher, R. Burke, A clustering approach to unsupervised attack detection in collaborative recommender systems, in: Proceedings of the International Conference on Data Mining, Citeseer, 2011, p. 1.

[11] M. Si, Q. Li, Shilling attacks against collaborative recommender systems: a review, Artificial Intelligence Review (2018) 1–29.

[12] X. Dong, L. Yu, Z. Wu, Y. Sun, L. Yuan, F. Zhang, A hybrid collaborative filtering model with deep structure for recommender systems, in: Thirty-First AAAI Conference on Artificial Intelligence, 2017.

[13] Y. Zhang, X. Chen, Explainable recommendation: A survey and new perspectives, arXiv preprint arXiv:1804.11192 (2018).

[14] W. X. Zhao, S. Li, Y. He, L. Wang, J.-R. Wen, X. Li, Exploring demographic information in social media for product recommendation, Knowledge and Information Systems 49 (1) (2016) 61–89.

[15] Y. Zhang, G. Lai, M. Zhang, Y. Zhang, Y. Liu, S. Ma, Explicit factor models for explainable recommendation based on phrase-level sentiment analysis, in: Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval, ACM, 2014, pp. 83–92.

[16] Y. Lin, P. Ren, Z. Chen, Z. Ren, J. Ma, M. De Rijke, Explainable outfit recommendation with joint outfit matching and comment generation, IEEE Transactions on Knowledge and Data Engineering (2019).

[17] B. Wang, M. Ester, J. Bu, D. Cai, Who also likes it? generating the most persuasive social explanations in recommender systems, in: Twenty-Eighth AAAI Conference on Artificial Intelligence, 2014.

[18] X. Chen, Z. Qin, Y. Zhang, T. Xu, Learning to rank features for recommendation over multiple categories, in: Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, ACM, 2016, pp. 305–314.

[19] R. Heckel, M. Vlachos, T. Parnell, C. Dünner, Scalable and interpretable product recommendations via overlapping co-clustering, in: 2017 IEEE 33rd International Conference on Data Engineering (ICDE), IEEE, 2017, pp. 1033–1044.

[20] Z. Ren, S. Liang, P. Li, S. Wang, M. de Rijke, Social collaborative viewpoint regression with explainable recommendations, in: Proceedings of the tenth ACM international conference on web search and data mining, ACM, 2017, pp. 485–494.

[21] S. Seo, J. Huang, H. Yang, Y. Liu, Interpretable convolutional neural networks with dual local and global attention for review rating prediction, in: Proceedings of the Eleventh ACM Conference on Recommender Systems, ACM, 2017, pp. 297–305.

[22] J. Davidson, B. Liebald, J. Liu, P. Nandy, T. Van Vleet, U. Gargi, S. Gupta, Y. He, M. Lambert, B. Livingston, et al., The youtube video

recommendation system, in: Proceedings of the fourth ACM conference on Recommender systems, ACM, 2010, pp. 293–296.

[23] F. Monti, M. Bronstein, X. Bresson, Geometric matrix completion with recurrent multi-graph neural networks, in: Advances in Neural Information Processing Systems, 2017, pp. 3697–3707.

[24] X. Yang, Y. Guo, Y. Liu, H. Steck, A survey of collaborative filtering based social recommender systems, Computer Communications 41 (2014) 1–10.

[25] I. Gunes, C. Kaleli, A. Bilge, H. Polat, Shilling attacks against recommender systems: a comprehensive survey, Artificial Intelligence Review 42 (4) (2014) 767–799.

[26] W. Fan, Y. Ma, Q. Li, Y. He, E. Zhao, J. Tang, D. Yin, Graph neural networks for social recommendation, arXiv preprint arXiv:1902.07243 (2019).

[27] Q. Wu, H. Zhang, X. Gao, P. He, P. Weng, H. Gao, G. Chen, Dual graph attention networks for deep latent representation of multifaceted social effects in recommender systems, arXiv preprint arXiv:1903.10433 (2019).

[28] M. Jamali, M. Ester, Using a trust network to improve top-n recommendation, in: Proceedings of the third ACM conference on Recommender systems, ACM, 2009, pp. 181–188.

[29] M. Jamali, M. Ester, Trustwalker: a random walk model for combining trust-based and item-based recommendation, in: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2009, pp. 397–406.

