Deep Learning Based Recommender System: A Survey and New Perspectives

SHUAI ZHANG and LINA YAO, University of New South Wales
AIXIN SUN and YI TAY, Nanyang Technological University

With the growing volume of online information, recommender systems have been an effective strategy to overcome information overload. The utility of recommender systems cannot be overstated, given their widespread adoption in many web applications, along with their potential impact to ameliorate many problems related to over-choice. In recent years, deep learning has garnered considerable interest in many research fields such as computer vision and natural language processing, owing not only to stellar performance but also to the attractive property of learning feature representations from scratch. The influence of deep learning is also pervasive, recently demonstrating its effectiveness when applied to information retrieval and recommender systems research. The field of deep learning in recommender system is flourishing. This article aims to provide a comprehensive review of recent research efforts on deep learning-based recommender systems. More concretely, we provide and devise a taxonomy of deep learning-based recommendation models, along with a comprehensive summary of the state of the art. Finally, we expand on current trends and provide new perspectives pertaining to this new and exciting development of the field.

CCS Concepts: • **Information systems → Recommender systems**;

Additional Key Words and Phrases: Recommender system, deep learning, survey

ACM Reference format:
Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2019. Deep Learning Based Recommender System: A Survey and New Perspectives. *ACM Comput. Surv.* 52, 1, Article 5 (February 2019), 38 pages.
https://doi.org/10.1145/3285029

1 INTRODUCTION

Recommender systems are an intuitive line of defense against consumer over-choice. Given the explosive growth of information available on the web, users are often greeted with countless products, movies, or restaurants. As such, personalization is an essential strategy for facilitating a better user experience. All in all, these systems have been playing a vital and indispensable role in various information access systems to boost business and facilitate the decision-making process [69, 122] and are pervasive across numerous web domains such as e-commerce and/or media websites.

In general, recommendation lists are generated based on user preferences, item features, user/item past interactions, and some other additional information such as temporal (e.g., sequence-aware recommender [117]) and spatial (e.g., POI) data. Recommendation models are

Authors' addresses: S. Zhang and L. Yao, University of New South Wales, K17, CSE, UNSW, Sydney, Australia; emails: {shuai.zhang, lina.yao}@unsw.edu.au; A. Sun and Y. Tay, Nanyang Technological University, scse, ntu, nanyang avenue, singapore; emails: axsun@ntu.edu.sg, ytay017@e.ntu.edu.sg.

0360-0300/2019/02-ART5 $15.00
https://doi.org/10.1145/3285029

mainly categorized into collaborative filtering, content-based recommender systems, and hybrid recommender systems based on the types of input data [1].

Deep learning enjoys massive hype at the moment. The past few decades have witnessed the tremendous success of the deep learning in many application domains such as computer vision and speech recognition. The academia and industry have been in a race to apply deep learning to a wider range of applications due to its capability in solving many complex tasks while providing start-of-the-art results [27]. Recently, deep learning has been changing recommendation architecture dramatically and bringing more opportunities to improve the performance (e.g., Recall, Precision, etc.) of recommender systems. Recent advances in deep learning-based recommender systems have gained significant attention by overcoming obstacles of conventional models and achieving high recommendation quality. Deep learning is able to effectively capture nonlinear and nontrivial user/item relationships and enable the codification of more complex abstractions as data representations in the higher layers. Furthermore, it catches the intricate relationships within the data itself, from abundant accessible data sources such as contextual, textual, and visual information.

Pervasiveness and ubiquity of deep learning in recommender systems. In industry, recommender systems are critical tools to enhance user experience and promote sales/services for many online websites and mobile applications [20, 27, 30, 43, 113]. For example, 80% of movies watched on Netflix came from recommendations [43], and 60% of video clicks came from home page recommendation in YouTube [30]. Recently, many companies employ deep learning for further enhancing their recommendation quality [20, 27, 113]. Covington et al. [27] presented a deep neural network-based recommendation algorithm for video recommendation on YouTube. Cheng et al. [20] proposed an App recommender system for Google Play with a wide and deep model. Shumpei et al. [113] presented an RNN-based news recommender system for Yahoo! News. All of these models have stood online testing and shown significant improvement over traditional models. Thus, we can see that deep learning has driven a remarkable revolution in industrial recommender applications.

The number of research publications on deep learning-based recommendation methods has increased exponentially recently, providing strong evidence of the inevitable pervasiveness of deep learning in recommender system research. The leading international conference on recommender system, RecSys[1], began organizing regular workshop on deep learning for recommender systems[2] in 2016. This workshop aims to promote research and encourage applications of deep learning-based recommender system.

The success of deep learning for recommendation both in academia and in industry requires a comprehensive review and summary for successive researchers and practitioners to better understand its strength and weakness and the application scenarios of these models.

What are the differences between this survey and former ones? Plenty of research has been done in the field of deep learning-based recommendation. However, to the best of our knowledge, there are very few comprehensive reviews that shape this area and position existing works and current progresses. Although some works have explored recommender applications built on deep learning techniques and have attempted to formalize this research field, few have sought to provide an in-depth summary of current efforts or detail the open problems present in the area. This survey seeks to provide such a comprehensive summary of current research on deep learning-based recommender systems, to identify open problems currently limiting real-world implementations, and to point out future directions along this dimension.

[1] https://recsys.acm.org/.
[2] http://dlrs-workshop.org/.

In the past few years, a number of surveys in traditional recommender systems have been presented. For example, Su et al. [139] presented a comprehensive review on collaborative filtering techniques; Burke et al. [8] proposed a comprehensive survey on hybrid recommender system; Fernández-Tobías et al. [40] and Khan et al. [74] reviewed the cross-domain recommendation models, to name a few. However, there is a lack of extensive reviews on deep learning-based recommender systems. To the extent of our knowledge, only two related short surveys [7, 98] are formally published. Betru et al. [7] introduced three deep learning-based recommendation models [124, 154, 160], and, although these three works are influential in this research area, this survey lost sight of other emerging high-quality contributions. Liu et al. [98] reviewed 13 papers on deep learning for recommendation and proposed classifying these models based on the form of inputs (approaches using content information and approaches without content information) and outputs (rating and ranking). However, with the constant advent of novel research contributions, this classification framework is no longer suitable and a new inclusive framework is required for better understanding this research field. Given the rising popularity and potential of deep learning applied in recommender systems, a comprehensive survey will be of high scientific and practical values. We analyzed existing publications from different perspectives and presented some new insights in this area. To this end, more than 100 studies were shortlisted and classified in this survey.

How do we collect the papers? We used Google Scholar as the main search engine, we also adopted the database Web of Science as an important tool to discover related papers. In addition, we screened most of the related high-profile conferences such as NIPS, ICML, ICLR, KDD, WWW, SIGIR, WSDM, and RecSys, just to name a few, to find out the recent work. The major keywords we used included recommender system, recommendation, deep learning, neural networks, collaborative filtering, and matrix factorization.

Contributions of this survey. The goal of this survey is to thoroughly review literature on the advances of deep learning-based recommender system. It provides a panorama through which readers can quickly understand and step into the field of deep learning-based recommendation. This survey lays the foundations to foster innovations in the area of recommender systems and tap into the richness of this research area. This survey serves those researchers, practitioners, and educators who are interested in recommender systems, with the hope that they will have a rough guideline when it comes to choosing the deep neural networks to solve recommendation tasks at hand. To summarize, the key contributions of this survey are three-fold: (i) We conduct a comprehensive review for recommendation models based on deep learning techniques and propose a classification scheme to position and organize the current work; (ii) we provide an overview and summary for the state of the art; and (iii) we discuss the challenges and open issues, and identify the new trends and future directions in this research field to share the vision and expand the horizons of deep learning-based recommender systems research.

The remainder of this article is organized as follows: Section 2 introduces the preliminaries for recommender systems and deep neural networks, and we also discuss the advantages and disadvantages of deep neural network-based recommendation models. Section 3 first presents our classification framework and then gives a detailed introduction to the state of the art. Section 4 discusses the challenges and prominent open research issues. Section 5 concludes the paper.

2 OVERVIEW OF RECOMMENDER SYSTEMS AND DEEP LEARNING

Before we dive into the details of this survey, we start with an introduction to the basic terminology and concepts regarding recommender systems and deep learning techniques. We also discuss the reasons and motivations of introducing deep neural networks into recommender systems.

2.1 Recommender Systems

Recommender systems estimate users' preference on items and proactively recommend items that users might like [1, 122]. Recommendation models are usually classified into three categories [1, 69]: collaborative filtering, content-based, and hybrid. Collaborative filtering makes recommendations by learning from user/item historical interactions, either through explicit (e.g., user's previous ratings) or implicit feedback (e.g., browsing history). Content-based recommendation is based primarily on comparisons across items' and users' auxiliary information. A diverse range of auxiliary information such as texts, images, and videos can be taken into account. Hybrid models are recommender systems that integrate two or more types of recommendation strategies [8, 69].

Suppose we have M users and N items, and R denotes the interaction matrix and $\hat{R}$ denotes the predicted interaction matrix. Let r_{ui} denote the preference of user u to item i, and $\hat{r}_{ui}$ denote the predicted score. Meanwhile, we use a partially observed vector (rows of R) $\mathbf{r}^{(u)} = \{r^{u1}, \ldots, r^{uN}\}$ to represent each user u and partially observed vector (columns of R) $\mathbf{r}^{(i)} = \{r^{1i}, \ldots, r^{Mi}\}$ to represent each item i. O and O^- denote the observed and unobserved interaction set. We use $U \in \mathcal{R}^{M \times k}$ and $V \in \mathcal{R}^{N \times k}$ to denote user and item latent factor. k is the dimension of latent factors. In addition, sequence information such as timestamp can also be considered to make sequence-aware recommendations [117]. Other notations and denotations will be introduced in corresponding sections.

2.2 Deep Learning Techniques

Deep learning can be generally considered a subfield of machine learning. The typical defining essence of deep learning is that it learns *deep representations*; that is, learning multiple levels of representations and abstractions from data. For practical reasons, we consider any neural differentiable architecture as *"deep learning"* as long as it optimizes a differentiable objective function using a variant of Stochastic Gradient Descent (SGD). Neural architectures have demonstrated tremendous success in both supervised and unsupervised learning tasks [31]. In this subsection, we clarify a diverse array of architectural paradigms that are closely related to this survey.

- The Multilayer Perceptron (MLP) is a feed-forward neural network with multiple (one or more) hidden layers between the input and output layers. Here, the perceptron can employ an arbitrary activation function and does not necessarily represent a strictly binary classifier. MLPs can be interpreted as stacked layers of nonlinear transformations, learning hierarchical feature representations. MLPs are also known to be universal approximators.
- An Autoencoder (AE) is an unsupervised model attempting to reconstruct its input data in the output layer. In general, the bottleneck layer (the middle-most layer) is used as a salient feature representation of the input data. There are many variants of autoencoders such as the denoising autoencoder, marginalized denoising autoencoder, sparse autoencoder, contractive autoencoder, and Variational Autoencoder (VAE) [15, 45].
- The Convolutional Neural Network (CNN) [45] is a special kind of feedforward neural network with convolution layers and pooling operations. It can capture the global and local features and significantly enhances efficiency and accuracy. It performs well in processing data with grid-like topology.
- The Recurrent Neural Network (RNN) [45] is suitable for modelling sequential data. Unlike the feedforward neural network, there are loops and memories in RNN to remember former computations. Variants such as Long Short Term Memory (LSTM) and Gated Recurrent Unit (GRU) networks are often deployed in practice to overcome the vanishing gradient problem.
- The Restricted Boltzmann Machine (RBM) is a two-layer neural network consisting of a visible layer and a hidden layer. It can be easily stacked into a deep net. *Restricted* here means that there are no intra-layer communications in the visible or hidden layer.

- Neural Autoregressive Distribution Estimation (NADE) [81, 153] is an unsupervised neural network built atop an autoregressive model and a feedforward neural network. It is a tractable and efficient estimator for modeling data distribution and densities.
- The Adversarial Network (AN) [46] is a generative neural network which consists of a discriminator and a generator. The two neural networks are trained simultaneously by competing with each other in a minimax game framework.
- Attentional Models (AM) are differentiable neural architectures that are capable of learning the relative importance of different segments of the target sequence/image/memory. The attention mechanism is typically ubiquitous and had its inception in the computer vision and natural language processing domains. However, it has also been an emerging trend in deep recommender systems research.
- Deep Reinforcement Learning (DRL) [106]. Reinforcement learning operates on a trial-and-error paradigm. The whole framework mainly consists of the following components: agents, environments, states, actions, and rewards. The combinations of deep neural networks and reinforcement learning formulates DRL, which has achieved human-level performance across multiple domains such as games and self-driving cars. Deep neural networks enable the agent to get knowledge from raw data and derive efficient representations without handcrafted features and domain heuristics.

Note that there are numerous advanced models emerging each year; here we only briefly listed some important ones. Readers who are interested in the details or more advanced models are referred to Goodfellow et al. [45].

2.3 Why Deep Neural Networks for Recommendation?

Before diving into the details of recent advances, it is beneficial to understand the reasons for applying deep learning techniques to recommender systems. It is evident that numerous deep recommender systems have been proposed in a short span of several years. The field is indeed bustling with innovation. At this point, it would be easy to question the *need* for so many different architectures and/or possibly even the utility of neural networks for the problem domain. Along the same tangent, it would be apt to provide a clear rationale for each proposed architecture and to describe which scenario it would be most beneficial for. All in all, this question is highly relevant to the issue of tasks, domains, and recommender scenarios. Among the most attractive properties of neural architectures is that they are (i) end-to-end differentiable [45] and (ii) provide suitable *inductive biases* catered to the input data type. As such, if there is an inherent structure that the model can exploit, then deep neural networks ought to be useful. For instance, CNNs and RNNs have long exploited the intrinsic structure in vision (and/or human language). Similarly, the sequential structures of sessions or click-logs are highly suitable for the inductive biases provided by recurrent/convolutional models [56, 144, 176].

Moreover, deep neural networks are also composite in the sense that multiple neural building blocks can be composed into a single (gigantic) differentiable function and trained end-to-end. The key advantage here is when dealing with *content-based* recommendation. This is inevitable when modeling users/items on the web, where multimodal data are commonplace. For instance, when dealing with textual data (reviews [203], tweets [44] etc.), image data (social posts, product images), CNNs/RNNs become indispensable neural building blocks. Here, the traditional alternative (designing modality-specific features etc.) becomes significantly less attractive and, consequently, the recommender systems cannot take advantage of joint (end-to-end) representation learning. In some sense, developments in the field of recommender systems are also tightly coupled with advanced research in related modalities (such as vision or language communities). For example, to

process reviews, one would have to perform costly preprocessing (e.g., keyphrase extraction, topic modeling etc.) while newer deep learning-based approaches are able to ingest all textual information end-to-end [203]. All in all, the capabilities of deep learning in this aspect can be regarded as paradigm-shifting, and the ability to represent images, text and interactions in a unified joint framework [198] is not possible without these recent advances.

Pertaining to the interaction-only setting (i.e., matrix completion or collaborative ranking problem), the key idea here is that deep neural networks are justified when there is a huge amount of complexity or when there is a large number of training instances. In He et al. [53], the authors used an MLP to approximate the interaction function and showed reasonable performance gains over traditional methods such as MF. While these neural models perform better, we also note that standard machine learning models such as BPR, MF, and CML are known to perform reasonably well when trained with momentum-based gradient descent on interaction-only data [146]. However, we can also consider these models to be neural architectures as well, since they take advantage of recent deep learning advances such as Adam, Dropout, or Batch Normalization [53, 196]. It is also easy to see that traditional recommender algorithms (matrix factorization, factorization machines, etc.) can also be expressed as neural/differentiable architectures [53, 54] and trained efficiently with a framework such as Tensorflow or Pytorch, enabling efficient GPU-enabled training and free automatic differentiation.

To recapitulate, we summarize the strengths and possible limitations of deep learning-based recommendation models that readers might bear in mind when trying to employ them for practice use.

- **Nonlinear Transformation**. Contrary to linear models, deep neural networks are capable of modeling the nonlinearity in data with nonlinear activations such as relu, sigmoid, tanh, and the like. This property makes it possible to capture complex and intricate user/item interaction patterns. Conventional methods such as matrix factorization, factorization machines, and sparse linear model are essentially linear models. For example, matrix factorization models the user/item interaction by linearly combining user and item latent factors [53]; Factorization machines are members of the multivariate linear family [54]; Obviously, SLIM is a linear regression model with sparsity constraints. The linear assumption, acting as the basis of many traditional recommenders, is oversimplified and will greatly limit their modeling expressiveness. It is well-established that neural networks are able to approximate any continuous function with an arbitrary precision by varying the activation choices and combinations [58, 59]. This property makes it possible to deal with complex interaction patterns and precisely reflect a user's preference.
- **Representation Learning**. Deep neural networks is efficacious in learning the underlying explanatory factors and useful representations from input data. In general, a large amount of descriptive information about items and users is available in real-world applications. Making use of this information provides a way to advance our understanding of items and users, thus resulting in a better recommender. As such, it is a natural choice to apply deep neural networks to representation learning in recommendation models. The advantages of using deep neural networks to assist representation learning are two-fold: (i) it reduces the efforts in hand-crafting feature design. Feature engineering is a labor-intensive work, and deep neural networks enable automatic feature learning from raw data in either an unsupervised or a supervised approach. (ii) It enables recommendation models to include heterogeneous content information such as text, images, audio, and even video. Deep learning networks have made breakthroughs in multimedia data processing and shown potential in representation learning from various sources.
- **Sequence Modeling**. Deep neural networks have shown promising results on a number of sequential modeling tasks such as machine translation, natural language understanding,

speech recognition, chatbots, and many others. RNN and CNN play critical roles in these tasks. RNN achives this with internal memory states, while CNN achieves this with filters sliding along with time. Both of them are widely applicable and flexible in mining sequential structures in data. Modeling sequential signals is an important topic for mining the temporal dynamics of user behaviour and item evolution. For example, next-item/basket prediction and session-based recommendation are typical applications. As such, deep neural networks become a perfect fit for this sequential pattern mining task.

