# Deep Learning based Recommender System: A Survey and New Perspectives

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



With the ever-growing volume, complexity and dynamicity of online information, recommender system has been an effective key solution to overcome such information overload. In recent years, deep learning's revolutionary advances in speech recognition, image analysis and natural language processing have gained significant attention. Meanwhile, recent studies also demonstrate its effectiveness in coping with information retrieval and recommendation tasks. Applying deep learning techniques into recommender system has been gaining momentum due to its state-of-the-art performances and high-quality recommendations. In contrast to traditional recommendation models, deep learning provides a better understanding of user's demands, item's characteristics and historical interactions between them.

This article aims to provide a comprehensive review of recent research efforts on deep learning based recommender systems towards fostering innovations of recommender system research. A taxonomy of deep learning based recommendation models is presented and used to categorize the surveyed articles. Open problems are identified based on the analytics of the reviewed works and discussed potential solutions.

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

Additional Key Words and Phrases: Recommender System; Deep Learning; Survey

**ACM Reference format:**
Shuai Zhang, Lina Yao, and Aixin Sun. 2017. Deep Learning based Recommender System: A Survey and New Perspectives. *ACM J. Comput. Cult. Herit.* 1, 1, Article 35 (July 2017), 35 pages.
DOI: 0000001.0000001

## 1 INTRODUCTION

The explosive growth of information available online frequently overwhelms users. Recommender system (RS) is a useful information filtering tool for guiding users in a personalized way of discovering products or services they might be interested in from a large space of possible options. Recommender system has been playing a more vital and essential role in various information access systems to boost business and facilitate decision-making process.

In general, the recommendation lists are generated based on user preferences, item features, user-item past interactions and some other additional information such as temporal and spatial data. Recommendation models are mainly categorized into collaborative filtering, content-based recommender system and hybrid recommender system based on the types of input data [1]. However, these models have their own limitations in dealing with data sparsity and cold-start problems, as well as balancing the recommendation qualities in terms of different evaluation metrics[1, 6, 74, 110].

Author's addresses: S. Zhang and L. Yao, School of Computer Science and Engineering, University of New South Wales, Sydney, NSW, 2052; emails: shuai.zhang@student.unsw.edu.au; lina.yao@unsw.edu.au; A. Sun, School of Computer Science and Engineering, Nanyang Technological University, Nanyang Avenue, Singapore 639798, email: axsun@ntu.edu.sg;

DOI: 0000001.0000001

The past few decades have witnessed the tremendous successes of the deep learning (DL) 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 [17]. Recently, deep learning has been revolutionizing the recommendation architectures dramatically and brings more opportunities in reinventing the user experiences for better customer satisfaction. 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 the non-linear and non-trivial 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.

### 1.1 Why should we care about deep learning based recommender system?

Recommender system is an essential part in industry area. It is a critical tool to promote sales and services for many online websites and mobile applications. For instances, 80 percent of movies watched on Netflix came from recommendations [31], 60 percent of video clicks came from home page recommendation in YouTube [20]. Recently, many companies resort to deep learning for further enhancing their recommendation quality [12, 17, 81]. Covington et al. [17] presented a deep neural network based recommendation algorithm for video recommendation on YouTube. Cheng et al. [12] proposed an App recommender system for Google Play with a wide & deep model. Shumpei et al. [81] presented a RNN based news recommender system for Yahoo News. All of these models have stood the 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.

Another noticeable change lies in the research area. The number of research publications on deep learning based recommendation methods has increased exponentially in these years. The leading international conference on recommender system, RecSys[1], started to organize regular workshop on deep learning for recommender system[2] since the year 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 the strength and weakness, and application scenarios of these models.

### 1.2 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 systematic reviews well shaping this area and positioning existing works and current progresses. Although some works have explored the recommender applications built on deep learning techniques and have attempted to formalize this research field, few has 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.

In the last few years, a number of surveys in traditional recommender systems have been presented. For example, Su et al. [100] presented a systematic review on collaborative filtering techniques; Burke et al. [6] proposed a comprehensive survey on hybrid recommender system; Fernández-Tobías et al. [27] and Khan et al. [50] reviewed the cross-domain recommendation models; to name a few. However, there is a lack of extensive

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

review on deep learning based recommender system. To the extent of our knowledge, only two related short surveys [5, 68] are formally published. Betru et al. [5] introduced three deep learning based recommendation models [89, 109, 113], although these three works are influential in this research area, this survey lost sight of other emerging high quality works. Liu et al. [68] reviewed 13 papers on deep learning for recommendation, and proposed to classify 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 works, this classification framework is no longer suitable and a new inclusive framework is required for better understanding of this research field. Given the rising popularity and potential of deep learning applied in recommender system, a systematic survey will be of high scientific and practical values. We analyzed these works from different perspectives and presented some new insights toward this area. To this end, over 100 studies were shortlisted and classified in this survey.