[30] C. Zhang, L. Yu, Y. Wang, C. Shah, X. Zhang, Collaborative user network embedding for social recommender systems, in: 17th SIAM International Conference on Data Mining, SDM 2017, Society for Industrial and Applied Mathematics Publications, 2017, pp. 381–389.

[31] Y. Wen, L. Guo, Z. Chen, J. Ma, Network embedding based recommendation method in social networks, in: Companion of the The Web

Conference 2018 on The Web Conference 2018, International World Wide Web Conferences Steering Committee, 2018, pp. 11–12.

[32] A. Grover, J. Leskovec, node2vec: Scalable feature learning for networks, in: Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2016, pp. 855–864.

[33] Z. Zhao, Q. Yang, H. Lu, T. Weninger, D. Cai, X. He, Y. Zhuang, Social-aware movie recommendation via multimodal network learning, IEEE Transactions on Multimedia 20 (2) (2017) 430–440.

[34] Y. Koren, Factorization meets the neighborhood: a multifaceted collaborative filtering model, in: Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2008, pp. 426–434.

[35] K. Ahn, K. Lee, H. Cha, C. Suh, Binary rating estimation with graph side information, in: Advances in Neural Information Processing Systems, 2018, pp. 4272–4283.

[36] H. Zhang, G. Liu, J. Wu, Social collaborative filtering ensemble, in: Pacific Rim International Conference on Artificial Intelligence, Springer, 2018, pp. 1005–1017.

[37] C. Nisha, A. Mohan, A social recommender system using deep architecture and network embedding, Applied Intelligence 49 (5) (2019) 1937–1953.

[38] M. Wang, Z. Wu, X. Sun, G. Feng, B. Zhang, Trust-aware collaborative filtering with a denoising autoencoder, Neural Processing Letters 49 (2) (2019) 835–849.

[39] L. Wu, P. Sun, R. Hong, Y. Ge, M. Wang, Collaborative neural social recommendation, IEEE Transactions on Systems, Man, and Cybernetics: Systems (2018).

[40] S. Deng, L. Huang, G. Xu, X. Wu, Z. Wu, On deep learning for trust-aware recommendations in social networks, IEEE transactions on neural networks and learning systems 28 (5) (2016) 1164–1177.

[41] D. Rafailidis, F. Crestani, Recommendation with social relationships via deep learning, in: Proceedings of the ACM SIGIR International Conference on Theory of Information Retrieval, ACM, 2017, pp. 151–158.

[42] P. Sun, L. Wu, M. Wang, Attentive recurrent social recommendation, in: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, ACM, 2018, pp. 185–194.

[43] W. Song, Z. Xiao, Y. Wang, L. Charlin, M. Zhang, J. Tang, Session-based social recommendation via dynamic graph attention networks, in: Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, ACM, 2019, pp. 555–563.

[44] L. Wu, H.-F. Yu, N. Rao, J. Sharpnack, C.-J. Hsieh, Graph dna: Deep neighborhood aware graph encoding for collaborative filtering, arXiv preprint arXiv:1905.12217 (2019).

[45] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, J. Leskovec, Graph convolutional neural networks for web-scale recommender systems, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, ACM, 2018, pp. 974–983.

[46] W. Fan, T. Derr, Y. Ma, J. Wang, J. Tang, Q. Li, Deep adversarial social recommendation, arXiv preprint arXiv:1905.13160 (2019).

[47] T. Xiao, H. Tian, H. Shen, Variational deep collaborative matrix factorization for social recommendation, in: Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, 2019, pp. 426–437.

[48] L. Wu, L. Chen, R. Hong, Y. Fu, X. Xie, M. Wang, A hierarchical attention model for social contextual image recommendation, IEEE Transactions on Knowledge and Data Engineering (2019).

[49] H. Bao, L. Wu, P. Sun, Contextual attention model for social recommendation, in: Pacific Rim Conference on Multimedia, Springer, 2018, pp. 630–641.

[50] C. H. Liu, J. Xu, J. Tang, J. Crowcroft, Social-aware sequential modeling of user interests: A deep learning approach, IEEE Transactions on Knowledge and Data Engineering (2018).

[51] J. Gao, T. Zhang, C. Xu, A unified personalized video recommendation via dynamic recurrent neural networks, in: Proceedings of the 25th ACM international conference on Multimedia, ACM, 2017, pp. 127–135.