- **Flexibility**. Deep learning techniques possess high flexibility, especially with the advent of many popular deep learning frameworks such as Tensorflow[3], Keras[4], Caffe[5], MXnet[6], DeepLearning4j[7], PyTorch[8], and Theano[9]. Most of these tools are developed in a modular way and have active community and professional support. This good modularization makes development and engineering a lot more efficient. For example, it is easy to combine different neural structures to formulate powerful hybrid models or to replace one module with others. Thus, we can easily build hybrid and composite recommendation models to simultaneously capture different characteristics and factors.

Despite its success, deep learning is well-known to behave as a black box, and providing explainable predictions seems to be a really challenging task. A common argument against deep neural networks is that the hidden weights and activations are generally noninterpretable, thus limiting explainability. A second possible limitation is that deep learning is known to be data-hungry, in the sense that it requires sufficient data in order to fully support its rich parameterization. A third well-established argument against deep learning is the need for extensive hyperparameter tuning. Naturally, complex networks generally introduce more hyperparameters, which require a costly tuning stage when training models (e.g., number and width of layers). Deep learning models operate across a spectrum of different model complexities. For example, RNNs are known to be significantly slower than feedforward techniques, especially due to its autoregressive, step-by-step computation. To this end, the choice of deep model for each domain is often not straightforward. On top of hyperparameter tuning, this may incur a nontrivial amount of architectural search, leading to greater computational demands. For some of these limitations, we provide detailed discussions in Section 4.

3 DEEP LEARNING BASED RECOMMENDATION: STATE-OF-THE-ART

In this section, we first introduce the categories of deep learning-based recommendation models and then highlight state-of-the-art research prototypes, aiming to identify the most notable and promising advanes in recent years.

3.1 Categories of Deep Learning Based Recommendation Models

To provide a bird-eye's view of this field, we classify the existing models based on the types of employed deep learning techniques. We further divide deep learning-based recommendation models into the following two categories. Figure 1 summarizes the classification scheme.

[3]https://www.tensorflow.org/.
[4]https://keras.io/.
[5]http://caffe.berkeleyvision.org/.
[6]https://mxnet.apache.org/.
[7]https://deeplearning4j.org/.
[8]https://pytorch.org/.
[9]http://deeplearning.net/software/theano/.

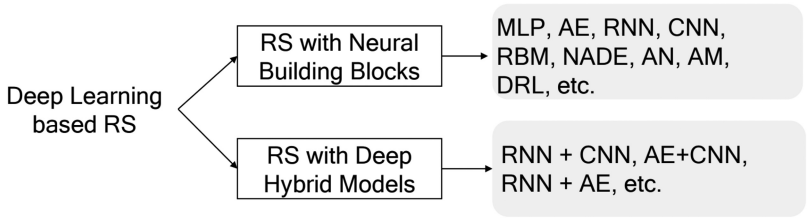

Fig. 1. Categories of deep neural network-based recommendation models.

Table 1. Publications Based on Different Deep Learning Techniques

Categories	Publications
MLP	[2, 13, 20, 27, 38, 47, 53, 54, 66, 92, 95, 158, 167, 186], [12, 39, 93, 112, 135, 155, 183, 184]
Autoencoder	[34, 88, 89, 114, 116, 126, 137, 138, 141, 160, 178, 188, 208], [4, 10, 32, 94, 151, 152, 159, 171, 172, 189, 197, 209, 210]
CNNs	[25, 49, 50, 75, 76, 99, 105, 128, 131, 154, 166, 173, 203, 207], [6, 44, 51, 83, 110, 127, 144, 149, 170, 191, 192]
RNNs	[5, 28, 35, 56, 57, 73, 78, 90, 118, 133, 140, 143, 175–177], [24, 29, 33, 55, 68, 91, 108, 113, 134, 142, 150, 174, 180]
RBM	[42, 71, 72, 101, 124, 168, 181]
NADE	[36, 204, 205]
Neural Attention	[14, 44, 70, 90, 100, 103, 128, 146, 170, 190, 195, 206], [62, 147, 194]
Adversary Network	[9, 52, 163, 165]
DRL	[16, 21, 107, 169, 199–201]
Hybrid Models	[17, 38, 41, 82, 84, 87, 119, 136, 161, 193, 194]

- *Recommendation with Neural Building Blocks.* In this category, models are divided into eight subcategories in conformity with the aforementioned eight deep learning models: MLP-, AE-, CNN-, RNN-, RBM-, NADE-, AM-, AN-, and DRL-based recommender systems. The deep learning technique in use determines the applicability of the recommendation model. For instance, MLP can easily model the nonlinear interactions between users and items; CNNs are capable of extracting local and global representations from heterogeneous data sources such as textual and visual information; RNNs enable the recommender systems to model the temporal dynamics and sequential evolution of content information.
- *Recommendation with Deep Hybrid Models.* Some deep learning-based recommendation models utilize more than one deep learning technique. The flexibility of deep neural networks makes it possible to combine several neural building blocks to complement one another and form a more powerful hybrid model. There are many possible combinations of these deep learning techniques, but not all have been exploited. Note that this differs from the hybrid deep networks in Deng et al. [31], which refer to the deep architectures that make use of both generative and discriminative components.

Table 1 lists all the reviewed models. We organize them following the previously mentioned classification scheme. Additionally, we also summarize some of the publications from the task perspective in Table 2. The reviewed publications are concerned with a variety of tasks. Some of the tasks have started to gain attention due to their use of deep neural networks, such as

Table 2. Deep Neural Network-Based Recommendation Models in Specific Application Fields

Data Sources/Tasks	Notes	Publications
Sequential Information	w/t User ID	[16, 29, 33, 35, 73, 91, 118, 134, 144, 161, 174, 176, 190, 195, 199, 206]
	Session based w/o User ID	[55–57, 68, 73, 100, 102, 103, 118, 143, 149, 150]
	Check-In, POI	[151, 152, 166, 186]
Text	Hash Tags	[44, 110, 119, 159, 183, 184, 194, 210]
	News	[10, 12, 113, 136, 170, 201]
	Review texts	[11, 87, 127, 147, 175, 198, 203]
	Quotes	[82, 142]
Images	Visual features	[2, 14, 25, 49, 50, 84, 99, 105, 112, 166, 173, 180, 192, 193, 198, 207]
Audio	Music	[95, 154, 168, 169]
Video	Videos	[14, 17, 27, 83]
Networks	Citation Network	[9, 38, 66]
	Social Network	[32, 116, 167]
	Cross Domain	[39, 92, 167]
Others	Cold-start	[155, 157, 171, 172]
	Multitask	[5, 73, 87, 175, 188]
	Explainability	[87, 127]

session-based recommendation, image, video recommendations. Some of the tasks might not be novel to the recommendation research area (a detailed review on the "side" information for recommender systems can be found in Shi et al. [132]), but deep learning provides more possibilities to find better solutions. For example, dealing with images and videos would be a tough task without the help of deep learning techniques. The sequence modeling capability of deep neural networks makes it easy to capture the sequential patterns of user behaviors. Some of the specific tasks will be discussed in the following text.

3.2 Multilayer Perceptron Based Recommendation

MLP is a concise but effective network which has been demonstrated to be able to approximate any measurable function to any desired degree of accuracy [59]. As such, it is the basis of numerous advanced approaches and is widely used in many areas.

Neural Extension of Traditional Recommendation Methods. Many existing recommendation models are essentially linear methods. MLP can be used to add nonlinear transformation to existing RS approaches and interpret them into neural extensions.

Neural Collaborative Filtering. In most cases, recommendation is deemed to be a two-way interaction between user preferences and item features. For example, matrix factorization decomposes the rating matrix into low-dimensional user/item latent factors. It is natural to construct a dual neural network to model the two-way interaction between users and items. Neural Network Matrix Factorization (NNMF) [37] and Neural Collaborative Filtering (NCF) [53] are two representative works. Figure 2(a) shows the NCF architecture. Let s_u^{user} and s_i^{item} denote the side information (e.g., user profiles and item features), or just one-hot identifier of user u and item i. The scoring function is defined as follows:

$$\hat{r}_{ui} = f\left(U^T \cdot s_u^{user}, V^T \cdot s_i^{item} | U, V, \theta\right), \tag{1}$$

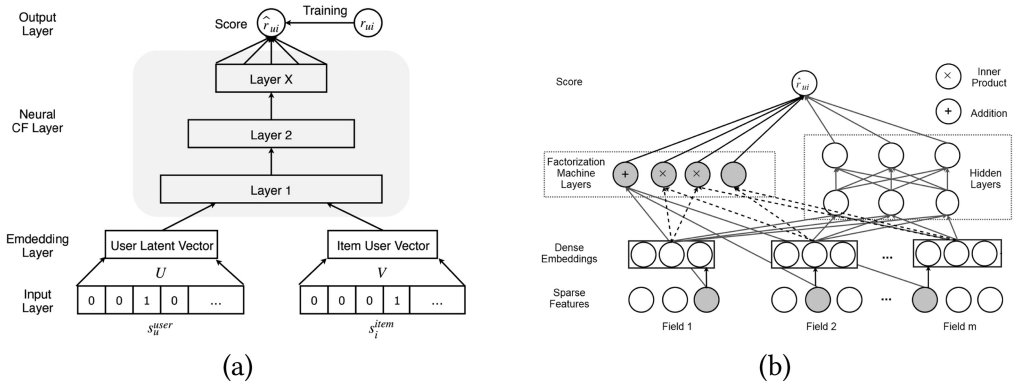

Fig. 2. Illustration of (a) neural collaborative filtering; (b) deep factorization machines.

where function $f(\cdot)$ represents the multilayer perceptron, and θ is the parameters of this network. Traditional MF can be viewed as a special case of NCF. Therefore, it is convenient to fuse the neural interpretation of matrix factorization with MLP to formulate a more general model which makes use of both the linearity of MF and the nonlinearity of MLP to enhance recommendation quality. The whole network can be trained with weighted square loss (for explicit feedback) or binary cross-entropy loss (for implicit feedback). The cross-entropy loss is defined as:

$$\mathcal{L} = - \sum_{(u,i) \in O \cup O^-} r_{ui} \log \hat{r}_{ui} + (1 - r_{ui}) \log(1 - \hat{r}_{ui}). \tag{2}$$

Negative sampling approaches can be used to reduce the number of training unobserved instances. Follow-up work [112, 135] proposed using pairwise ranking loss to enhance performance. He et al. [92, 167] extended the NCF model to cross-domain recommendations. Xue et al. [185] and Zhang et al. [196] showed that the one hot identifier can be replaced with columns or rows of the interaction matrix to retain user/item interaction patterns.

Deep Factorization Machines. DeepFM [47] is an end-to-end model which seamlessly integrates factorization machines and MLP. It is able to model the high-order feature interactions via deep neural networks and low-order interactions with factorization machines. Factorization machines (FM) utilizes addition and inner product operations to capture the linear and pairwise interactions between features (refer to equation (1) in Rendle [120] for more details). MLP leverages the nonlinear activations and deep structure to model the high-order interactions. Combining MLP with FM is inspired by wide and deep networks. It replaces the wide component with a neural interpretation of factorization machines. Compared to wide and deep models, DeepFM does not require tedious feature engineering. Figure 2(b) illustrates the structure of DeepFM. The input of DeepFM x is an m-fields data consisting of pairs (u, i) (identity and features of user and item). For simplicity, the outputs of FM and MLP are denoted as $y_{FM}(x)$ and $y_{MLP}(x)$, respectively. The prediction score is calculated by:

$$\hat{r}_{ui} = \sigma(y_{FM}(x) + y_{MLP}(x)), \tag{3}$$

where $\sigma(\cdot)$ is the sigmoid activation function.

Lian et al. [93] improved DeepMF by proposing an eXtreme deep factorization machine to jointly model the explicit and implicit feature interactions. The explicit high-order feature interactions are learned via a compressed interaction network. A parallel work proposed by He et al. [54] replaces the second-order interactions with MLP and proposed regularizing the model with dropout and batch normalization.

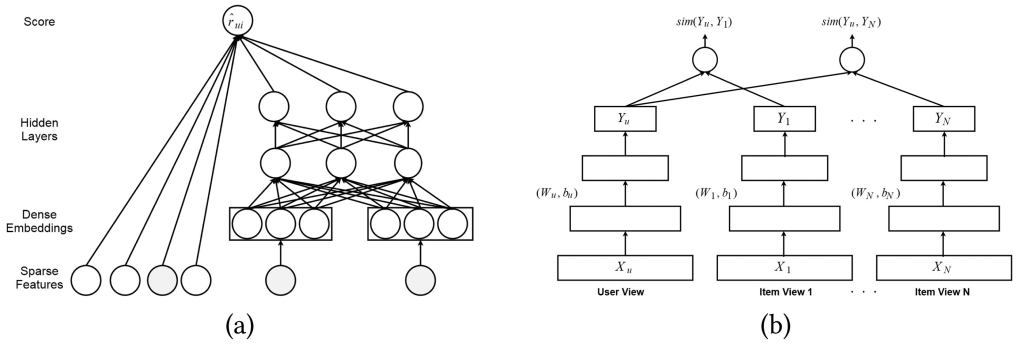

Fig. 3. Illustration of (a) wide and deep learning; (b) multiview deep neural network.

Feature Representation Learning with MLP. Using MLP for feature representation is very straightforward and highly efficient, even though it might not be as expressive as autoencoders, CNNs, and RNNs.

Wide & Deep Learning. This general model (shown in Figure 3(a)) can solve both regression and classification problems, but was initially introduced for an app recommendation for Google play [20]. The wide learning component is a single-layer perceptron which can also be regarded as a generalized linear model. The deep learning component is a multilayer perceptron. The rationale of combining these two learning techniques is that it enables the recommender to capture both memorization and generalization. Memorization achieved by the wide learning component represents the capability of catching direct features from historical data. Meanwhile, the deep learning component catches generalization by producing more general and abstract representations. This model can improve the accuracy as well as the diversity of recommendation.

Formally, wide learning is defined as $y = W_{wide}^T\{x, \phi(x)\} + b$, where W_{wide}^T, b are the model parameters. The input $\{x, \phi(x)\}$ is the concatenated feature set consisting of raw input feature x and transformed (e.g., cross-product transformation to capture the correlations between features) feature $\phi(x)$. Each layer of the deep neural component is in the form of $\alpha^{(l+1)} = f(W_{deep}^{(l)}a^{(l)} + b^{(l)})$, where l indicates the lth layer, and $f(\cdot)$ is the activation function. $W_{deep}^{(l)}$ and $b^{(l)}$ are weight and bias terms. The wide and deep learning model is attained by fusing these two models:

$$P(\hat{r}_{ui} = 1|x) = \sigma\Big(W_{wide}^T\{x, \phi(x)\} + W_{deep}^T a^{(l_f)} + bias\Big). \tag{4}$$

where $\sigma(\cdot)$ is the sigmoid function, $\hat{r}_{ui}$ is the binary rating label, and $a^{(l_f)}$ is the final activation. This joint model is optimized with stochastic back-propagation (follow-the-regularized-leader algorithm). The list of recommendations is generated based on the predicted scores.

By extending this model, Chen et al. [13] devised a locally connected wide and deep learning model for large-scale industrial-level recommendation task. It employs the efficient locally connected network to replace the deep learning component, which decreases the running time by one order of magnitude. An important step of deploying wide and deep learning is selecting features for the wide and deep parts. In other words, the system should be able to determine which features are memorized or generalized. Moreover, the cross-product transformation also is required to be manually designed. These pre-steps will greatly influence the utility of this model. The previously mentioned deep factorization-based model can alleviate the effort in feature engineering.

Covington et al. [27] explored applying MLP in YouTube recommendation. This system divides the recommendation task into two stages: candidate generation and candidate ranking. The candidate generation network retrieves a subset (hundreds) from all video corpus. The ranking network

generates a top-n list (dozens) based on the nearest-neighbor scores from the candidates. We note that the industrial world cares more about feature engineering (e.g., transformation, normalization, crossing) and scalability of recommendation models.

Alashkar et al. [2] proposed an MLP-based model for makeup recommendation. This work uses two identical MLPs to model labeled examples and expert rules, respectively. Parameters of these two networks are updated simultaneously by minimizing the differences between their outputs. This approach demonstrates the efficacy of adopting expert knowledge to guide the learning process of the recommendation model in an MLP framework. It is highly precise even though the expertise acquisition needs a lot of human involvement.

Collaborative Metric Learning (CML). CML [60] replaces the dot product of MF with Euclidean distance because the dot product does not satisfy the triangle inequality of a distance function. The user and item embeddings are learned via maximizing the distance between users and their disliked items and minimizing that between users and their preferred items. In CML, MLP is used to learn representations from item features such as text, images, and tags.

Recommendation with Deep Structured Semantic Model. Deep Structured Semantic Model (DSSM) [65] is a deep neural network for learning semantic representations of entities in a common continuous semantic space and measuring their semantic similarities. It is widely used in information retrieval and is supremely suitable for top-n recommendation [39, 183]. DSSM projects different entities into a common low-dimensional space and computes their similarities with a cosine function. Basic DSSM is made up of MLP, so we put it in this section. Note that more advanced neural layers, such as convolution and max-pooling layers, can also be easily integrated into DSSM.

Deep Semantic Similarity-Based Personalized Recommendation (DSPR) [183] is a tag-aware personalized recommender where each user x_u and item x_i are represented by tag annotations and mapped into a common tag space. Cosine similarity $sim(u, i)$ is applied to decide the relevance of items and users (or user's preference over the item). The loss function of DSPR is defined as follows:

$$\mathcal{L} = -\sum_{(u,i*)} \left[log(e^{sim(u,i*)}) - log\left(\sum_{(u,i^-)\in D^-} e^{sim(u,i^-)} \right) \right], \tag{5}$$

where (u, i^-) are negative samples which are randomly sampled from the negative user/item pairs. The authors [184] further improved DSPR using an autoencoder to learn low-dimensional representations from user/item profiles.

Multi-View Deep Neural Network (MV-DNN) [39] is designed for cross-domain recommendation. It treats users as the pivot view and each domain (supposing Z domains) as an auxiliary view. Apparently, there are Z similarity scores for Z user-domain pairs. Figure 3(b) illustrates the structure of MV-DNN. The loss function of MV-DNN is defined as:

$$\mathcal{L} = \underset{\theta}{argmin} \sum_{j=1}^{Z} \frac{exp(\gamma \cdot cosine(Y_u, Y_{a,j}))}{\sum_{X'\in R^{da}} exp(\gamma \cdot cosine(Y_u, f_a(X')))}, \tag{6}$$

where θ is the model parameters, γ is the smoothing factor, Y_u is the output of user view, a is the index of active view, and R^{da} is the input domain of view a. MV-DNN is capable of scaling up to many domains. However, it is based on the hypothesis that users who have similar tastes in one domain should have similar tastes in other domains. Intuitively, this assumption might be unreasonable in many cases. Therefore, we should have some preliminary knowledge on the correlations across different domains to make the most of MV-DNN.

Table 3. Summary of Four Autoencoder Based Recommendation Models

Vanilla/Denoising AE	Variational AE	Contractive AE	Marginalized AE
[114, 126, 137, 138, 160, 178] [70, 116, 171, 172, 189]	[19, 89, 94]	[197]	[88]

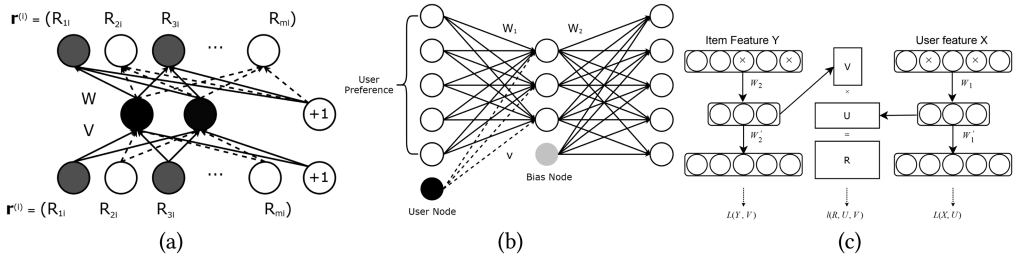

Fig. 4. Illustration of (a) item-based autorec; (b) collaborative denoising autoencoder; (c) deep collaborative filtering framework.

3.3 Autoencoder Based Recommendation

There are two general types of autoencoder-based recommender systems: (i) using autoencoder to learn lower dimensional feature representations at the bottleneck layer or (ii) filling the blanks of the interaction matrix directly in the reconstruction layer. Almost all the autoencoder variants (such as denoising autoencoder, variational autoencoder, contactive autoencoder, and marginalized autoencoder) can be applied to the recommendation task. Table 3 summarizes the recommendation models based on the types of autoencoder in use.

Autoencoder-Based Collaborative Filtering. One of successful application of this method is to interpret the collaborative filtering framework from the autoencoder perspective.

AutoRec [126] takes user partial vectors $\mathbf{r}^{(u)}$ or item partial vectors $\mathbf{r}^{(i)}$ as input and aims to reconstruct them in the output layer. It has two variants: item-based AutoRec (I-AutoRec) and user-based AutoRec (U-AutoRec), corresponding to the two types of inputs. Here, we only introduce I-AutoRec, while U-AutoRec can be easily derived accordingly. Figure 4(a) illustrates the structure of I-AutoRec. Given input $\mathbf{r}^{(i)}$, the reconstruction is $h(\mathbf{r}^{(i)};\theta) = f(W \cdot g(V \cdot \mathbf{r}^{(i)} + \mu) + b)$, where $f(\cdot)$ and $g(\cdot)$ are the activation functions, parameter $\theta = \{W, V, \mu, b\}$. The objective function of I-AutoRec is formulated as follows:

$$\underset{\theta}{argmin} \sum_{i=1}^{N} \left\| \mathbf{r}^{(i)} - h\left(\mathbf{r}^{(i)};\theta\right) \right\|_O^2 + \lambda \cdot reg. \tag{7}$$

Here, $\|\cdot\|_O^2$ means that it only considers observed ratings. The objective function can be optimized by resilient propagation (converges faster and produces comparable results) or the Limited-Memory Broyden Fletcher Goldfarb Shanno (L-BFGS) algorithm. There are four important points about AutoRec that are worth noting before deployment: (i) I-AutoRec performs better than U-AutoRec, which may be due to the higher variance of user partially observed vectors. (ii) Different combinations of activation functions $f(\cdot)$ and $g(\cdot)$ will influence the performance considerably. (iii) Increasing the hidden unit size moderately will improve the result as expanding the hidden layer dimensionality gives AutoRec more capacity to model the characteristics of the input. And (iv) adding more layers to formulate a deep network can lead to slight improvement.

CFN [137, 138] is an extension of AutoRec and has the following two advantages: (i) it deploys denoising techniques, which makes CFN more robust; and (ii) it incorporates side information, such as user profiles and item descriptions, to mitigate the sparsity and cold start influence. The input of CFN is also partial observed vectors, so it also has two variants: I-CFN and U-CFN, taking $\mathbf{r}^{(i)}$ and $\mathbf{r}^{(u)}$ as input, respectively. Masking noise is imposed as a strong regularizer to better deal with missing elements (their values are zero). The authors introduced three widely used corruption approaches to corrupt the input: Gaussian noise, masking noise, and salt-and-pepper noise. A further extension of CFN also incorporates side information. However, instead of just integrating side information in the first layer, CFN injects side information in every layer. Thus, the reconstruction becomes:

$$h\Big(\big\{\tilde{\mathbf{r}}^{(i)}, \mathbf{s}_i\big\}\Big) = f\Big(W_2 \cdot \Big\{g\Big(W_1 \cdot \big\{\mathbf{r}^{(i)}, \mathbf{s}_i\big\} + \mu\Big), \mathbf{s}_i\Big\} + b\Big), \tag{8}$$

where $\mathbf{s}_i$ is side information and $\{\tilde{\mathbf{r}}^{(i)}, \mathbf{s}_i\}$ indicates the concatenation of $\tilde{\mathbf{r}}^{(i)}$ and $\mathbf{s}_i$. Incorporating side information improves the prediction accuracy, speeds up the training process, and enables the model to be more robust.

Collaborative Denoising Auto-Encoder (CDAE). The three models reviewed earlier are mainly designed for rating prediction, while CDAE [178] is principally used for ranking prediction. The input of CDAE is user partially observed implicit feedback $\mathbf{r}^{(u)}_{pref}$. The entry value is 1 if the user likes the item, otherwise 0. It can also be regarded as a preference vector which reflects a user's interests to items. Figure 4(b) illustrates the structure of CDAE. The input of CDAE is corrupted by Gaussian noise. The corrupted input $\tilde{\mathbf{r}}^{(u)}_{pref}$ is drawn from a conditional Gaussian distribution $p(\tilde{\mathbf{r}}^{(u)}_{pref}|\mathbf{r}^{(u)}_{pref})$. The reconstruction is defined as:

$$h\Big(\tilde{\mathbf{r}}^{(u)}_{pref}\Big) = f\Big(W_2 \cdot g\Big(W_1 \cdot \tilde{\mathbf{r}}^{(u)}_{pref} + V_u + b_1\Big) + b_2\Big), \tag{9}$$

where $V_u \in \mathbb{R}^K$ denotes the weight matrix for user node (see Figure 4(b)). This weight matrix is unique for each user and has significant influence on the model's performance. Parameters of CDAE are also learned by minimizing the reconstruction error:

$$\underset{W_1, W_2, V, b_1, b_2}{argmin} \frac{1}{M} \sum_{u=1}^{M} \mathbf{E}_{p(\tilde{\mathbf{r}}^{(u)}_{pref}|\mathbf{r}^{(u)}_{pref})}\Big[\ell\Big(\tilde{\mathbf{r}}^{(u)}_{pref}, h\Big(\tilde{\mathbf{r}}^{(u)}_{pref}\Big)\Big)\Big] + \lambda \cdot reg, \tag{10}$$

where the loss function $\ell(\cdot)$ can be square loss or logistic loss.

CDAE initially updates its parameters using SGD over all feedback. However, the authors argued that it is impractical to take all ratings into consideration in real-world applications, so they proposed a negative sampling technique to sample a small subset from the negative set (items with which the user has not interacted), which reduces the time complexity substantially without degrading the ranking quality.

Multi-VAE and Multi-DAE [94] are proposed variants of the varitional autoencoder for recommendation with implicit data, and they show better performance than CDAE. The authors introduced a principled Bayesian inference approach for parameter estimation and show favorable results over the commonly used likelihood functions.

To the extent of our knowledge, Autoencoder-Based Collaborative Filtering (ACF) [114] is the first autoencoder-based collaborative recommendation model. Instead of using the original partial observed vectors, it decomposes them by integer ratings. For example, if the rating score is an integer in the range of [1–5], each $\mathbf{r}^{(i)}$ will be divided into five partial vectors. Similar to AutoRec and CFN, the cost function of ACF aims at reducing the mean squared error. However, there are two demerits for ACF: (i) it fails to deal with non-integer ratings and (ii) the decomposition of partial observed vectors increases the sparseness of input data and leads to worse prediction accuracy.

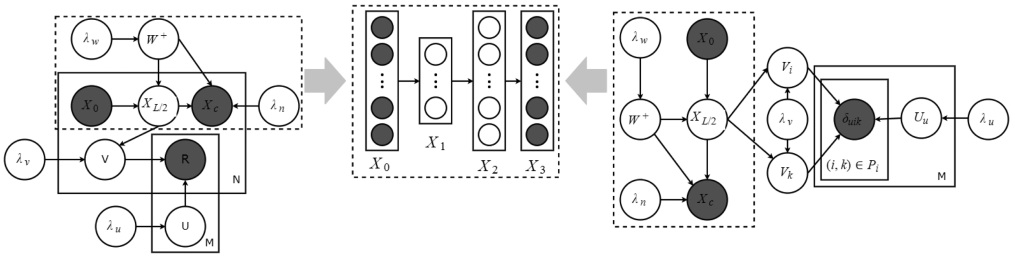

Fig. 5. Graphical model of collaborative deep learning (left) and collaborative deep ranking (right).

Feature Representation Learning with Autoencoder. Autoencoder is a class of powerful feature representation learning approaches. As such, it can also be used in recommender systems to learn feature representations from user/item content features.

Collaborative Deep Learning (CDL). CDL [160] is a hierarchical Bayesian model which integrates a Stacked Denoising Autoencoder (SDAE) into probabilistic matrix factorization. To seamlessly combine deep learning and a recommendation model, the authors proposed a general Bayesian deep learning framework [162] consisting of two tightly hinged components: a perception component (deep neural network) and a task-specific component. Specifically, the perception component of CDL is a probabilistic interpretation of ordinal SDAE, and PMF acts as the task-specific component. This tight combination enables CDL to balance the influences of side information and interaction history. The generative process of CDL is as follows:

(1) For each layer l of the SDAE: (a) For each column n of weight matrix W_l, draw $W_{l,*n} \sim \mathcal{N}(0, \lambda_w^{-1}\mathbf{I}_{D_l})$. (b) Draw the bias vector $b_l \sim \mathcal{N}(0, \lambda_w^{-1}\mathbf{I}_{D_l})$. (c) For each row i of X_l, draw $X_{l,i*} \sim \mathcal{N}(\sigma(X_{l-1,i*}W_l + b_l), \lambda_s^{-1}\mathbf{I}_{D_l})$.
(2) For each item i: (a) Draw a clean input $X_{c,i*} \sim \mathcal{N}(X_{L,i*}, \lambda_n^{-1}\mathbf{I}_{I_i})$. (b) Draw a latent offset vector $\epsilon_i \sim \mathcal{N}(0, \lambda_v^{-1}\mathbf{I}_D)$ and set the latent item vector: $V_i = \epsilon_i + X_{\frac{L}{2},i*}^T$.
(3) Draw a latent user vector for each user u, $U_u \sim \mathcal{N}(0, \lambda_u^{-1}\mathbf{I}_D)$.
(4) Draw a rating r_{ui} for each user/item pair (u, i), $r_{ui} \sim \mathcal{N}(U_u^T V_i, C_{ui}^{-1})$,

where W_l and b_l are the weight matrix and bias vector for layer l, X_l represents layer l. λ_w, λ_s, λ_n, λ_v, λ_u are hyper-parameters, and C_{ui} is a confidence parameter for determining the confidence of observations [63]. Figure 5 (left) illustrates the graphical model of CDL. The authors exploited an EM-style algorithm to learn the parameters. In each iteration, it updates U and V first, and then updates W and b by fixing U and V. The authors also introduced a sampling-based algorithm [162] to avoid the local optimum.

Before CDL, Wang et al. [159] proposed a similar model, a relational SDAE (RSDAE), for tag recommendation. The difference between CDL and RSDAE is that RSDAE replaces the PMF with a relational information matrix. Another extension of CDL is the Collaborative Variational Autoencoder (CVAE) [89], which replaces the deep neural component of CDL with a variational autoencoder. CVAE learns probabilistic latent variables for content information and can easily incorporate multimedia (video, images) data sources.

Collaborative Deep Ranking (CDR). CDR [189] is devised specifically in a pairwise framework for top-n recommendation. Some studies have demonstrated that a pairwise model is more suitable for ranking lists generation [121, 178, 189]. Experimental results also show that CDR outperforms CDL in terms of ranking prediction. Figure 5(right) presents the structure of CDR. The first and second generative process steps of CDR are the same as CDL. The third and fourth steps are replaced by the following step:

- For each user u: (a) Draw a latent user vector for u, $U_u \sim \mathcal{N}(0, \lambda_u^{-1}\mathbf{I}_D)$. (b) For each pairwise preference $(i, j) \in P_i$, where $P_i = \{(i, j) : r_{ui} - r_{uj} > 0\}$, draw the estimator, $\delta_{uij} \sim \mathcal{N}(U_u^T V_i - U_u^T V_j, C_{uij}^{-1})$,

where $\delta_{uij} = r_{ui} - r_{uj}$ represents the pairwise relationship of user's preference on item i and item j and C_{uij}^{-1} is a confidence value which indicates how much user u prefers item i than item j. The optimization process is performed in the same manner as CDL.

Deep Collaborative Filtering Framework is a general framework for unifying deep learning approaches with a collaborative filtering model [88]. This framework makes it easy to utilize deep feature learning techniques to build hybrid collaborative models. The previously mentioned work (e.g., [154, 160, 168]) can be viewed as special cases of this general framework. Formally, the deep collaborative filtering framework is defined as follows:

$$\arg\min_{U,V} \ell(R, U, V) + \beta\left(\|U\|_F^2 + \|V\|_F^2\right) + \gamma\mathcal{L}(X, U) + \delta\mathcal{L}(Y, V), \tag{11}$$

where β, γ, and δ are tradeoff parameters to balance the influences of these three components, X and Y are side information, and $\ell(\cdot)$ is the loss of collaborative filtering model. $\mathcal{L}(X, U)$ and $\mathcal{L}(Y, V)$ act as hinges for connecting deep learning and collaborative models and link side information with latent factors. On top of this framework, the authors proposed the Marginalized Denoising Autoencoder-Based Collaborative Filtering (mDA-CF) model. Compared to CDL, mDA-CF explores a more computationally efficient variant of autoencoder: the marginalized denoising autoencoder [15]. It saves the computational costs for searching for a sufficiently corrupted version of input by marginalizing out the corrupted input, which makes mDA-CF more scalable than CDL. In addition, mDA-CF embeds the content information of items and users while CDL only considers the effects of item features.

AutoSVD++ [197] makes use of a contractive autoencoder [123] to learn item feature representations, then integrates them into the classic recommendation model, SVD++ [79]. The proposed model has the following advantages: (i) compared to other autoencoder variants, the contractive autoencoder captures infinitesimal input variations, (ii) it models the implicit feedback to further enhance the accuracy, and (iii) an efficient training algorithm is designed to reduce the training time.

HRCD [171, 172] is a hybrid collaborative model based on an autoencoder and time SVD++ [80]. It is a time-aware model which uses SDAE to learn item representations from raw features and aims at solving the cold item problem.

3.4 Convolutional Neural Networks Based Recommendation

CNNs are powerful in processing unstructured multimedia data with convolution and pool operations. Most of the CNN-based recommendation models utilize CNNs for feature extraction.

Feature Representation Learning with CNNs. CNNs can be used for feature representation learning from multiple sources such as image, text, audio, video, and more.