[52] X. Geng, H. Zhang, J. Bian, T.-S. Chua, Learning image and user features for recommendation in social networks, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 4274–4282.

[53] S. Liu, P. Cui, W. Zhu, S. Yang, Learning socially embedded visual representation from scratch, in: Proceedings of the 23rd ACM international conference on Multimedia, ACM, 2015, pp. 109–118.

[54] C. Chen, M. Zhang, Y. Liu, S. Ma, Social attentional memory network: Modeling aspect-and friend-level differences in recommendation, in: Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, ACM, 2019, pp. 177–185.

[55] D. Rafailidis, G. Weiss, A neural attention model for adaptive learning of social friends' preferences, arXiv preprint arXiv:1907.01644 (2019).

[56] L. Xiao, Z. Min, L. Yiqun, M. Shaoping, A neural network model for social-aware recommendation, in: Asia Information Retrieval Symposium, Springer, 2017, pp. 125–137.

[57] W. Fan, Q. Li, M. Cheng, Deep modeling of social relations for recommendation, in: Thirty-Second AAAI Conference on Artificial Intelligence, 2018.

[58] X. Wang, X. He, L. Nie, T.-S. Chua, Item silk road: Recommending items from information domains to social users, in: Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval, ACM, 2017, pp. 185–194.

[59] C.-Y. Liu, C. Zhou, J. Wu, Y. Hu, L. Guo, Social recommendation with an essential preference space, in: Thirty-Second AAAI Conference on Artificial Intelligence, 2018.

[60] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.-A. Manzagol, Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion, Journal of machine learning research 11 (Dec) (2010) 3371–3408.

[61] S. Lai, L. Xu, K. Liu, J. Zhao, Recurrent convolutional neural networks for text classification, in: Twenty-ninth AAAI conference on artificial intelligence, 2015.

[62] P. H. Pinheiro, R. Collobert, Recurrent convolutional neural networks for scene labeling, in: 31st International Conference on Machine Learning (ICML), no. CONF, 2014.

[63] S. Wu, W. Ren, C. Yu, G. Chen, D. Zhang, J. Zhu, Personal recommendation using deep recurrent neural networks in netease, in: 2016 IEEE 32nd International Conference on Data Engineering (ICDE), IEEE, 2016, pp. 1218–1229.

[64] T. Donkers, B. Loepp, J. Ziegler, Sequential user-based recurrent neural network recommendations, in: Proceedings of the Eleventh ACM Conference on Recommender Systems, ACM, 2017, pp. 152–160.

[65] C.-Y. Wu, A. Ahmed, A. Beutel, A. J. Smola, H. Jing, Recurrent recommender networks, in: Proceedings of the tenth ACM international conference on web search and data mining, ACM, 2017, pp. 495–503.

[66] C.-Y. Wu, A. Ahmed, A. Beutel, A. J. Smola, Joint training of ratings and reviews with recurrent recommender networks (2016).

[67] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Advances in neural information processing systems, 2014, pp. 2672–2680.

[68] D. P. Kingma, M. Welling, Auto-encoding variational bayes, arXiv preprint arXiv:1312.6114 (2013).

[69] Z. Wang, M. Gao, X. Wang, J. Yu, J. Wen, Q. Xiong, A minimax game for generative and discriminative sample models for recommendation, in: Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, 2019, pp. 420–431.

[70] G. Karamanolakis, K. R. Cherian, A. R. Narayan, J. Yuan, D. Tang, T. Jebara, Item recommendation with variational autoencoders and heterogeneous priors, in: Proceedings of the 3rd Workshop on Deep Learning for Recommender Systems, ACM, 2018, pp. 10–14.

[71] B. Perozzi, R. Al-Rfou, S. Skiena, Deepwalk: Online learning of social representations, in: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2014, pp. 701–710.

[72] M.-T. Luong, H. Pham, C. D. Manning, Effective approaches to attention-based neural machine translation, arXiv preprint arXiv:1508.04025 (2015).

[73] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, Y. Bengio, Show, attend and tell: Neural image caption generation with visual attention, in: International conference on machine learning, 2015, pp. 2048–2057.

[74] A. Mnih, R. R. Salakhutdinov, Probabilistic matrix factorization, in: Advances in neural information processing systems, 2008, pp. 1257–1264.

[75] A. M. Elkahky, Y. Song, X. He, A multi-view deep learning approach for cross domain user modeling in recommendation systems, in: Proceedings of the 24th International Conference on World Wide Web, International World Wide Web Conferences Steering Committee, 2015, pp. 278–288.

[76] G. Guo, J. Zhang, Z. Sun, N. Yorke-Smith, Librec: A java library for recommender systems., in: UMAP Workshops, Vol. 4, 2015.

[77] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, nature 521 (7553) (2015) 436.

[78] S. Alonso, J. Bobadilla, F. Ortega, R. Moya, Robust model-based reliability approach to tackle shilling attacks in collaborative filtering recommender systems, IEEE Access 7 (2019) 41782–41798.

[79] F. Zhang, Y. Lu, J. Chen, S. Liu, Z. Ling, Robust collaborative filtering based on non-negative matrix factorization and r1-norm, Knowledge-based systems 118 (2017) 177–190.

[80] H. Yu, R. Gao, K. Wang, F. Zhang, A novel robust recommendation method based on kernel matrix factorization, Journal of Intelligent & Fuzzy Systems 32 (3) (2017) 2101–2109.

[81] X. He, Z. He, X. Du, T.-S. Chua, Adversarial personalized ranking for recommendation, in: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, ACM, 2018, pp. 355–364.

[82] R. Chalapathy, S. Chawla, Deep learning for anomaly detection: A survey, arXiv preprint arXiv:1901.03407 (2019).

[83] Y. Ren, D. Ji, Learning to detect deceptive opinion spam: A survey, IEEE Access 7 (2019) 42934–42945.

[84] S. Zhao, Z. Xu, L. Liu, M. Guo, Towards accurate deceptive opinion spam detection based on word order-preserving cnn, arXiv preprint arXiv:1711.09181 (2017).

[85] W. Zhang, Y. Du, T. Yoshida, Q. Wang, Dri-rcnn: An approach to deceptive review identification using recurrent convolutional neural network, Information Processing & Management 54 (4) (2018) 576–592.

[86] L. Li, W. Ren, B. Qin, T. Liu, Learning document representation for deceptive opinion spam detection, in: Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data, Springer, 2015, pp. 393–404.

[87] X. Wang, K. Liu, J. Zhao, Detecting deceptive review spam via attention-based neural networks, in: National CCF Conference on Natural Language Processing and Chinese Computing, Springer, 2017, pp. 866–876.

[88] X. Wang, K. Liu, J. Zhao, Handling cold-start problem in review spam detection by jointly embedding texts and behaviors, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2017, pp. 366–376.

[89] J. Chung, C. Gulcehre, K. Cho, Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling, arXiv preprint arXiv:1412.3555 (2014).

[90] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural computation 9 (8) (1997) 1735–1780.

[91] Y. Ren, Y. Zhang, Deceptive opinion spam detection using neural network, in: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, 2016, pp. 140–150.

[92] C.-C. Wang, M.-Y. Day, C.-C. Chen, J.-W. Liou, Detecting spamming reviews using long short-term memory recurrent neural network framework, in: Proceedings of the 2nd International Conference on E-commerce, E-Business and E-Government, ACM, 2018, pp. 16–20.

[93] X. Wang, K. Liu, S. He, J. Zhao, Learning to represent review with tensor decomposition for spam detection, in: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, 2016, pp. 866–875.

[94] M. Dong, L. Yao, X. Wang, B. Benatallah, C. Huang, X. Ning, Opinion fraud detection via neural autoencoder decision forest, Pattern Recognition Letters (2018).

[95] H. Aghakhani, A. Machiry, S. Nilizadeh, C. Kruegel, G. Vigna, Detecting deceptive reviews using generative adversarial networks, in: 2018 IEEE Security and Privacy Workshops (SPW), IEEE, 2018, pp. 89–95.

[96] Y. Wu, C. DuBois, A. X. Zheng, M. Ester, Collaborative denoising auto-encoders for top-n recommender systems, in: Proceedings of the Ninth ACM International Conference on Web Search and Data Mining, ACM, 2016, pp. 153–162.

[97] F. Strub, R. Gaudel, J. Mary, Hybrid recommender system based on autoencoders, in: Proceedings of the 1st Workshop on Deep Learning for Recommender Systems, ACM, 2016, pp. 11–16.