CNNs for Image Feature Extraction. Wang et al. [166] investigated the influences of visual features to Point-of-Interest (POI) recommendation and proposed a visual content enhanced POI recommender system (VPOI). VPOI adopts CNNs to extract image features. The recommendation model is built on PMF by exploring the interactions between (i) visual content and latent user factor and (ii) visual content and latent location factor. Chu et al. [25] exploited the effectiveness of visual information (e.g., images of food and furnishings of the restaurant) in restaurant recommendation. The visual features extracted by CNN jointly with the text representation are input into MF, BPRMF, and FM to test their performance. Results show that visual information improves

the performance to some degree but not significantly. He et al. [50] designed a Visual Bayesian Personalized Ranking (VBPR) algorithm by incorporating visual features (learned via CNNs) into matrix factorization. He et al. [49] extended VBPR by exploring user's fashion awareness and the evolution of visual factors that a user considers when selecting items. Yu et al. [192] proposed a coupled matrix and tensor factorization model for aesthetic-based clothing recommendation in which CNNs are used to learn the images features and aesthetic features. Nguyen et al. [110] proposed a personalized tag recommendation model based on CNNs. It utilizes the convolutional and max-pooling layers to get visual features from images. User information is injected to generate a personalized recommendation. To optimize this network, the BPR objective is adopted to maximize the differences between the relevant and irrelevant tags. Lei et al. [84] proposed a comparative deep learning model with CNNs for image recommendation. This network consists of two CNNs which are used for image representation learning and an MLP for user preferences modeling. It compares two images (one positive image that the user likes and one negative image that the user dislikes) against a user. The training data are made up of triplets: t (user U_t, positive image I_t^+, negative image I_t^-). This assumes that the distance between user and positive image $D(\pi(U_t), \phi(I_t^+))$ should be closer than the distance between user and negative images $D(\pi(U_t), \phi(I_t^-))$, where $D(\cdot)$ is the distance metric (e.g., Euclidean distance). ConTagNet [119] is a context-aware tag recommender system. The image features are learned by CNNs. The context representations are processed by a two-layer fully connected feedforward neural network. The outputs of two neural networks are concatenated and fed into a softmax function to predict the probability of candidate tags.

CNNs for Text Feature Extraction. DeepCoNN [203] adopts two parallel CNNs to model user behaviors and item properties from review texts. This model alleviates the sparsity problem and enhances the model's interpretability by exploiting rich semantic representations of review texts with CNNs. It utilizes a word embedding technique to map the review texts into a lower-dimensional semantic space as well as to keep the word sequence information. The extracted review representations then pass through a convolutional layer with different kernels, a max-pooling layer, and a fully connected layer consecutively. The output of the user network x_u and item network x_i are finally concatenated as the input of the prediction layer where the factorization machine is applied to capture their interactions for rating prediction. Catherine et al. [11] mentioned that DeepCoNN only works well when the review text written by the target user for the target item is available at test time, which is unreasonable. As such, they extended it by introducing a latent layer to represent the target user/target item pair. This model does not access the reviews during validation/test and can retain good accuracy. Shen et al. [131] built an e-learning resources recommendation model. It uses CNNs to extract item features from text information of learning resources such as the introduction and content of learning material, and it follows the same procedure as Van den Oord et al. [154] to perform recommendations. ConvMF [75] combines CNNs with PMF in a similar way as CDL. CDL uses an autoencoder to learn the item feature representations, while ConvMF employs CNNs to learn high-level item representations. The main advantage of ConvMF over CDL is that CNNs are able to capture more accurate contextual information of the items via word embedding and convolutional kernels. Tuan et al. [149] proposed using CNNs to learn feature representations from item content information (e.g., name, descriptions, identifier, and category) to enhance the accuracy of session-based recommendation.

CNNs for Audio and Video Feature Extraction. Van et al. [154] proposed using CNNs to extract features from music signals. The convolutional kernels and pooling layers allow operations at multiple timescales. This content-based model can alleviate the cold start problem (music has not been consumed) of music recommendation. Lee et al. [83] proposed extracting audio features with the prominent CNNs model ResNet. The recommendation is performed in a collaborative metric learning framework, similar to CML.

CNN-Based Collaborative Filtering. Directly applying CNNs to vanilla collaborative filtering is also viable. For example, He et al. [51] proposed using CNNs to improve NCF and presented the ConvNCF. It uses the outer product instead of the dot product to model user/item interaction patterns. CNNs are applied over the result of the outer product and could capture the high-order correlations among embeddings dimensions. Tang et al. [144] presented sequential recommendation (with user identifier) with CNNs, where two CNNs (hierarchical and vertical) are used to model the union-level sequential patterns and skip behaviors for sequence-aware recommendation.

Graph CNNs for Recommendation. The Graph Convolutional Network is a powerful tool for non-Eulcidean data such as social networks, knowledge graphs, protein-interaction networks, and the like [77]. Interactions in the recommendation area can also be viewed as a structured dataset (bipartite graph). Thus, it can also be applied to recommendation tasks. For example, Berg et al. [6] proposed considering the recommendation problem as a link prediction task with graph CNNs. This framework makes it easy to integrate user/item side information, such as social networks and item relationships, into the recommendation model. Ying et al. [191] proposed using graph CNNs for recommendations in Pinterest[10]. This model generates item embeddings from both graph structure as well item feature information with random walk and graph CNNs, and is suitable for very large-scale web recommenders. The proposed model has been deployed in Pinterest to address a variety of real-world recommendation tasks.

3.5 Recurrent Neural Networks Based Recommendation

RNNs are suitable for sequential data processing. As such, RNNs become a natural choice for dealing with the temporal dynamics of interactions and sequential patterns of user behaviours, as well as side information with sequential signals, such as texts, audio, and the like.

Session-Based Recommendation Without User Identifier. In many real-world applications or websites, the system usually does not bother users to log in so that it has no access to a user's identifier or her long-term consumption habits or interests. However, session or cookie mechanisms enable those systems to get a user's short term preferences. This is a relatively unappreciated task in recommender systems due to the extreme sparsity of training data. Recent advances have demonstrated the efficacy of RNNs in solving this issue [56, 143, 177].

GRU4Rec. Hidasi et al. [56] proposed a session-based recommendation model, GRU4Rec, based on GRU (shown in Figure 6(a)). The input is the actual state of a session with 1-of-N encoding, where N is the number of items. The coordinate will be 1 if the corresponding item is active in this session, otherwise 0. The output is the likelihood of being the next in the session for each item. To efficiently train the proposed framework, the authors proposed a session-parallel mini-batches algorithm and a sampling method for output. The ranking loss, which is coined TOP1, has the following form:

$$\mathcal{L}_s = \frac{1}{S}\sum_{j=1}^{S}\sigma(\hat{r}_{sj} - \hat{r}_{si}) + \sigma(\hat{r}_{sj}^2), \tag{12}$$

where S is the sample size, $\hat{r}_{si}$ and $\hat{r}_{sj}$ are the scores on negative item i and positive item j at session s, and σ is the logistic sigmoid function. The last term is used as a regularization. Note that BPR loss is also viable. A recent work [55] found that the original TOP1 loss and BPR loss defined in Hidasi et al. [56] suffer from the gradient vanishing problem; as such, two novel loss functions: TOP1-max and BPR-max are proposed.

[10] https://www.pinterest.com.

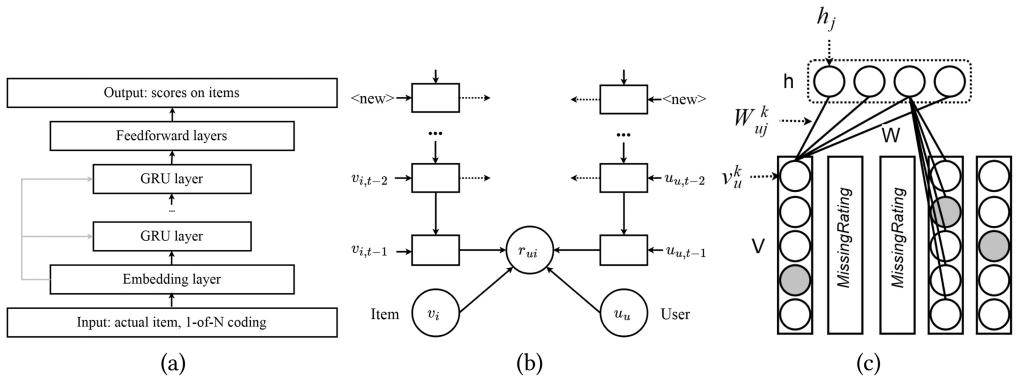

Fig. 6. Illustration of (a) session-based recommendation with RNN; (b) recurrent recommender network; (c) restricted Boltzmann machine-based collaborative filtering.

The follow-up work [143] proposed several strategies to further improve this model: (i) augment the click sequences with sequence preprocessing and dropout regularization, (ii) adapt to temporal changes by pretraining with full training data and fine-tuning the model with more recent click-sequences, (iii) distill the model with *privileged information* with a teacher model, and (iv) use item embedding to decrease the number of parameters for faster computation.

Wu et al. [177] designed a session-based recommendation model for a real-world e-commerce website. It utilizes the basic RNNs to predict what a user will buy next based on the click history. To minimize the computation costs, it only keeps a finite number of the latest states while collapsing the older states into a single history state. This method helps to balance the tradeoff between computation costs and prediction accuracy. Quadrana et al. [118] presented a hierarchical recurrent neural network for session-based recommendation. This model can also deal with session-aware recommendations when user identifiers are present.

The previously mentioned three session-based models do not consider any side information. Two extensions [57, 133] demonstrate that side information has an effect on enhancing session recommendation quality. Hidasi et al. [57] introduced a parallel architecture for session-based recommendation which utilizes three GRUs to learn representations from identity one-hot vectors, image feature vectors, and text feature vectors. The outputs of these three GRUs are concatenated and fed into a nonlinear activation to predict the next items in that session. Smirnova et al. [133] proposed a context-aware session-based recommender system based on conditional RNNs. It injects context information into input and output layers. Experimental results from these two models suggest that models incorporating additional information outperform those solely based on historical interactions.

Despite the success of RNNs in session-based recommendation, Jannach and Ludewig [68] indicated that some trivial methods (e.g., simple neighbourhood approach) could achieve the same or even better accuracy results as GRU4Rec while being computationally much more efficient. Combining the neighbourhood with RNNs methods can usually lead to best performance. Such cases are not accidental in the deep learning and machine learning research, and we suggest that more experiments should be done and that detailed analyses such as ablation studies and hyper-parameters studies should be conducted to justify the proposed approaches and baselines. More detailed discussions can be found in Lipton and Steinhardt [97] and in Section 4.8.

Sequential Recommendation with User Identifier. Unlike a session-based recommender, where user identifiers are usually not present, the following studies deal with the sequential recommendation task with known user identifications.

Recurrent Recommender Network (RRN) [176] is a nonparametric recommendation model built on RNNs (shown in Figure 6(b)). It is capable of modeling the seasonal evolution of items and changes in user preferences over time. RRN uses two LSTM networks as the building blocks to model dynamic user state u_{ut} and item state v_{it}. In the meantime, considering fixed properties such as user long-term interests and item static features, the model also incorporates the stationary latent attributes of user and item: u_u and v_i. The predicted rating of item j given by user i at time t is defined as:

$$\hat{r}_{ui|t} = f(u_{ut}, v_{it}, u_u, v_i), \tag{13}$$

where u_{ut} and v_{it} are learned from LSTM, and u_u and v_i are learned by the standard matrix factorization. The optimization is to minimize the square error between predicted and actual rating values.

Wu et al. [175] further improved the RRN model by modeling text reviews and ratings simultaneously. Unlike most text review-enhanced recommendation models [128, 203], this model aims to generate reviews with a character-level LSTM network with user and item latent states. The review generation task can be viewed as an auxiliary task to facilitate rating prediction. This model is able to improve the rating prediction accuracy, but cannot generate coherent and readable review texts. NRT [87], which will be introduced in the following text, can generate readable review tips. Jing et al. [73] proposed a multitask learning framework to simultaneously predict the returning time of users and recommend items. The returning time prediction is motivated by a survival analysis model designed for estimating the probability of patient survival. The authors modified this model by using LSTM to estimate the return time of consumers. The item recommendation is also performed via LSTM from a user's past session actions. Unlike previously mentioned session-based recommendations which focus on recommending in the same session, this model aims to provide intersession recommendations. Li et al. [91] presented a behavior-intensive model for sequential recommendation. This model consists of two components: neural item embedding and discriminative behaviors learning. The latter part is made up of two LSTMs for session and preference behaviors learning, respectively. Christakopoulou et al. [24] designed an interactive recommender with RNNs. The proposed framework aims to address two critical tasks in interactive recommender: ask and respond. RNNs are used to tackle both tasks: predict questions that the user might ask based on her recent behaviors (e.g., watch event) and predict the responses. Donkers et al. [35] designed a novel type of GRU to explicitly represent an individual user for the next item recommendation.

Feature Representation Learning with RNNs. For side information with sequential patterns, using RNNs as the representation learning tool is an advisable choice.

Dai et al. [29] presented a co-evolutionary latent model to capture the co-evolutionary nature of users' and items' latent features. The interactions between users and items play an important role in driving the changes in user preferences and item status. To model the historical interactions, the author proposed using RNNs to automatically learn representations of the influences from drift, evolution, and co-evolution of user and item features.

Bansal et al. [5] proposed using GRUs to encode text sequences into a latent factor model. This hybrid model solves both warm-start and cold-start problems. Furthermore, the authors adopted a multitask regularizer to prevent overfitting and alleviate the sparsity of training data. The main task is rating prediction, while the auxiliary task is item meta-data (e.g., tags, genres) prediction.

Okura et al. [113] proposed using GRUs to learn more expressive aggregation for user browsing history (browsed news) and to recommend news articles with a latent factor model. The results show a significant improvement compared with the traditional word-based approach. The system has been fully deployed to online production services and serves more than 10 million unique users everyday.

Li et al. [87] presented a multitask learning framework, NRT, for predicting ratings as well as generating textual tips for users simultaneously. The generated tips provide concise suggestions and anticipate user's experience with and feelings about certain products. The rating prediction task is modeled by nonlinear layers over item and user latent factors $U \in \mathbb{R}^{k_u \times M}$, $V \in \mathbb{R}^{k_v \times M}$, where k_u and k_v (not necessarily equal) are latent factor dimensions for users and items. The predicted rating r_{ui} and two latent factor matrices are fed into a GRU for tips generation. Here, r_{ui} is used as context information to decide the sentiment of the generated tips. The multitask learning framework enables the whole model to be trained efficiently in an end-to-end paradigm.

Song et al. [136] designed a temporal DSSM model which integrates RNNs into DSSM for recommendation. Based on traditional DSSM, TDSSM replace the left network with item static features, and the right network with two subnetworks to model user static features (with MLP) and user temporal features (with RNNs).

3.6 Restricted Boltzmann Machine Based Recommendation

Salakhutdinov et al. [124] proposed a restricted Boltzmann machine-based recommender system (shown in Figure 6(c)). To the best of our knowledge, it is the first recommendation model built on neural networks. The visible unit of RBM is limited to binary values; therefore, the rating score is represented in a one-hot vector to adapt to this restriction. For example, [0, 0, 0, 1, 0] represents that the user gives a rating score of 4 to this item. Let $h_j, j = 1, \ldots, F$ denote the hidden units with fixed size F. Each user has a unique RBM with shared parameters. Suppose a user rated m items, and the number of visible units is m. Let X be a $K \times m$ matrix where $x_i^y = 1$ if user u rated item i as y and $x_i^y = 0$ otherwise. Then:

$$p\left(v_i^y = 1|h\right) = \frac{exp(b_i^y + \sum_{j=1}^{F} h_j W_{ij}^y)}{\sum_{l=1}^{K} exp(b_i^l + \sum_{j=1}^{F} h_j W_{ij}^l)}, \quad p(h_j = 1|X) = \sigma\left(b_j + \sum_{i=1}^{m}\sum_{y=1}^{K} x_i^y W_{ij}^y\right), \tag{14}$$

where W_{ij}^y represents the weight on the connection between the rating y of item i and the hidden unit j, b_i^y is the bias of rating y for item i, and b_j is the bias of hidden unit j. RBM is not tractable, but the parameters can be learned via the Contrastive Divergence (CD) algorithm [45]. The authors further proposed using a conditional RBM to incorporate implicit feedback. The essence here is that users implicitly tell their preferences by giving ratings, regardless of how they rate items.

The RBM-CF is user-based, where a given user's rating is clamped on to the visible layer. Similarly, we can easily design an item-based RBM-CF if we clamp a given item's rating on the visible layer. Georgiev et al. [42] proposed to combine the user-based and item-based RBM-CF in a unified framework. In this case, the visible units are determined both by user and item hidden units. Liu et al. [101] designed a hybrid RBM-CF which incorporates item features (item categories). This model is also based on conditional RBM. There are two differences between this hybrid model and the conditional RBM-CF with implicit feedback: (i) the conditional layer here is modeled with the binary item genres and (ii) the conditional layer affects both the hidden layer and the visible layer with different connected weights.

3.7 Neural Attention Based Recommendation

The attention mechanism is motivated by human visual attention. For example, people only need to focus on specific parts of visual inputs to understand or recognize them. An attention mechanism is capable of filtering out the uninformative features from raw inputs and reducing the side effects of noisy data. It is an intuitive but effective technique and has garnered considerable attention over recent years across areas such as computer vision [3], natural language processing [104, 156], and speech recognition [22, 23]. Neural attention can not only be used in conjunction with MLP, CNNs,

Table 4. Categories of Neural Attention-Based Recommendation Models

Vanilla Attention	Co-Attention
[14, 44, 70, 90, 100, 103, 128, 146, 170, 190]	[62, 147, 194, 195, 206]

and RNNs, but it also addresses some tasks independently [156]. Integrating an attention mechanism into RNNs enables the RNNs to process long and noisy inputs [23]. Although LSTM can solve the long memory problem theoretically, it is still problematic when dealing with long-range dependencies. An attention mechanism provides a better solution and helps the network to better memorize inputs. Attention-based CNNs are capable of capturing the most informative elements of the inputs [128]. By applying an attention mechanism to a recommender system, one could leverage the attention mechanism to filter out uninformative content and select the most representative items [14] while providing good interpretability. Although a neural attention mechanism is not exactly a standalone deep neural technique, it is still worth discussing it separately due to its widespread use.