[98] H. Wang, S. Xingjian, D.-Y. Yeung, Collaborative recurrent autoencoder: Recommend while learning to fill in the blanks, in: Advances in Neural Information Processing Systems, 2016, pp. 415–423.

[99] F. Yuan, L. Yao, B. Benatallah, Adversarial collaborative neural network for robust recommendation (2019).

[100] Q. Wang, H. Yin, Z. Hu, D. Lian, H. Wang, Z. Huang, Neural memory streaming recommender networks with adversarial training, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, ACM, 2018, pp. 2467–2475.

[101] Y. Jhamb, T. Ebesu, Y. Fang, Attentive contextual denoising autoencoder for recommendation, in: Proceedings of the 2018 ACM SIGIR International Conference on Theory of Information Retrieval, ACM, 2018, pp. 27–34.

[102] C. Zhou, J. Bai, J. Song, X. Liu, Z. Zhao, X. Chen, J. Gao, Atrank: An attention-based user behavior modeling framework for recommendation, in: Thirty-Second AAAI Conference on Artificial Intelligence, 2018.

[103] P. Loyola, C. Liu, Y. Hirate, Modeling user session and intent with an attention-based encoder-decoder architecture, in: Proceedings of the Eleventh ACM Conference on Recommender Systems, ACM, 2017, pp. 147–151.

[104] H. Ying, F. Zhuang, F. Zhang, Y. Liu, G. Xu, X. Xie, H. Xiong, J. Wu, Sequential recommender system based on hierarchical attention networks, in: the 27th International Joint Conference on Artificial Intelligence, 2018.

[105] Q. Liu, Y. Zeng, R. Mokhosi, H. Zhang, Stamp: short-term attention/memory priority model for session-based recommendation, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, ACM, 2018, pp. 1831–1839.

[106] S. Seo, J. Huang, H. Yang, Y. Liu, Representation learning of users and items for review rating prediction using attention-based convolutional neural network, in: 3rd international workshop on machine learning methods for recommender systems (MLRec)(SDM'17), 2017.

[107] Y. Tay, L. Anh Tuan, S. C. Hui, Latent relational metric learning via memory-based attention for collaborative ranking, in: Proceedings of the 2018 World Wide Web Conference, International World Wide Web Conferences Steering Committee, 2018, pp. 729–739.

[108] J. Chen, H. Zhang, X. He, L. Nie, W. Liu, T.-S. Chua, Attentive collaborative filtering: Multimedia recommendation with item-and component-level attention, in: Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval, ACM, 2017, pp. 335–344.

[109] N. Tintarev, J. Masthoff, Designing and evaluating explanations for recommender systems, in: Recommender systems handbook, Springer, 2011, pp. 479–510.

[110] A. Sharma, D. Cosley, Do social explanations work?: studying and modeling the effects of social explanations in recommender systems, in: Proceedings of the 22nd international conference on World Wide Web, ACM, 2013, pp. 1133–1144.

[111] C. Chen, M. Zhang, Y. Liu, S. Ma, Neural attentional rating regression with review-level explanations, in: Proceedings of the 2018 World Wide Web Conference on World Wide Web, International World Wide Web Conferences Steering Committee, 2018, pp. 1583–1592.

[112] X. Chen, Y. Zhang, H. Xu, Y. Cao, Z. Qin, H. Zha, Visually explainable recommendation, arXiv preprint arXiv:1801.10288 (2018).

[113] X. Wang, Y. Chen, J. Yang, L. Wu, Z. Wu, X. Xie, A reinforcement learning framework for explainable recommendation, in: 2018 IEEE International Conference on Data Mining (ICDM), IEEE, 2018, pp. 587–596.

[114] J. McInerney, B. Lacker, S. Hansen, K. Higley, H. Bouchard, A. Gruson, R. Mehrotra, Explore, exploit, and explain: personalizing explain-

able recommendations with bandits, in: Proceedings of the 12th ACM Conference on Recommender Systems, ACM, 2018, pp. 31–39.

[115] B. Abdollahi, O. Nasraoui, Explainable restricted boltzmann machines for collaborative filtering, arXiv preprint arXiv:1606.07129 (2016).

[116] L. Wu, C. Quan, C. Li, Q. Wang, B. Zheng, X. Luo, A context-aware user-item representation learning for item recommendation, ACM Transactions on Information Systems (TOIS) 37 (2) (2019) 22.

[117] X. Chen, Y. Zhang, Z. Qin, Dynamic explainable recommendation based on neural attentive models, in: Proceedings of the 33rd AAAI Conference on Artificial Intelligence, 2019.

[118] D. Cong, Y. Zhao, B. Qin, Y. Han, M. Zhang, A. Liu, N. Chen, Hierarchical attention based neural network for explainable recommendation, in: Proceedings of the 2019 on International Conference on Multimedia Retrieval, ACM, 2019, pp. 373–381.

[119] F. Costa, S. Ouyang, P. Dolog, A. Lawlor, Automatic generation of natural language explanations, in: Proceedings of the 23rd International Conference on Intelligent User Interfaces Companion, ACM, 2018, p. 57.

[120] P. Li, Z. Wang, Z. Ren, L. Bing, W. Lam, Neural rating regression with abstractive tips generation for recommendation, in: Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval, ACM, 2017, pp. 345–354.

[121] Y. Lu, R. Dong, B. Smyth, Why i like it: multi-task learning for recommendation and explanation, in: Proceedings of the 12th ACM Conference on Recommender Systems, ACM, 2018, pp. 4–12.

[122] S. Ouyang, A. Lawlor, F. Costa, P. Dolog, Improving explainable recommendations with synthetic reviews, arXiv preprint arXiv:1807.06978 (2018).

[123] G. Zhao, H. Fu, R. Song, T. Sakai, X. Xie, X. Qian, Why you should listen to this song: Reason generation for explainable recommendation, in: 2018 IEEE International Conference on Data Mining Workshops (ICDMW), IEEE, 2018, pp. 1316–1322.

[124] T. Suzuki, S. Oyama, M. Kurihara, Toward explainable recommendations: Generating review text from multicriteria evaluation data, in: 2018 IEEE International Conference on Big Data (Big Data), IEEE, 2018, pp. 3549–3551.

[125] J. Lin, Y. Liu, A neural network based explainable recommender system, arXiv preprint arXiv:1812.11740 (2018).

[126] M. Hou, L. Wu, E. Chen, Z. Li, V. W. Zheng, Q. Liu, Explainable fashion recommendation: A semantic attribute region guided approach, arXiv preprint arXiv:1905.12862 (2019).

[127] W.-C. Kang, C. Fang, Z. Wang, J. McAuley, Visually-aware fashion recommendation and design with generative image models, in: 2017 IEEE International Conference on Data Mining (ICDM), IEEE, 2017, pp. 207–216.

[128] S. Kumar, M. D. Gupta, cgan: Complementary fashion item recommendation, arXiv preprint arXiv:1906.05596 (2019).

[129] H. Bharadhwaj, Layer-wise relevance propagation for explainable recommendations, arXiv preprint arXiv:1807.06160 (2018).

[130] J. Tang, K. Wang, Personalized top-n sequential recommendation via convolutional sequence embedding, in: Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, ACM, 2018, pp. 565–573.

[131] X. Chen, H. Xu, Y. Zhang, J. Tang, Y. Cao, Z. Qin, H. Zha, Sequential recommendation with user memory networks, in: Proceedings of the eleventh ACM international conference on web search and data mining, ACM, 2018, pp. 108–116.

[132] L. Zheng, V. Noroozi, P. S. Yu, Joint deep modeling of users and items using reviews for recommendation, in: Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, ACM, 2017, pp. 425–434.

[133] T. Mikolov, M. Karafiát, L. Burget, J. Černocký, S. Khudanpur, Recurrent neural network based language model, in: Eleventh annual conference of the international speech communication association, 2010.

[134] Y. Hu, Y. Koren, C. Volinsky, Collaborative filtering for implicit feedback datasets, in: 2008 Eighth IEEE International Conference on Data Mining, Ieee, 2008, pp. 263–272.

[135] R. He, J. McAuley, Vbpr: visual bayesian personalized ranking from implicit feedback, in: Thirtieth AAAI Conference on Artificial Intelligence, 2016.

[136] Q. Liu, S. Wu, L. Wang, Deepstyle: Learning user preferences for visual recommendation, in: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2017, pp. 841–844.