In attention-based methods, inputs are weighted with attention scores. As such, calculating the attention scores lives at the heart of neural attention models. Based on the method used for calculating the attention scores, we classify the neural attention models into (i) standard vanilla attention and (ii) co-attention. Vanilla attention utilizes a parameterized context vector to learn to attend, while co-attention is concerned with learning attention weights from two sequences. Self-attention is a special case of co-attention. Recent works [14, 44, 128] demonstrate the capability of attention mechanism in enhancing recommendation performance. Table 4 summarizes the attention-based recommendation models.

Recommendation with Vanilla Attention. Chen et al. [14] proposed an attentive collaborative filtering model by introducing a two-level attention mechanism to a latent factor model. It consists of item-level and component-level attention. The item-level attention is used to select the most representative items to characterize users. The component-level attention aims to capture the most informative features from multimedia auxiliary information for each user. Tay et al. [146] proposed memory-based attention for collaborative metric learning. It introduces a latent relation vector learned via attention to CML. Jhamb et al. [70] proposed using an attention mechanism to improve the performance of an autoencoder-based CF. Liu et al. [100] proposed a short-term attention and memory priority-based model, in which both long- and short-term user interests are integrated for session-based recommendation. Ying et al. [190] proposed a hierarchical attention model for sequential recommendation. Two attention networks are used to model user long- and short-term interests.

Introducing an attention mechanism into RNNs could significantly improve their performance. Li et al. [90] proposed such an attention-based LSTM model for hashtag recommendation. This work takes advantage of both RNNs and attention mechanisms to capture sequential properties and recognize informative words from microblog posts. Loyala et al. [103] proposed an encoder-decoder architecture with attention for user session and intents modeling. This model consists of two RNNs and could capture transition regularities in a more expressive way.

Vanilla attention can also work in conjunction with CNNs for recommender tasks. Gong et al. [44] proposed an attention-based CNN system for hashtag recommendations in a microblog. It treats hashtag recommendation as a multilabel classification problem. The proposed model consists of a global channel and a local attention channel. The global channel is made up of convolution filters and max-pooling layers. All words are encoded in the input of global channel. The local attention channel has an attention layer with given window size and threshold to select informative

words (known as trigger words in this work). Hence, only trigger words are at play in the subsequent layers. In the follow-up work [128], Seo et al. made use of two neural networks (as in [44], but without the last two layers) to learn feature representations from user and item review texts and to predict rating scores with a dot product in the final layer. Wang et al. [170] presented a combined model for article recommendation, in which CNNs are used to learn article representations and attention is utilized to deal with the diverse variance of editors' selection behavior.

Recommendation with Co-Attention. Zhang et al. [195] proposed a combined model, AttRec, which improves sequential recommendation performance by capitalizing on the strength of both self-attention and metric learning. It uses self-attention to learn a user's short-term intents from her recent interactions and takes advantage of metric learning to learn more expressive user and item embemddings. Zhou et al. [206] proposed using self-attention for user heterogeneous behaviour modeling. Self-attention is a simple yet effective mechanism and has shown superior performance over CNNs and RNNs in terms of the sequential recommendation task. We believe that it has the capability to replace many complex neural models, and more investigations are expected. Tay et al. [147] proposed a review-based recommendation system with multipointer co-attention. The co-attention enables the model to select information reviews via co-learning from both user and item reviews. Zhang et al. [194] proposed a co-attention-based hashtag recommendation model that integrates both visual and textual information. Shi et al. [62] proposed a neural co-attention model for a personalized ranking task with meta-path.

3.8 Neural AutoRegressive Based Recommendation

As mentioned earlier, RBM is not tractable, thus we usually use the Contrastive Divergence algorithm to approximate the log-likelihood gradient on the parameters [81], which also limits the use of RBM-CF. The so-called Neural Autoregressive Distribution Estimator (NADE) is a tractable distribution estimator that provides a desirable alternative to RBM. Inspired by RBM-CF, Zheng et al. [205] proposed a NADE-based collaborative filtering model (CF-NADE). CF-NADE models the distribution of user ratings. Here, we present a detailed example to illustrate how CF-NADE works. Suppose we have 4 items: m1 (rating is 4), m2 (rating is 2), m3 (rating is 3), and m4 (rating is 5). The CF-NADE models the joint probability of the rating vector r by the chain rule: $p(\mathbf{r}) = \prod_{i=1}^{D} p(r_{m_{o_i}} | \mathbf{r}_{m_{o_{<i}}})$, where D is the number of items that the user has rated, o is the D-tuple in the permutations of $(1, 2, \ldots, D)$, m_i is the index of the ith rated item, and $r_{m_{o_i}}$ is the rating that the user gives to item m_{o_i}. More specifically, the procedure is as follows: (i) the probability that the user gives $m1$ 4 stars is conditioned on nothing; (ii) the probability that the user gives $m2$ 2 stars is conditioned on giving $m1$ 4 stars; (iii) the probability that the user gives $m3$ 3 stars is conditioned on giving $m1$ 4 stars and $m2$ 2 stars; and (iv) the probability that the user gives $m4$ 5 stars conditioned on giving $m1$ 4 stars, $m2$ 2 stars, and $m3$ 3 stars.

Ideally, the order of items should follow the timestamps of ratings. However, an empirical study shows that random drawing also yields good performances. This model can be further extended to a deep model. In the follow-up paper, Zheng et al. [204] proposed incorporating implicit feedback to overcome the sparsity problem of the rating matrix. Du et al. [36] further improved this model with a user/item co-autoregressive approach, which achieves better performance in both rating estimation and personalized ranking tasks.

3.9 Deep Reinforcement Learning for Recommendation

Most recommendation models consider the recommendation process as static, which makes it difficult to capture a user's temporal intentions and to respond in a timely manner. In recent years, DRL has begun to garner attention [21, 107, 169, 199–201] in making personalized

recommendation. Zhao et al. [200] proposed a DRL framework, DEERS, for recommendation with both negative and positive feedback in a sequential interaction setting. Zhao et al. [199] explored the page-wise recommendation scenario with DRL; the proposed framework DeepPage is able to adaptively optimize a page of items based on a user's real-time actions. Zheng et al. [201] proposed a news recommendation system, DRN, using DRL to tackle three challenges: (i) dynamic changes of news content and user preference, (ii) incorporating return patterns (to the service) of users, and (iii) increasing the diversity of recommendations. Chen et al. [16] proposed a robust deep Q-learning algorithm to address the unstable reward estimation issue with two strategies: stratified sampling replay and approximate regretted reward. Choi et al. [21] proposed solving the cold-start problem with RL and bi-clustering. Munemasa et al. [107] proposed using DRL for stores recommendation.

Reinforcement Learning techniques, such as the contextual-bandit approach [86], have shown superior recommendation performance in real-world applications. Deep neural networks increase the practicality of RL and make it possible to model various extra information for designing real-time recommendation strategies.

3.10 Adversarial Network Based Recommendation

IRGAN [163] is the first model which applies GAN to the information retrieval area. Specifically, the authors demonstrated its capability in three information retrieval tasks, including web search, item recommendation, and question answering. In this survey, we mainly focus on how to use IRGAN to recommend items.

First, we introduce the general framework of IRGAN. Traditional GAN consists of a discriminator and a generator. There are two schools of thinking in information retrieval: generative retrieval and discriminative retrieval. Generative retrieval assumes that there is an underlying generative process between documents and queries, and retrieval tasks can be achieved by generating relevant document d given a query q. Discriminative retrieval learns to predict the relevance score r given labeled relevant query-document pairs. The aim of IRGAN is to combine these two currents of thought into a unified model and make them play a minimax game, like a generator and discriminator in GAN. The generative retrieval aims to generate relevant documents similar to ground truth to fool the discriminative retrieval model.

Formally, let $p_{true}(d|q_n, r)$ refer to the user's relevance (preference) distribution. The generative retrieval model $p_\theta(d|q_n, r)$ tries to approximate the true relevance distribution. Discriminative retrieval $f_\phi(q, d)$ tries to distinguish between relevant and nonrelevant documents. Similar to the objective function of GAN, the overall objective is formulated as follows:

$$J^{G^*,D^*} = \min_\theta \max_\phi \sum_{n=1}^{N} (\mathbb{E}_{d\sim p_{true}(d|q_n,r)}[logD(d|q_n)] + \mathbb{E}_{d\sim p_\theta(d|q_n,r)}[log(1 - D(d|q_n))]), \quad (15)$$

where $D(d|q_n) = \sigma(f_\phi(q, d))$, σ represents the sigmoid function and θ and ϕ are the parameters for generative and discriminative retrieval, respectively. Parameter θ and ϕ can be learned alternately with gradient descent.

The preceding objective equation is constructed for pointwise relevance estimation. In some specific tasks, it should be in a pairwise paradigm to generate higher quality ranking lists. Here, suppose $p_\theta(d|q_n, r)$ is given by a softmax function:

$$p_\theta(d_i|q, r) = \frac{exp(g_\theta(q, d_i))}{\sum_{d_j} exp(g_\theta(q, d_j))}. \quad (16)$$

$g_\theta(q, d)$ is the chance of document d being generated from query q. In a real-word retrieval system, both $g_\theta(q, d)$ and $f_\phi(q, d)$ are task-specific. They can either have the same or different

formulations. The authors modeled them with the same function for convenience and defined them as:$g_\theta(q,d) = s_\theta(q,d)$ and $f_\phi(q,d) = s_\phi(q,d)$. In the item recommendation scenario, the authors adopted the matrix factorization to formulate $s(\cdot)$. It can be substituted with other advanced models, such as factorization machines or a neural network.

He et al. [52] proposed an adversarial personalized ranking approach which enhances the Bayesian personalized ranking with adversarial training. It plays a minimax game between the original BPR objective and the adversary, which adds noise or permutations to maximize the BPR loss. Cai et al. [9] proposed a GAN-based representation learning approach for a heterogeneous bibliographic network, which can effectively address the personalized citation recommendation task. Wang et al. [165] proposed using GAN to generate negative samples for a memory network-based streaming recommender system. Experiments show that the proposed GAN-based sampler could significantly improve performance.

3.11 Deep Hybrid Models for Recommendation

With the good flexibility of deep neural networks, many neural building blocks can be integrated to formalize more powerful and expressive models. Here, we summarize the existing models which have been proved to be effective in some application fields.

CNNs and Autoencoder. Collaborative Knowledge Based Embedding (CKE) [193] combines CNNs with an autoencoder for images feature extraction. CKE can be viewed as a further step of CDL. CDL only considers item text information (e.g., abstracts of articles and plots of items), while CKE leverages structural content, textual content, and visual content with different embedding techniques. Structural information includes the attributes of items and the relationships among items and users. CKE adopts TransR [96], a heterogeneous network embedding method, for interpreting structural information. Similarly, CKE employs SDAE to learn feature representations from textual information. As for visual information, CKE adopts a stacked convolutional autoencoder (SCAE). SCAE makes efficient use of convolution by replacing the fully connected layers of SDAE with convolutional layers. The recommendation process is done in a probabilistic form similar to CDL.

CNNs and RNNs. Lee et al. [82] proposed a deep hybrid model with RNNs and CNNs for quote recommendation. Quote recommendation is viewed as a task of generating a ranked list of quotes given the query texts or dialogues (each dialogue contains a sequence of tweets). It applies CNNs to learn significant local semantics from tweets and maps them to a distributional vectors. These distributional vectors are further processed by LSTM to compute the relevance of target quotes to the given tweet dialogues.

Zhang et al. [194] proposed a CNNs and RNNs-based hybrid model for hashtag recommendation. Given a tweet with corresponding images, the authors utilized CNNs to extract features from images and LSTM to learn text features from tweets. Meanwhile, the authors proposed a co-attention mechanism to model the correlation influences and balance the contribution of texts and images.

Ebsesu et al. [38] presented a neural citation network which integrates CNNs with RNNs in an encoder-decoder framework for citation recommendation. In this model, CNNs act as the encoder that captures the long-term dependencies from citation context. The RNNs work as a decoder which learns the probability of a word in the cited paper's title, given all previous words together with representations attained by CNNs.

Chen et al. [17] proposed an integrated framework with CNNs and RNNs for personalized key frame (in videos) recommendation, in which CNNs are used to learn feature representations from key frame images and RNNs are used to process the textual features.

RNNs and Autoencoder. The former mentioned collaborative deep learning model is not very robust and is incapable of modeling sequences of text information. Wang et al. [161] further

exploited integrating RNNs and a denoising autoencoder to overcome this limitations. The authors first designed a generalization of RNNs called a robust recurrent network. Based on the robust recurrent network, the authors proposed the hierarchical Bayesian recommendation model called CRAE. CRAE also consists of encoding and decoding parts, but it replaces feedforward neural layers with RNNs, which enables CRAE to capture the sequences of item content information. Furthermore, the authors designed a wildcard denoising and a beta-pooling technique to prevent the model from overfitting.

RNNs with DRL. Wang et al. [164] proposed combining supervised deep reinforcement learning with RNNs for treatment recommendation. The framework can learn prescription policy from the indicator and evaluation signals. Experiments demonstrate that this system could infer and discover optimal treatments automatically.

4 FUTURE RESEARCH DIRECTIONS AND OPEN ISSUES

While existing works have established a solid foundation for deep recommender systems research, this section outlines several promising prospective research directions. We also elaborate on several open issues that we believe are critical to the present state of the field.

4.1 Joint Representation Learning from User and Item Content Information

Making accurate recommendations requires deep understanding of an item's characteristics and a user's actual demands and preferences [1, 85]. Naturally, this can be achieved by exploiting the abundantly available auxiliary information. For example, context information tailors services and products according to a user's circumstances and surroundings [152] and mitigates cold-start influence. Implicit feedback indicates users' implicit intention and is easy to collect, whereas gathering explicit feedback is a resource-demanding task. Although existing works have investigated the efficacy of a deep learning model in mining user and item profiles [92, 197], implicit feedback [50, 189, 197, 204], contextual information [38, 75, 119, 150, 152], and review texts [87, 128, 175, 203] for recommendation, they do not utilize these various forms of side information in a comprehensive manner to take the full advantages of the available data. Moreover, there are few works investigating users' footprints (e.g., tweets or Facebook posts) from social media [61] and the physical world (e.g., Internet of things) [187]. One can infer a user's temporal interests or intentions from these side data resources, and the deep learning method is a desirable and powerful tool for integrating these additional pieces of information. The capability of deep learning in processing heterogeneous data sources also brings more opportunities in recommending diverse items using unstructured data such as textual, visual, audio, and video features.

Additionally, feature engineering has not been fully studied in the recommendation research community, but it is essential and widely employed in industrial applications [20, 27]. However, most existing models require manually crafted and selected features, which is time-consuming and tedious. The deep neural network is a promising tool for automatic feature crafting because it reduces manual intervention [130]. There is also an added advantage of representation learning from free texts, images, or data that exists in the wild, without having to design intricate feature engineering pipelines. More intensive studies on deep feature engineering specific to recommender systems are expected to save human effort as well as improve recommendation quality.

An interesting forward-looking research problem is how to design neural architectures that best exploit the availability of other modes of data. One recent work potentially paving the way toward models of this nature is the Joint Representation Learning framework [198]. Learning joint (possibly multimodal) representations of user and items will likely become a next emerging trend in recommender systems research. To this end, deep learning taking on this aspect could be used to

design better inductive biases (hybrid neural architectures) in an end-to-end fashion (e.g., example, reasoning over different modalities (text, images, interaction) data for better recommendation performance).

4.2 Explainable Recommendation with Deep Learning

A common interpretation is that deep neural networks are highly noninterpretable. As such, making explainable recommendations seem to be an uphill task. Along the same vein, it would be also natural to assume that big, complex neural models are just fitting the data without any *true* understanding (see subsequent section on machine reasoning for recommendation). This is precisely why this direction is both exciting and also crucial. There are mainly two ways that explainable deep learning is important. The first is to make explainable predictions to users, allowing them to understand the factors behind the network's recommendations (i.e., why was this item/service recommended?) [127, 179]. The second track is mainly focused on explainability to the practitioner, probing weights and activations to understand more about the model [146].

Currently, attentional models [127, 147, 179] have more or less eased the noninterpretable concerns of neural models. If anything, attention models have instead led to greater extents of interpretability since the attention weights not only give insights about the inner workings of the model but are also able to provide explainable results to users. While this has been an existing direction of research "pre deep learning," attentional models are not only capable of enhancing performance but also enjoy greater explainability. This further motivates the use of deep learning for recommendation.

Notably, it is both intuitive and natural that a model's explainability and interpretability strongly rely on the application domain and usage of content information. For example, some authors [127, 147] mainly use reviews as a medium of interpretability (with reviews leading to predictions). Many other mediums/modalities can be considered, such as images [18].

To this end, a promising direction and next step would to be to design *better* attentional mechanisms, possibly to the level of providing conversational or generative explanations (as in [87]). Given that models are already capable of highlighting what contributes to their decision, we believe that this is the next frontier.