[137] S. Wang, Y. Wang, J. Tang, K. Shu, S. Ranganath, H. Liu, What your images reveal: Exploiting visual contents for point-of-interest recommendation, in: Proceedings of the 26th International Conference on World Wide Web, International World Wide Web Conferences Steering Committee, 2017, pp. 391–400.

[138] W. Yu, H. Zhang, X. He, X. Chen, L. Xiong, Z. Qin, Aesthetic-based clothing recommendation, in: Proceedings of the 2018 World Wide Web Conference on World Wide Web, International World Wide Web Conferences Steering Committee, 2018, pp. 649–658.

[139] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[140] S. Rendle, C. Freudenthaler, Z. Gantner, L. Schmidt-Thieme, Bpr: Bayesian personalized ranking from implicit feedback, in: Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence, AUAI Press, 2009, pp. 452–461.

[141] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, Grad-cam: Visual explanations from deep networks via gradient-based localization, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 618–626.

[142] R. Hadsell, S. Chopra, Y. LeCun, Dimensionality reduction by learning an invariant mapping, in: 2006 IEEE Computer Society Conference on

Computer Vision and Pattern Recognition (CVPR'06), Vol. 2, IEEE, 2006, pp. 1735–1742.

[143] M. Mirza, S. Osindero, Conditional generative adversarial nets, arXiv preprint arXiv:1411.1784 (2014).

[144] M. S. Sajjadi, G. Parascandolo, A. Mehrjou, B. Schölkopf, Tempered adversarial networks, arXiv preprint arXiv:1802.04374 (2018).

[145] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, W. Samek, On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation, PloS one 10 (7) (2015) e0130140.

[146] L. Gatys, A. S. Ecker, M. Bethge, Texture synthesis using convolutional neural networks, in: Advances in neural information processing systems, 2015, pp. 262–270.

[147] J. Vinagre, A. M. Jorge, J. Gama, An overview on the exploitation of time in collaborative filtering, Wiley interdisciplinary reviews: Data mining and knowledge discovery 5 (5) (2015) 195–215.

[148] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, T.-S. Chua, Neural collaborative filtering, in: Proceedings of the 26th International Conference on World Wide Web, International World Wide Web Conferences Steering Committee, 2017, pp. 173–182.

[149] E. Kiperwasser, Y. Goldberg, Simple and accurate dependency parsing using bidirectional lstm feature representations, Transactions of the Association for Computational Linguistics 4 (2016) 313–327.

[150] Y. Goldberg, J. Nivre, A dynamic oracle for arc-eager dependency parsing, Proceedings of COLING 2012 (2012) 959–976.

[151] W. Li, M. Gao, W. Rong, J. Wen, Q. Xiong, R. Jia, T. Dou, Social recommendation using euclidean embedding, in: 2017 International Joint Conference on Neural Networks (IJCNN), IEEE, 2017, pp. 589–595.

[152] H. Park, H. Jeon, J. Kim, B. Ahn, U. Kang, Uniwalk: Explainable and accurate recommendation for rating and network data, arXiv preprint arXiv:1710.07134 (2017).

[153] M. Defferrard, X. Bresson, P. Vandergheynst, Convolutional neural networks on graphs with fast localized spectral filtering, in: Advances in neural information processing systems, 2016, pp. 3844–3852.

[154] Y. Ouyang, B. Guo, X. Tang, X. He, J. Xiong, Z. Yu, Learning cross-domain representation with multi-graph neural network, arXiv preprint arXiv:1905.10095 (2019).

[155] J. Vanschoren, Meta-learning: A survey, arXiv preprint arXiv:1810.03548 (2018).

[156] L. Yao, X. Wang, Q. Z. Sheng, S. Dustdar, S. Zhang, Recommendations on the internet of things: Requirements, challenges, and directions, IEEE Internet Computing 23 (3) (2019) 46–54.

[157] M. Crosby, P. Pattanayak, S. Verma, V. Kalyanaraman, et al., Blockchain technology: Beyond bitcoin, Applied Innovation 2 (6-10) (2016) 71.

[158] J.-S. Weng, J. Weng, M. Li, Y. Zhang, W. Luo, Deepchain: Auditable and privacy-preserving deep learning with blockchain-based incentive., IACR Cryptology ePrint Archive 2018 (2018) 679.