4.3 Going Deeper for Recommendation

From former studies [53, 53, 178, 196], we found that the performance of most neural CF models plateaus at three to four layers. Going deeper has shown promising performance over shallow networks in many tasks [48, 64]; nonetheless, going deeper in the context of deep neural network-based recommendation systems remains largely unclear. If going deeper produces favorable results, how do we train the deep architecture? If not, what is the reason behind this? A possibility is to look into auxiliary losses at different layers (similar to [148]), albeit hierarchically instead of sequentially. Another possibility is to vary layer-wise learning rates for each layer of the deep network or apply some residual strategies.

4.4 Machine Reasoning for Recommendation

There have been numerous recent advances in *machine reasoning* in deep learning, often involving reasoning over natural language or visual input [67, 125, 182]. We believe that tasks like machine reading, reasoning, question answering, or even visual reasoning will have big impacts on the field of recommender systems. These tasks are often glazed over, given that they seem completely arbitrary and irrelevant with respect to recommender systems. However, recommender systems often require reasoning over a single (or multiple) modality (reviews, text, images, meta-data), which would eventually require borrowing (and adapting) techniques from these related fields.

Fundamentally, recommendation and reasoning (e.g., question answering) are highly related in the sense that they are both information retrieval problems.

The single most impactful architectural innovation with neural architectures that are capable of machine reasoning is the key idea of attention [156, 182]. Notably, this key intuition has already (and very recently) demonstrated effectiveness on several recommender problems. Tay et al. [147] proposed a co-attentive architecture for *reasoning over reviews* and showed that different recommendation domains have different "evidence aggregation" patterns. For interaction-only recommendation, similar reasoning architectures have utilized similar co-attentive mechanisms for reasoning over meta-paths [62]. To this end, a next frontier for recommender systems is possibly to adapt to situations that require multistep inference and reasoning. A simple example would be to reason over a user's social profile, purchases , and the like (i.e., reasoning over multiple modalities) to recommend a product. All in all, we can expect that reasoning architectures will take the foreground in recommender systems research.

4.5 Cross Domain Recommendation with Deep Neural Networks

Currently, many large companies offer diversified products or services to customers. For example, Google provides us with web searches, mobile applications, and news services, and we can buy books, electronics, and clothes from Amazon. Single-domain recommender systems only focus on one domain while ignoring user interests in other domains, which also exacerbates sparsity and cold-start problems [74]. A cross-domain recommender system, which assists target domain recommendation with the knowledge learned from source domains, provides a desirable solution for these problems. One of the most widely studied topics in cross-domain recommendation is *transfer learning*, which aims to improve learning tasks in one domain by using knowledge transferred from other domains [40, 115]. Deep learning is well suited to transfer learning as it learns high-level abstractions that disentangle the variations of different domains. Several existing works [39, 92] indicate the efficacy of deep learning in catching the generalizations and differences across different domains and generating better recommendations on cross-domain platforms. Therefore, this is a promising but largely underexplored area where more studies are expected.

4.6 Deep Multi-Task Learning for Recommendation

Multitask learning has led to successes in many deep learning tasks, from computer vision to natural language processing [26, 31]. Among the reviewed studies, several works [5, 73, 87, 188] also applied multitask learning to recommender systems in a deep neural framework and achieved some improvements over single-task learning. The advantages of applying deep neural network-based multitask learning are three-fold: (i) learning several tasks at a time can prevent overfitting by generalizing the shared hidden representations, (ii) introducing auxiliary task enables the model to provide interpretable output for explaining the recommendation, and (iii) the multitask mechanism provides implicit data augmentation for alleviating the sparsity problem. Multitasks can be utilized in traditional recommender systems [111], while deep learning enables systems to be integrated in a tighter fashion. Apart from introducing side tasks, we can also deploy multitask learning for cross-domain recommendation, with each specific task generating recommendations for each domain.

4.7 Scalability of Deep Neural Networks for Recommendation

The increasing data volumes in the big data era pose challenges to real-world applications. Consequently, scalability is critical to the usefulness of recommendation models in real-world systems, and the time complexity will also be a principal consideration for choosing models. Fortunately, deep learning has been demonstrated to be very effective and promising in big data analytics [109]

especially with the increase of GPU computation power. However, more future works should be focus on efficient recommendation by exploring the following problems: (i) incremental learning for nonstationary and streaming data, such as large volumes of incoming users and items; (ii) computation efficiency for high-dimensional tensors and multimedia data sources; and (iii) balancing of model complexity and scalability with the exponential growth of parameters. A promising area of research in this area involves knowledge distillation, which has been explored in Tang and Wang [145] for learning small/compact models for inference in recommender systems. The key idea is to train a smaller student model that absorbs knowledge from the large teacher model. Given that inference time is crucial for real-time applications at a million/billion user scale, we believe that this is another promising direction which warrants further investigation. Another promising direction involves compression techniques [129]. The high-dimensional input data can be compressed to compact embedding to reduce the space and computation time during model learning.

4.8 The Field Needs Better, More Unified and Harder Evaluation

Each time a new model is proposed, it is expected that the publication offers evaluation and comparisons against several baselines. The selection of baselines and datasets on most papers is seemingly arbitrary, and authors generally have free rein over the choices of datasets/baselines. There are several issues with this.

First, this creates an inconsistent reporting of scores, with each author reporting his or her own assortment of results. As of now, there is seemingly no consensus on a general ranking of models (notably, we acknowledge that the *no free lunch theorem* exists). Occasionally, we find that results can be conflicting, and relative positions change very frequently. For example, the scores of NCF in Zheng et al. [202] are relatively ranked very low as compared to the original paper that proposed the model [53]. This makes the relative benchmark of new neural models extremely challenging. The question is, how do we solve this? Looking into neighbouring fields (computer vision or natural language processing), this is indeed perplexing. Why is there no MNIST, ImageNet, or SQuAD for recommender systems? As such, we believe that a suite of standardized evaluation datasets should be proposed.

We also note that datasets such as MovieLens are commonly used by many practioners in evaluating their models. However, test splits are often arbitrary (randomized). The second problem is that there is no control over the evaluation procedure. To this end, we urge the recommender systems community to follow the CV/NLP communities and establish a hidden/blinded test set in which prediction results can be only submitted via a web interface (such as Kaggle).

Finally, a third recurring problem is that there is no control over the difficulty of test samples in recommender system results. Is splitting by time best? How do we know if test samples are either too trivial or impossible to infer? Without designing proper test sets, we argue that it is in fact hard to estimate and measure progress in the field. To this end, we believe that the field of recommender systems has a lot to learn from the computer vision or NLP communities.

5 CONCLUSION

In this article, we provided an extensive review of the most notable works to date on deep learning-based recommender systems. We proposed a classification scheme for organizing and clustering existing publications, and we highlighted a bunch of influential research prototypes. We also discussed the advantages/disadvantages of using deep learning techniques for recommendation tasks. Additionally, we detail some of the most pressing open problems and promising future extensions. Both deep learning and recommender systems are ongoing hot research topics in recent decades. There are a large number of newly developing techniques and emerging models each year. We

hope this survey can provide readers with a comprehensive understanding of the key aspects of this field, clarify the most notable advances, and shed some light on future studies.

REFERENCES

[1] Gediminas Adomavicius and Alexander Tuzhilin. 2005. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering* 17, 6 (2005), 734–749.

[2] Taleb Alashkar, Songyao Jiang, Shuyang Wang, and Yun Fu. 2017. Examples-rules guided deep neural network for makeup recommendation. In *Proceedings of the AAAI*. 941–947. https://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14773

[3] Jimmy Ba, Volodymyr Mnih, and Koray Kavukcuoglu. 2014. Multiple object recognition with visual attention. *arXiv preprint arXiv:1412.7755* (2014).

[4] Bing Bai, Yushun Fan, Wei Tan, and Jia Zhang. 2017. DLTSR: A deep learning framework for recommendation of long-tail web services. *IEEE Transactions on Services Computing* (2017), 1–1.

[5] Trapit Bansal, David Belanger, and Andrew McCallum. 2016. Ask the gru: Multi-task learning for deep text recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*. 107–114. http://doi.acm.org/10.1145/2959100.2959180

[6] Rianne van den Berg, Thomas N. Kipf, and Max Welling. 2017. Graph convolutional matrix completion. *arXiv preprint arXiv:1706.02263* (2017).

[7] Basiliyos Tilahun Betru, Charles Awono Onana, and Bernabe Batchakui. 2017. Deep learning methods on recommender system: A survey of state-of-the-art. *International Journal of Computer Applications* 162, 10 (Mar 2017).

[8] Robin Burke. 2002. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction* 12, 4 (2002), 331–370.

[9] Xiaoyan Cai, Junwei Han, and Libin Yang. 2018. Generative adversarial network based heterogeneous bibliographic network representation for personalized citation recommendation. In *Proceedings of the AAAI*. 5747–5754. https://aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16310

[10] S. Cao, N. Yang, and Z. Liu. 2017. Online news recommender based on stacked auto-encoder. In *Proceedings of the ICIS*. 721–726. https://ieeexplore.ieee.org/document/7960088

[11] Rose Catherine and William Cohen. 2017. Transnets: Learning to transform for recommendation. In *Proceedings of the Recsys*. 288–296. http://doi.acm.org/10.1145/3109859.3109878

[12] Cheng Chen, Xiangwu Meng, Zhenghua Xu, and Thomas Lukasiewicz. 2017. Location-aware personalized news recommendation with deep semantic analysis. *IEEE Access* 5 (2017), 1624–1638.

[13] Cen Chen, Peilin Zhao, Longfei Li, Jun Zhou, Xiaolong Li, and Minghui Qiu. Locally connected deep learning framework for industrial-scale recommender systems. In *Proceedings of the WWW*. 769–770. https://doi.org/10.1145/3041021.3054227

[14] Jingyuan Chen, Hanwang Zhang, Xiangnan He, Liqiang Nie, Wei Liu, and Tat-Seng Chua. 2017. Attentive collaborative filtering: Multimedia recommendation with item- and component-level attention. In *Proceedings of the SIGIR*. ACM. http://doi.acm.org/10.1145/3077136.3080797

[15] Minmin Chen, Zhixiang Xu, Kilian Weinberger, and Fei Sha. 2012. Marginalized denoising autoencoders for domain adaptation. *arXiv preprint arXiv:1206.4683* (2012).

[16] Shi-Yong Chen, Yang Yu, Qing Da, Jun Tan, Hai-Kuan Huang, and Hai-Hong Tang. 2018. Stabilizing reinforcement learning in dynamic environment with application to online recommendation. In *Proceedings of the SIGKDD*. 1187–1196. http://doi.acm.org/10.1145/3219819.3220122

[17] Xu Chen, Yongfeng Zhang, Qingyao Ai, Hongteng Xu, Junchi Yan, and Zheng Qin. 2017. Personalized key frame recommendation. In *Proceedings of the SIGIR*. 315–324. http://doi.acm.org/10.1145/3077136.3080776

[18] Xu Chen, Yongfeng Zhang, Hongteng Xu, Yixin Cao, Zheng Qin, and Hongyuan Zha. 2018. Visually explainable recommendation. *arXiv preprint arXiv:1801.10288* (2018).

[19] Yifan Chen and Maarten de Rijke. 2018. A collective variational autoencoder for top-N recommendation with side information. *arXiv preprint arXiv:1807.05730* (2018).

[20] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, and others. 2016. Wide & deep learning for recommender systems. In *Proceedings of the Recsys*. 7–10. http://doi.acm.org/10.1145/2988450.2988454

[21] Sungwoon Choi, Heonseok Ha, Uiwon Hwang, Chanju Kim, Jung-Woo Ha, and Sungroh Yoon. 2018. Reinforcement learning based recommender system using biclustering technique. *arXiv preprint arXiv:1801.05532* (2018).

[22] Jan Chorowski, Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. End-to-end continuous speech recognition using attention-based recurrent NN: first results. *arXiv preprint arXiv:1412.1602* (2014).

[23] Jan K. Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio. 2015. Attention-based models for speech recognition. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., Montreal, 577–585.

[24] Konstantina Christakopoulou, Alex Beutel, Rui Li, Sagar Jain, and Ed H. Chi. 2018. Q&R: A two-stage approach toward interactive recommendation. In *Proceedings of the SIGKDD*. 139–148. http://doi.acm.org/10.1145/3219819.3219894

[25] Wei-Ta Chu and Ya-Lun Tsai. 2017. A hybrid recommendation system considering visual information for predicting favorite restaurants. *WWWJ* (2017), 1–19.

[26] Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning, Helsinki, Finland*. 160–167.

[27] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for YouTube recommendations. In *Proceedings of the Recsys*. 191–198. http://doi.acm.org/10.1145/2959100.2959190

[28] Hanjun Dai, Yichen Wang, Rakshit Trivedi, and Le Song. 2016. Deep coevolutionary network: Embedding user and item features for recommendation. arXiv preprint. *arXiv preprint arXiv:1609.03675* (2016).

[29] Hanjun Dai, Yichen Wang, Rakshit Trivedi, and Le Song. 2016. Recurrent coevolutionary latent feature processes for continuous-time recommendation. In *Proceedings of the Recsys*. 29–34. http://doi.acm.org/10.1145/2988450.2988451

[30] James Davidson, Benjamin Liebald, Junning Liu, Palash Nandy, Taylor Van Vleet, Ullas Gargi, Sujoy Gupta, Yu He, Mike Lambert, Blake Livingston, and Dasarathi Sampath. 2010. The Youtube video recommendation system. In *Proceedings of the Recsys*. 293–296. http://doi.acm.org/10.1145/1864708.1864770

[31] Li Deng, Dong Yu, and others. 2014. Deep learning: Methods and applications. *Foundations and Trends® in Signal Processing* 7, 3–4 (2014), 197–387.

[32] Shuiguang Deng, Longtao Huang, Guandong Xu, Xindong Wu, and Zhaohui Wu. 2017. On deep learning for trust-aware recommendations in social networks. *IEEE Transactions on Neural Networks and Learning Systems* 28, 5 (2017), 1164–1177.

[33] Robin Devooght and Hugues Bersini. 2016. Collaborative filtering with recurrent neural networks. *arXiv preprint arXiv:1608.07400* (2016).

[34] Xin Dong, Lei Yu, Zhonghuo Wu, Yuxia Sun, Lingfeng Yuan, and Fangxi Zhang. 2017. A hybrid collaborative filtering model with deep structure for recommender systems. In *Proceedings of the AAAI*. 1309–1315. https://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14676

[35] Tim Donkers, Benedikt Loepp, and Jürgen Ziegler. 2017. Sequential user-based recurrent neural network recommendations. In *Proceedings of the Recsys*. 152–160. http://doi.acm.org/10.1145/3109859.3109877

[36] Chao Du, Chongxuan Li, Yin Zheng, Jun Zhu, and Bo Zhang. 2016. Collaborative filtering with user-item co-autoregressive models. *arXiv preprint arXiv:1612.07146* (2016).

[37] Gintare Karolina Dziugaite and Daniel M. Roy. 2015. Neural network matrix factorization. *arXiv preprint arXiv:1511.06443* (2015).

[38] Travis Ebesu and Yi Fang. 2017. Neural citation network for context-aware citation recommendation. In *Proceedings of the SIGIR*. ACM, Shinjuku, Tokyo, 1093–1096.

[39] Ali Mamdouh Elkahky, Yang Song, and Xiaodong He. 2015. A multi-view deep learning approach for cross domain user modeling in recommendation systems. In *WWW*. 278–288. https://doi.org/10.1145/2736277.2741667

[40] Ignacio Fernández-Tobías, Iván Cantador, Marius Kaminskas, and Francesco Ricci. 2012. Cross-domain recommender systems: A survey of the state of the art. In *Proceedings of the Spanish Conference on Information Retrieval*. 24.

[41] Jianfeng Gao, Li Deng, Michael Gamon, Xiaodong He, and Patrick Pantel. 2014. Modeling interestingness with deep neural networks. (June 13 2014). US Patent App. 14/304,863.

[42] Kostadin Georgiev and Preslav Nakov. 2013. A non-iid framework for collaborative filtering with restricted boltzmann machines. In *Proceedings of the ICML*. 1148–1156. http://proceedings.mlr.press/v28/georgiev13.html

[43] Carlos A Gomez-Uribe and Neil Hunt. 2016. The netflix recommender system: Algorithms, business value, and innovation. *TMIS* 6, 4 (2016), 13.

[44] Yuyun Gong and Qi Zhang. 2016. Hashtag recommendation using attention-based convolutional neural network. In *Proceedings of the IJCAI*. 2782–2788. http://dl.acm.org/citation.cfm?id=3060832.3061010

[45] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

[46] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Proceedings of the NIPS*. 2672–2680. http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf.

[47] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: A factorization-machine based neural network for CTR prediction. In *Proceedings of the IJCAI*. 2782–2788. http://dl.acm.org/citation.cfm?id=3172077.3172127

[48] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, Las Vegas, NV, 770–778.
[49] Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proceedings of the WWW*. 507–517. https://doi.org/10.1145/2736277.2741667
[50] Ruining He and Julian McAuley. 2016. VBPR: Visual bayesian personalized ranking from implicit feedback. In *Proceedings of the AAAI*. 144–150. http://dl.acm.org/citation.cfm?id=3015812.3015834
[51] Xiangnan He, Xiaoyu Du, Xiang Wang, Feng Tian, Jinhui Tang, and Tat-Seng Chua. 2018. Outer product-based neural collaborative filtering. *arXiv preprint arXiv:1808.03912* (2018).
[52] Xiangnan He, Zhankui He, Xiaoyu Du, and Tat-Seng Chua. 2018. Adversarial personalized ranking for recommendation. In *Proceedings of the SIGIR*. 355–364. http://doi.acm.org/10.1145/3209978.3209981
[53] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the WWW*. 173–182. https://doi.org/10.1145/3038912.3052569
[54] Xiangnan He and Chua Tat-Seng. 2017. Neural factorization machines for sparse predictive analytics. In *Proceedings of the SIGIR*. ACM, Shinjuku, Tokyo, 355–364.
[55] Balázs Hidasi and Alexandros Karatzoglou. 2017. Recurrent neural networks with top-k gains for session-based recommendations. *arXiv preprint arXiv:1706.03847* (2017).
[56] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2015. Session-based recommendations with recurrent neural networks. *Proceedings of the International Conference on Learning Representations*. https://arxiv.org/abs/1511.06939
[57] Balázs Hidasi, Massimo Quadrana, Alexandros Karatzoglou, and Domonkos Tikk. 2016. Parallel recurrent neural network architectures for feature-rich session-based recommendations. In *Proceedings of the Recsys*. 241–248. http://doi.acm.org/10.1145/2959100.2959167
[58] Kurt Hornik. 1991. Approximation capabilities of multilayer feedforward networks. *Neural Networks* 4, 2 (1991), 251–257.
[59] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. 1989. Multilayer feedforward networks are universal approximators. *Neural Networks* 2, 5 (1989), 359–366.
[60] Cheng-Kang Hsieh, Longqi Yang, Yin Cui, Tsung-Yi Lin, Serge Belongie, and Deborah Estrin. 2017. Collaborative metric learning. In *Proceedings of the WWW*. 193–201. https://doi.org/10.1145/3038912.3052639a
[61] Cheng-Kang Hsieh, Longqi Yang, Honghao Wei, Mor Naaman, and Deborah Estrin. 2016. Immersive recommendation: News and event recommendations using personal digital traces. In *Proceedings of the WWW*. 51–62. https://doi.org/10.1145/2872427.2883006
[62] Binbin Hu, Chuan Shi, Wayne Xin Zhao, and Philip S. Yu. 2018. Leveraging meta-path based context for top-n recommendation with a neural co-attention model. In *Proceedings of the SIGKDD*. 1531–1540. http://doi.acm.org/10.1145/3219819.3219965
[63] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative filtering for implicit feedback datasets. In *Proceedings of the ICDM*. 263–272. http://dx.doi.org/10.1109/ICDM.2008.22
[64] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. 2017. Densely connected convolutional networks. In *Proceedings of the CVPR*, Vol. 1. 3. http://dx.doi.org/10.1109/CVPR.2017.243
[65] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the CIKM*. 2333–2338. http://doi.acm.org/10.1145/2505515.2505665
[66] Wenyi Huang, Zhaohui Wu, Liang Chen, Prasenjit Mitra, and C. Lee Giles. 2015. A neural probabilistic model for context based citation recommendation. In *Proceedings of the AAAI*. 2404–2410. http://dl.acm.org/citation.cfm?id=2886521.2886655
[67] Drew A. Hudson and Christopher D. Manning. 2018. Compositional attention networks for machine reasoning. *arXiv preprint arXiv:1803.03067* (2018).
[68] Dietmar Jannach and Malte Ludewig. 2017. When recurrent neural networks meet the neighborhood for session-based recommendation. In *Proceedings of the Recsys (RecSys'17)*. ACM, Como, 306–310.
[69] Dietmar Jannach, Markus Zanker, Alexander Felfernig, and Gerhard Friedrich. 2010. *Recommender Systems: An Introduction*.
[70] Yogesh Jhamb, Travis Ebesu, and Yi Fang. 2018. Attentive contextual denoising autoencoder for recommendation. (2018), 27–34. http://doi.acm.org/10.1145/3234944.3234956
[71] X. Jia, X. Li, K. Li, V. Gopalakrishnan, G. Xun, and A. Zhang. 2016. Collaborative restricted Boltzmann machine for social event recommendation. In *Proceedings of the ASONAM*. 402–405. http://dl.acm.org/citation.cfm?id=3192424.3192498
[72] Xiaowei Jia, Aosen Wang, Xiaoyi Li, Guangxu Xun, Wenyao Xu, and Aidong Zhang. 2015. Multi-modal learning for video recommendation based on mobile application usage. In *Proceedings of the 2015 IEEE International Conference on Big Data (Big Data)*. IEEE Computer Society, 837–842.

[73] How Jing and Alexander J. Smola. 2017. Neural survival recommender. In *Proceedings of the WSDM*. ACM, 515–524. http://doi.acm.org/10.1145/3018661.3018719
[74] Muhammad Murad Khan, Roliana Ibrahim, and Imran Ghani. 2017. Cross domain recommender systems: A systematic literature review. *ACM Computing Survyes* 50, 3 (June 2017).
[75] Donghyun Kim, Chanyoung Park, Jinoh Oh, Sungyoung Lee, and Hwanjo Yu. 2016. Convolutional matrix factorization for document context-aware recommendation. In *Proceedings of the Recsys*. 233–240. http://doi.acm.org/10.1145/2959100.2959165
[76] Donghyun Kim, Chanyoung Park, Jinoh Oh, and Hwanjo Yu. 2017. Deep hybrid recommender systems via exploiting document context and statistics of items. *Information Sciences* (2017), 72–87.
[77] Thomas N. Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).
[78] Young-Jun Ko, Lucas Maystre, and Matthias Grossglauser. 2016. Collaborative recurrent neural networks for dynamic recommender systems. In *Proceedings of the Asian Conference on Machine Learning*. 366–381.
[79] Yehuda Koren. 2008. Factorization meets the neighborhood: A multifaceted collaborative filtering model. In *Proceedings of the SIGKDD*. 426–434. http://doi.acm.org/10.1145/1401890.1401944
[80] Yehuda Koren. 2010. Collaborative filtering with temporal dynamics. *Communications of the ACM* 53, 4 (2010), 89–97.
[81] Hugo Larochelle and Iain Murray. 2011. The neural autoregressive distribution estimator. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*. 29–37.
[82] Hanbit Lee, Yeonchan Ahn, Haejun Lee, Seungdo Ha, and Sang-goo Lee. 2016. Quote recommendation in dialogue using deep neural network. In *Proceedings of the SIGIR*. 957–960. http://dx.doi.org/10.1145/2911451.2914734
[83] Joonseok Lee, Sami Abu-El-Haija, Balakrishnan Varadarajan, and Apostol Paul Natsev. 2018. Collaborative deep metric learning for video understanding. In *Proceedings of the KDD'18*. ACM, 481–490.
[84] Chenyi Lei, Dong Liu, Weiping Li, Zheng-Jun Zha, and Houqiang Li. 2016. Comparative deep learning of hybrid representations for image recommendations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2545–2553. http://dx.doi.org/10.1109/CVPR.2016.279
[85] Jure Leskovec. 2015. New directions in recommender systems. In *Proceedings of the WSDM*. 3–4. https://dl.acm.org/citation.cfm?doid=2684822.2697044
[86] Lihong Li, Wei Chu, John Langford, and Robert E. Schapire. 2010. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th International Conference on World Wide Web*. ACM, Raleigh, North Carolina, 661–670.
[87] Piji Li, Zihao Wang, Zhaochun Ren, Lidong Bing, and Wai Lam. 2017. Neural rating regression with abstractive tips generation for recommendation. In *Proceedings of the SIGIR*. ACM, London, 345–354.
[88] Sheng Li, Jaya Kawale, and Yun Fu. 2015. Deep collaborative filtering via marginalized denoising auto-encoder. In *Proceedings of the CIKM*. ACM, Melbourne, 811–820.
[89] Xiaopeng Li and James She. 2017. Collaborative variational autoencoder for recommender systems. In *Proceedings of the SIGKDD*. 305–314. http://doi.acm.org/10.1145/3097983.3098077
[90] Yang Li, Ting Liu, Jing Jiang, and Liang Zhang. Hashtag recommendation with topical attention-based LSTM. In *Proceedings of the COLING*. 943–952.
[91] Zhi Li, Hongke Zhao, Qi Liu, Zhenya Huang, Tao Mei, and Enhong Chen. 2018. Learning from history and present: Next-item recommendation via discriminatively exploiting user behaviors. In *Proceedings of the SIGKDD*. 1734–1743. http://doi.acm.org/10.1145/3219819.3220014
[92] Jianxun Lian, Fuzheng Zhang, Xing Xie, and Guangzhong Sun. 2017. CCCFNet: A content-boosted collaborative filtering neural network for cross domain recommender systems. In *Proceedings of the WWW*. 817–818. https://doi.org/10.1145/3041021.3054207
[93] Jianxun Lian, Xiaohuan Zhou, Fuzheng Zhang, Zhongxia Chen, Xing Xie, and Guangzhong Sun. 2018. xDeepFM: Combining explicit and implicit feature interactions for recommender systems. *arXiv preprint arXiv:1803.05170* (2018).
[94] Dawen Liang, Rahul G Krishnan, Matthew D. Hoffman, and Tony Jebara. 2018. Variational autoencoders for collaborative filtering. *arXiv preprint arXiv:1802.05814* (2018).
[95] Dawen Liang, Minshu Zhan, and Daniel P. W. Ellis. 2015. Content-aware collaborative music recommendation using pre-trained neural networks. In *Proceedings of the ISMIR*. 295–301.
[96] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the AAAI*. AAAI Press, Austin, Texas, 2181–2187.
[97] Zachary C. Lipton and Jacob Steinhardt. 2018. Troubling trends in machine learning scholarship. *arXiv preprint arXiv:1807.03341* (2018).
[98] Juntao Liu and Caihua Wu. 2017. *Deep Learning Based Recommendation: A Survey*. Springer Singapore.

[99] Qiang Liu, Shu Wu, and Liang Wang. 2017. DeepStyle: Learning user preferences for visual recommendation. In *Proceedings of the SIGIR*. ACM, 841–844.
[100] Qiao Liu, Yifu Zeng, Refuoe Mokhosi, and Haibin Zhang. 2018. STAMP: Short-term attention/memory priority model for session-based recommendation. In *Proceedings of the SIGKDD*. ACM, London, United Kingdom, 1831–1839.
[101] Xiaomeng Liu, Yuanxin Ouyang, Wenge Rong, and Zhang Xiong. 2015. Item category aware conditional restricted boltzmann machine based recommendation. In *Proceedings of the International Conference on Neural Information Processing*. Springer-Verlag New York, Inc., Istanbul, 609–616.
[102] Pablo Loyola, Chen Liu, and Yu Hirate. 2017. Modeling user session and intent with an attention-based encoder-decoder architecture. In *Proceedings of the Recsys*. ACM, Como, Italy, 147–151.
[103] Pablo Loyola, Chen Liu, and Yu Hirate. 2017. Modeling user session and intent with an attention-based encoder-decoder architecture. In *Proceedings of the Recsys*. ACM, Como, Italy, 147–151.
[104] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025* (2015).
[105] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based recommendations on styles and substitutes. In *Proceedings of the SIGIR*. ACM, Santiago, 43–52.
[106] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, and others. 2015. Human-level control through deep reinforcement learning. *Nature* 518, 7540 (2015), 529.
[107] Isshu Munemasa, Yuta Tomomatsu, Kunioki Hayashi, and Tomohiro Takagi. Deep reinforcement learning for recommender systems. IEEE, 226–233.
[108] Cataldo Musto, Claudio Greco, Alessandro Suglia, and Giovanni Semeraro. 2016. Ask me any rating: A content-based recommender system based on recurrent neural networks. In *Proceedings of the IIR*.
[109] Maryam M. Najafabadi, Flavio Villanustre, Taghi M. Khoshgoftaar, Naeem Seliya, Randall Wald, and Edin Muharemagic. 2015. Deep learning applications and challenges in big data analytics. *Journal of Big Data* 2, 1 (2015), 1.
[110] Hanh T. H. Nguyen, Martin Wistuba, Josif Grabocka, Lucas Rego Drumond, and Lars Schmidt-Thieme. 2017. *Personalized Deep Learning for Tag Recommendation*. Springer International Publishing, 186–197.
[111] Xia Ning and George Karypis. 2010. Multi-task learning for recommender system. In *Proceedings of 2nd Asian Conference on Machine Learning*. 269–284.
[112] Wei Niu, James Caverlee, and Haokai Lu. 2018. Neural personalized ranking for image recommendation. In *Proceedings of the 11th ACM International Conference on Web Search and Data Mining*. ACM, Marina Del Rey, CA, 423–431.
[113] Shumpei Okura, Yukihiro Tagami, Shingo Ono, and Akira Tajima. 2017. Embedding-based news recommendation for millions of users. In *Proceedings of the SIGKDD*. ACM, Halifax, NS, 1933–1942.
[114] Yuanxin Ouyang, Wenqi Liu, Wenge Rong, and Zhang Xiong. 2014. Autoencoder-based collaborative filtering. In *International Conference on Neural Information Processing*. Springer International Publishing, 284–291.
[115] Weike Pan, Evan Wei Xiang, Nathan Nan Liu, and Qiang Yang. 2010. Transfer learning in collaborative filtering for sparsity reduction. In *Proceedings of the AAAI*, Vol. 10. 230–235.
[116] Yiteng Pana, Fazhi Hea, and Haiping Yua. 2017. Trust-aware collaborative denoising auto-encoder for top-n recommendation. *arXiv preprint arXiv:1703.01760* (2017).
[117] Massimo Quadrana, Paolo Cremonesi, and Dietmar Jannach. 2018. Sequence-aware recommender systems. *arXiv preprint arXiv:1802.08452* (2018).
[118] Massimo Quadrana, Alexandros Karatzoglou, Balázs Hidasi, and Paolo Cremonesi. 2017. Personalizing session-based recommendations with hierarchical recurrent neural networks. In *Proceedings of the Recsys*. ACM, Como, 130–137.
[119] Yogesh Singh Rawat and Mohan S. Kankanhalli. 2016. ConTagNet: Exploiting user context for image tag recommendation. In *Proceedings of the 2016 ACM on Multimedia Conference*. ACM, Amsterdam, 1102–1106.
[120] S. Rendle. 2010. Factorization machines. In *2010 IEEE International Conference on Data Mining*.
[121] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*. IEEE Computer Society, 452–461.
[122] Francesco Ricci, Lior Rokach, and Bracha Shapira. 2015. Recommender systems: Introduction and challenges. In *Recommender Systems Handbook*. Springer-Verlag, 1–34.
[123] Salah Rifai, Pascal Vincent, Xavier Muller, Xavier Glorot, and Yoshua Bengio. 2011. Contractive auto-encoders: Explicit invariance during feature extraction. In *Proceedings of the ICML*. Omnipress, Bellevue, Washington, 833–840.
[124] Ruslan Salakhutdinov, Andriy Mnih, and Geoffrey Hinton. 2007. Restricted boltzmann machines for collaborative filtering. In *Proceedings of the ICML*. 791–798. http://doi.acm.org/10.1145/1273496.1273596

[125] Adam Santoro, David Raposo, David G. Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Tim Lillicrap. 2017. A simple neural network module for relational reasoning. In *Proceedings of the NIPS*. 4967–4976. https://arxiv.org/abs/1706.01427

[126] Suvash Sedhain, Aditya Krishna Menon, Scott Sanner, and Lexing Xie. 2015. Autorec: Autoencoders meet collaborative filtering. In *Proceedings of the WWW*. 111–112. http://doi.acm.org/10.1145/2740908.2742726

[127] Sungyong Seo, Jing Huang, Hao Yang, and Yan Liu. 2017. Interpretable convolutional neural networks with dual local and global attention for review rating prediction. In *Proceedings of the Recsys*. 297–305. http://doi.acm.org/10.1145/3109859.3109890

[128] Sungyong Seo, Jing Huang, Hao Yang, and Yan Liu. 2017. Representation learning of users and items for review rating prediction using attention-based convolutional neural network. In *Proceedings of the MLRec*.

[129] Joan Serrà and Alexandros Karatzoglou. 2017. Getting deep recommenders fit: Bloom embeddings for sparse binary input/output networks. In *Proceedings of the Recsys*. 279–287. http://doi.acm.org/10.1145/3109859.3109876

[130] Ying Shan, T. Ryan Hoens, Jian Jiao, Haijing Wang, Dong Yu, and J. C. Mao. 2016. Deep crossing: Web-scale modeling without manually crafted combinatorial features. In *Proceedings of the SIGKDD*. 255–262. http://doi.acm.org/10.1145/2939672.2939704

[131] Xiaoxuan Shen, Baolin Yi, Zhaoli Zhang, Jiangbo Shu, and Hai Liu. 2016. Automatic recommendation technology for learning resources with convolutional neural network. In *Proceedings of the International Symposium on Educational Technology*. IEEE, Beijing, China, 30–34.

[132] Yue Shi, Martha Larson, and Alan Hanjalic. 2014. Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges. *ACM Computing Surveys (CSUR)* 47, 1 (2014), 3.

[133] Elena Smirnova and Flavian Vasile. 2017. Contextual sequence modeling for recommendation with recurrent neural networks. In *Proceedings of the DLRS*. ACM, Como, 2–9.

[134] Harold Soh, Scott Sanner, Madeleine White, and Greg Jamieson. 2017. Deep sequential recommendation for personalized adaptive user interfaces. In *Proceedings of the 22nd International Conference on Intelligent User Interfaces*. 589–593. http://doi.acm.org/10.1145/3025171.3025207

[135] Bo Song, Xin Yang, Yi Cao, and Congfu Xu. 2018. Neural collaborative ranking. *arXiv preprint arXiv:1808.04957* (2018).

[136] Yang Song, Ali Mamdouh Elkahky, and Xiaodong He. 2016. Multi-rate deep learning for temporal recommendation. In *Proceedings of the SIGIR*. 909–912. http://doi.acm.org/10.1145/2911451.2914726

[137] Florian Strub, Romaric Gaudel, and Jérémie Mary. 2016. Hybrid recommender system based on autoencoders. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*. 11–16.

[138] Florian Strub and Jeremie Mary. 2015. Collaborative filtering with stacked denoising autoencoders and sparse inputs. In *Proceedings of the NIPS Workshop*.

[139] Xiaoyuan Su and Taghi M. Khoshgoftaar. 2009. A survey of collaborative filtering techniques. *Advances in Artificial Intelligence* 2009 (2009), 4.

[140] Alessandro Suglia, Claudio Greco, Cataldo Musto, Marco de Gemmis, Pasquale Lops, and Giovanni Semeraro. 2017. A deep architecture for content-based recommendations exploiting recurrent neural networks. In *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization*. ACM, Bratislava, 202–211.

[141] Yosuke Suzuki and Tomonobu Ozaki. 2017. Stacked denoising autoencoder-based deep collaborative filtering using the change of similarity. In *Proceedings of the WAINA*. IEEE, Taipei, 498–502.

[142] Jiwei Tan, Xiaojun Wan, and Jianguo Xiao. 2016. A neural network approach to quote recommendation in writings. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. ACM, Indianapolis, Indiana, 65–74.

[143] Yong Kiam Tan, Xinxing Xu, and Yong Liu. 2016. Improved recurrent neural networks for session-based recommendations. In *Proceedings of the Recsys*. 17–22. http://doi.acm.org/10.1145/2988450.2988452

[144] Jiaxi Tang and Ke Wang. 2018. Personalized top-n sequential recommendation via convolutional sequence embedding. In *Proceedings of the WSDM*. 565–573. http://doi.acm.org/10.1145/3159652.3159656

[145] Jiaxi Tang and Ke Wang. 2018. Ranking distillation: Learning compact ranking models with high performance for recommender system. In *Proceedings of the SIGKDD*. 2289–2298. http://doi.acm.org/10.1145/3219819.3220021

[146] Yi Tay, Luu Anh Tuan, and Siu Cheung Hui. 2018. Latent relational metric learning via memory-based attention for collaborative ranking. In *Proceedings of the WWW*. 729–739. https://doi.org/10.1145/3178876.3186154

[147] Yi Tay, Anh Tuan Luu, and Siu Cheung Hui. 2018. Multi-pointer co-attention networks for recommendation. In *Proceedings of the SIGKDD*. 2309–2318. http://doi.acm.org/10.1145/3219819.3220086

[148] Trieu H. Trinh, Andrew M. Dai, Thang Luong, and Quoc V. Le. 2018. Learning longer-term dependencies in rnns with auxiliary losses. *arXiv preprint arXiv:1803.00144* (2018).

[149] Trinh Xuan Tuan and Tu Minh Phuong. 2017. 3D convolutional networks for session-based recommendation with content features. In *Proceedings of the Recsys*. 138–146. http://doi.acm.org/10.1145/3109859.3109900

[150] Bartlomiej Twardowski. Modelling contextual information in session-aware recommender systems with neural networks. In *Proceedings of the Recsys*. 273–276. http://doi.acm.org/10.1145/2959100.2959162

[151] Moshe Unger. 2015. Latent context-aware recommender systems. In *Proceedings of the Recsys*. 383–386. http://doi.acm.org/10.1145/2792838.2796546

[152] Moshe Unger, Ariel Bar, Bracha Shapira, and Lior Rokach. 2016. Towards latent context-aware recommendation systems. *Knowledge-Based Systems* 104 (2016), 165–178.

[153] Benigno Uria, Marc-Alexandre Côté, Karol Gregor, Iain Murray, and Hugo Larochelle. 2016. Neural autoregressive distribution estimation. *Journal of Machine Learning Research* 17, 205 (2016), 1–37.

[154] Aaron Van den Oord, Sander Dieleman, and Benjamin Schrauwen. 2013. Deep content-based music recommendation. In *Proceedings of the NIPS*. Curran Associates, Inc., 2643–2651.

[155] Manasi Vartak, Arvind Thiagarajan, Conrado Miranda, Jeshua Bratman, and Hugo Larochelle. 2017. A meta-learning perspective on cold-start recommendations for items. In *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnet (Eds.). Curran Associates, Inc., 6904–6914.

[156] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 5998–6008.

[157] Maksims Volkovs, Guangwei Yu, and Tomi Poutanen. 2017. DropoutNet: Addressing cold start in recommender systems. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 4957–4966.

[158] Jeroen B. P. Vuurens, Martha Larson, and Arjen P. de Vries. Exploring deep space: Learning personalized ranking in a semantic space. In *Proceedings of the Recsys*. 23–28.

[159] Hao Wang, Xingjian Shi, and Dit-Yan Yeung. 2015. Relational stacked denoising autoencoder for tag recommendation. In *Proceedings of the AAAI, Boston, MA, USA*. ACM, 3052–3058.

[160] Hao Wang, Naiyan Wang, and Dit-Yan Yeung. 2015. Collaborative deep learning for recommender systems. In *Proceedings of the SIGKDD*. 1235–1244.

[161] Hao Wang, SHI Xingjian, and Dit-Yan Yeung. 2016. Collaborative recurrent autoencoder: Recommend while learning to fill in the blanks. In *Proceedings of the NIPS*. 415–423. http://dl.acm.org/citation.cfm?id=3157096.3157143

[162] Hao Wang and Dit-Yan Yeung. 2016. Towards Bayesian deep learning: A framework and some existing methods. *TKDE* 28, 12 (2016), 3395–3408.

[163] Jun Wang, Lantao Yu, Weinan Zhang, Yu Gong, Yinghui Xu, Benyou Wang, Peng Zhang, and Dell Zhang. IRGAN: A minimax game for unifying generative and discriminative information retrieval models. In *Proceedings of the SIGIR*. ACM, 515–524.

[164] Lu Wang, Wei Zhang, Xiaofeng He, and Hongyuan Zha. 2018. Supervised reinforcement learning with recurrent neural network for dynamic treatment recommendation. In *Proceedings of the SIGKDD*. 2447–2456. http://doi.acm.org/10.1145/3219819.3219961

[165] Qinyong Wang, Hongzhi Yin, Zhiting Hu, Defu Lian, Hao Wang, and Zi Huang. 2018. Neural memory streaming recommender networks with adversarial training. In *Proceedings of the SIGKDD*. 2467–2475. http://doi.acm.org/10.1145/3219819.3220004

[166] Suhang Wang, Yilin Wang, Jiliang Tang, Kai Shu, Suhas Ranganath, and Huan Liu. 2017. What your images reveal: Exploiting visual contents for point-of-interest recommendation. In *Proceedings of the WWW*. 391–400. https://doi.org/10.1145/3038912.3052638

[167] Xiang Wang, Xiangnan He, Liqiang Nie, and Tat-Seng Chua. 2017. Item silk road: Recommending items from information domains to social users. In *Proceedings of the SIGIR*. ACM, 185–194.

[168] Xinxi Wang and Ye Wang. 2014. Improving content-based and hybrid music recommendation using deep learning. In *Proceedings of the MM*. 627–636. http://doi.acm.org/10.1145/2647868.2654940

[169] Xinxi Wang, Yi Wang, David Hsu, and Ye Wang. 2014. Exploration in interactive personalized music recommendation: A reinforcement learning approach. *TOMM* 11, 1 (2014), 7.

[170] Xuejian Wang, Lantao Yu, Kan Ren, Guangyu Tao, Weinan Zhang, Yong Yu, and Jun Wang. 2017. Dynamic attention deep model for article recommendation by learning human editors' demonstration. In *Proceedings of the SIGKDD*. http://doi.acm.org/10.1145/3097983.3098096

[171] Jian Wei, Jianhua He, Kai Chen, Yi Zhou, and Zuoyin Tang. 2016. Collaborative filtering and deep learning based hybrid recommendation for cold start problem. In *IEEE 14th International Conference on Dependable, Autonomic and Secure Computing*. IEEE, 874–877.

[172] Jian Wei, Jianhua He, Kai Chen, Yi Zhou, and Zuoyin Tang. 2017. Collaborative filtering and deep learning based recommendation system for cold start items. *Expert Systems with Applications* 69 (2017), 29–39.

[173] Jiqing Wen, Xiaopeng Li, James She, Soochang Park, and Ming Cheung. 2016. Visual background recommendation for dance performances using dancer-shared images. IEEE, 521–527.

[174] Caihua Wu, Junwei Wang, Juntao Liu, and Wenyu Liu. 2016. Recurrent neural network based recommendation for time heterogeneous feedback. *Knowledge-Based Systems* 109 (2016), 90–103.
[175] Chao-Yuan Wu, Amr Ahmed, Alex Beutel, and Alexander J. Smola. 2016. Joint training of ratings and reviews with recurrent recommender networks. ACM.
[176] Chao-Yuan Wu, Amr Ahmed, Alex Beutel, Alexander J. Smola, and How Jing. 2017. Recurrent recommender networks. In *Proceedings of the WSDM*. 495–503. https://openreview.net/pdf?id=Bkv9FyHYx
[177] Sai Wu, Weichao Ren, Chengchao Yu, Gang Chen, Dongxiang Zhang, and Jingbo Zhu. 2016. Personal recommendation using deep recurrent neural networks in NetEase. In *Proceedings of the ICDE*. 1218–1229. http://doi.org/10.1109/ICDE.2016.7498326
[178] Yao Wu, Christopher DuBois, Alice X. Zheng, and Martin Ester. 2016. Collaborative denoising auto-encoders for top-n recommender systems. In *Proceedings of the WSDM*. 153–162. http://doi.org/10.1109/ICDE.2016.7498326
[179] Jun Xiao, Hao Ye, Xiangnan He, Hanwang Zhang, Fei Wu, and Tat-Seng Chua. 2017. Attentional factorization machines: Learning the weight of feature interactions via attention networks. *arXiv preprint arXiv:1708.04617* (2017).
[180] Ruobing Xie, Zhiyuan Liu, Rui Yan, and Maosong Sun. 2016. Neural emoji recommendation in dialogue systems. *arXiv preprint arXiv:1612.04609* (2016).
[181] Weizhu Xie, Yuanxin Ouyang, Jingshuai Ouyang, Wenge Rong, and Zhang Xiong. 2016. User occupation aware conditional restricted boltzmann machine based recommendation. IEEE, 454–461.
[182] Caiming Xiong, Victor Zhong, and Richard Socher. 2016. Dynamic coattention networks for question answering. *arXiv preprint arXiv:1611.01604* (2016).
[183] Zhenghua Xu, Cheng Chen, Thomas Lukasiewicz, Yishu Miao, and Xiangwu Meng. 2016. Tag-aware personalized recommendation using a deep-semantic similarity model with negative sampling. In *Proceedings of the CIKM*. 1921–1924. http://doi.acm.org/10.1145/2983323.2983874
[184] Zhenghua Xu, Thomas Lukasiewicz, Cheng Chen, Yishu Miao, and Xiangwu Meng. 2017. Tag-aware personalized recommendation using a hybrid deep model. IJCAI, 3196–3202.
[185] Hong-Jian Xue, Xinyu Dai, Jianbing Zhang, Shujian Huang, and Jiajun Chen. 2017. Deep matrix factorization models for recommender systems. In *Proceedings of the IJCAI*. 3203–3209. http://dl.acm.org/citation.cfm?id=3172077.3172336
[186] Carl Yang, Lanxiao Bai, Chao Zhang, Quan Yuan, and Jiawei Han. Bridging collaborative filtering and semi-supervised learning: A neural approach for POI recommendation. In *Proceedings of the SIGKDD*. 1245–1254. http://dl.acm.org/citation.cfm?id=3172077.3172336
[187] Lina Yao, Quan Z. Sheng, Anne HH Ngu, and Xue Li. 2016. Things of interest recommendation by leveraging heterogeneous relations in the internet of things. *ACM Transactions on Internet Technology (TOIT)* 16, 2 (2016), 9.
[188] Baolin Yi, Xiaoxuan Shen, Zhaoli Zhang, Jiangbo Shu, and Hai Liu. 2016. Expanded autoencoder recommendation framework and its application in movie recommendation. In *Proceedings of the SKIMA*. 298–303.
[189] Haochao Ying, Liang Chen, Yuwen Xiong, and Jian Wu. 2016. Collaborative deep ranking: A hybrid pair-wise recommendation algorithm with implicit feedback. In *Proceedings of the PAKDD*. 555–567. http://doi.org/10.1007/978-3-319-31750-2_44
[190] Haochao Ying, Fuzhen Zhuang, Fuzheng Zhang, Yanchi Liu, Guandong Xu, Xing Xie, Hui Xiong, and Jian Wu. 2018. Sequential recommender system based on hierarchical attention networks. In *Proceedings of the IJCAI*. 3926–3932. https://doi.org/10.24963/ijcai.2018/546
[191] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L. Hamilton, and Jure Leskovec. 2018. Graph convolutional neural networks for web-scale recommender systems. *arXiv preprint arXiv:1806.01973* (2018).
[192] Wenhui Yu, Huidi Zhang, Xiangnan He, Xu Chen, Li Xiong, and Zheng Qin. 2018. Aesthetic-based clothing recommendation. In *Proceedings of the WWW*. 649–658. https://doi.org/10.1145/3178876.3186146
[193] Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. 2016. Collaborative knowledge base embedding for recommender systems. In *Proceedings of the SIGKDD*. 353–362. http://doi.acm.org/10.1145/2939672.2939673
[194] Qi Zhang, Jiawen Wang, Haoran Huang, Xuanjing Huang, and Yeyun Gong. Hashtag recommendation for multimodal microblog using co-attention network. IJCAI, 3420–3426.
[195] Shuai Zhang, Yi Tay, Lina Yao, and Aixin Sun. 2018. Next item recommendation with self-attention. *arXiv preprint arXiv:1808.06414* (2018).
[196] Shuai Zhang, Lina Yao, Aixin Sun, Sen Wang, Guodong Long, and Manqing Dong. 2018. NeuRec: On nonlinear transformation for personalized ranking. *arXiv preprint arXiv:1805.03002* (2018).
[197] Shuai Zhang, Lina Yao, and Xiwei Xu. 2017. AutoSVD++: An efficient hybrid collaborative filtering model via contractive auto-encoders. ACM, 957–960.
[198] Yongfeng Zhang, Qingyao Ai, Xu Chen, and W. Bruce Croft. 2017. Joint representation learning for top-n recommendation with heterogeneous information sources. In *Proceedings of the CIKM*. 1449–1458. http://doi.acm.org/10.1145/3132847.3132892

[199] Xiangyu Zhao, Long Xia, Liang Zhang, Zhuoye Ding, Dawei Yin, and Jiliang Tang. 2018. Deep reinforcement learning for page-wise recommendations. *arXiv preprint arXiv:1805.02343* (2018).
[200] Xiangyu Zhao, Liang Zhang, Zhuoye Ding, Long Xia, Jiliang Tang, and Dawei Yin. 2018. Recommendations with negative feedback via pairwise deep reinforcement learning. *arXiv preprint arXiv:1802.06501* (2018).
[201] Guanjie Zheng, Fuzheng Zhang, Zihan Zheng, Yang Xiang, Nicholas Jing Yuan, Xing Xie, and Zhenhui Li. 2018. DRN: A deep reinforcement learning framework for news recommendation. In *Proceedings of the WWW*. 167–176. https://doi.org/10.1145/3178876.3185994
[202] Lei Zheng, Chun-Ta Lu, Lifang He, Sihong Xie, Vahid Noroozi, He Huang, and Philip S Yu. 2018. MARS: Memory attention-aware recommender system. *arXiv preprint arXiv:1805.07037* (2018).
[203] Lei Zheng, Vahid Noroozi, and Philip S. Yu. 2017. Joint deep modeling of users and items using reviews for recommendation. In *Proceedings of the WSDM*. 425–434. http://doi.acm.org/10.1145/3018661.3018665
[204] Yin Zheng, Cailiang Liu, Bangsheng Tang, and Hanning Zhou. 2016. Neural autoregressive collaborative filtering for implicit feedback. In *Proceedings of the Recsys*. 2–6. http://doi.acm.org/10.1145/2988450.2988453
[205] Yin Zheng, Bangsheng Tang, Wenkui Ding, and Hanning Zhou. 2016. A neural autoregressive approach to collaborative filtering. In *Proceedings of the ICML*. 764–773. https://dl.acm.org/citation.cfm?id=3045390.3045472
[206] Chang Zhou, Jinze Bai, Junshuai Song, Xiaofei Liu, Zhengchao Zhao, Xiusi Chen, and Jun Gao. 2017. ATRank: An attention-based user behavior modeling framework for recommendation. *arXiv preprint arXiv:1711.06632* (2017).
[207] Jiang Zhou, Cathal Gurrin, and Rami Albatal. 2016. Applying visual user interest profiles for recommendation & personalisation. Springer International Publishing, 361–366.
[208] Fuzhen Zhuang, Dan Luo, Nicholas Jing Yuan, Xing Xie, and Qing He. 2017. Representation learning with pair-wise constraints for collaborative ranking. In *WSDM*. 567–575. http://doi.acm.org/10.1145/3018661.3018720
[209] Fuzhen Zhuang, Zhiqiang Zhang, Mingda Qian, Chuan Shi, Xing Xie, and Qing He. 2017. Representation learning via dual-autoencoder for recommendation. *Neural Networks* 90 (2017), 83–89.
[210] Yi Zuo, Jiulin Zeng, Maoguo Gong, and Licheng Jiao. 2016. Tag-aware recommender systems based on deep neural networks. *Neurocomputing* 204 (2016), 51–60.

Received August 2017; revised October 2018; accepted October 2018