### 1.3 Contributions of this survey

The goal of this survey is to thoroughly review literatures on the advances of deep learning based recommender system. It provides a panorama with 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 system and tap into the richness of this research area. This survey serves the researchers, practitioners, and educators who are interested in recommender system, with particular emphasis on deep learning based recommender system. The key contributions of this survey are three-fold:

- We conduct a systematic review on recommendation models built on deep learning techniques and propose a new classification scheme to position and organize the current works;
- We provide an overview of the state-of-the-art research and summarize their advantages and limitations. The professionals can easily locate the models for a specific problem or find the unsolved problems;
- We discuss the challenges and open issues, and identify the new trends and future directions in this research field to share the visions and expand the horizons of deep learning based recommender system research.

The remaining of this article is organized as follows: Section 2 introduces the basic terminologies and notations used throughout this paper. Section 3 presents our classification framework and performs qualitative analysis to the shortlisted papers. Sections 4 and 5 review the major deep learning based recommendation models following the proposed classification framework. Section 6 discusses the challenges and prominent open research issues. Section 7 concludes the paper.

## 2 TERMINOLOGY AND BACKGROUND CONCEPTS

Before we dive into the details of this survey, we start with an introduction of the basic terminologies and concepts regarding recommender system and deep learning techniques.

### 2.1 Recommender System

Recommender system is used for estimating users' preferences on items they have not seen [1]. There mainly exists three types of recommendation tasks based on the forms of outputs: rating prediction, ranking prediction (top-n recommendation) and classification. Rating prediction aims to fill the missing entries of the user-item rating matrix. Top-n recommendation produces a ranked list with $n$ items to users. Classification task aims at classifying the candidate items into the correct categories for recommendation.

Recommendation models are usually classified into three categories [1]: collaborative filtering, content based and hybrid recommender system. Collaborative filtering makes recommendations by learning from user-item historical interactions, either explicit (e.g. user's previous ratings) or implicit feedback (e.g. browsing histories).

Table 1. Notations and Denotations

| Notation | Description | Notation | Description |
|---|---|---|---|
| $R$ | Rating matrix | $\hat{R}$ | Predicted rating matrix |
| $M$ | Number of users | $N$ | Number of items |
| $r_{ui}$ | Rating of item $i$ given by user $u$ | $\hat{r}_{ui}$ | Predicted rating of item $i$ given by user $u$ |
| $\mathbf{r}^{(i)}$ | Partial observed vector for item $i$ | $\mathbf{r}^{(u)}$ | Partial observed vector for user $u$ |
| $U$ | User latent factor | $V$ | Item latent factor |
| $k$ | The size of latent factor | $K$ | The maximum value of rating (explicit) |
| $\mathcal{O}$, $\mathcal{O}^-$ | Observed, Unobserved rating set | $I$ | Indicator, $I_{ui} = 1$ if $r_{ui} \neq 0$ otherwise $I_{ui} = 0$ |
| $W_*$ | Weight matrices for neural network | $b_*$ | Bias terms for neural network |

Content-based recommendation is mainly based on comparisons across items' and users' auxiliary information. A various range of auxiliary information such as texts, images and videos can be taken into account. Hybrid model refers to recommender system that integrates two or more types of recommendation strategies [6].

Suppose we have $M$ users and $N$ items, and $R$ denotes the rating matrix. Meanwhile, we use a partial observed vector $\mathbf{r}^{(u)} = \{r^{u1}, ..., r^{uN}\}$ to represent each user $u$, and partial observed vector $\mathbf{r}^{(i)} = \{r^{1i}, ..., r^{Mi}\}$ to represent each item $i$. Table 1 summarizes the notations and corresponding descriptions that are used throughout this survey.

## 2.2 Deep Learning Techniques

Deep learning is a sub research field of machine learning. It learns multiple levels of representations and abstractions from data, which can solve both supervised and unsupervised learning tasks [21]. In this subsection, we clarify a diverse array of deep learning concepts closely related to this survey.

- Multilayer Perceptron (MLP) is a feedforward neural network with multiple (one or more) hidden layers between input layer and output layer. Here, the perceptron can employ arbitrary activation function and does not necessarily represent strictly binary classifier.
- 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 denoising autoencoder, marginalized denoising autoencoder, sparse autoencoder, contractive autoencoder and variational autoencoder (VAE) [11, 33].
- Convolutional Neural Network (CNN) [33] is a special kind of feedforward neural network with convolution layers and pooling operations. It is capable of capturing the global and local features and significantly enhancing the efficiency and accuracy. It performs well in processing data with grid-like topology.
- Recurrent Neural Network (RNN) [33] is suitable for modelling sequential data. Unlike 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) network are often deployed in practice to overcome the vanishing gradient problem.
- Deep Semantic Similarity Model (DSSM), or more specifically, Deep Structured Semantic Model [45], is a deep neural network for learning semantic representations of entities in a common continuous semantic space and measuring their semantic similarities.
- Restricted Boltzmann Machine (RBM) is a two layer neural network consisting of a visible layer and a hidden layer. It can be easily stacked to a deep net. *Restricted* here means that there are no intra-layer communications in visible layer or hidden layer.

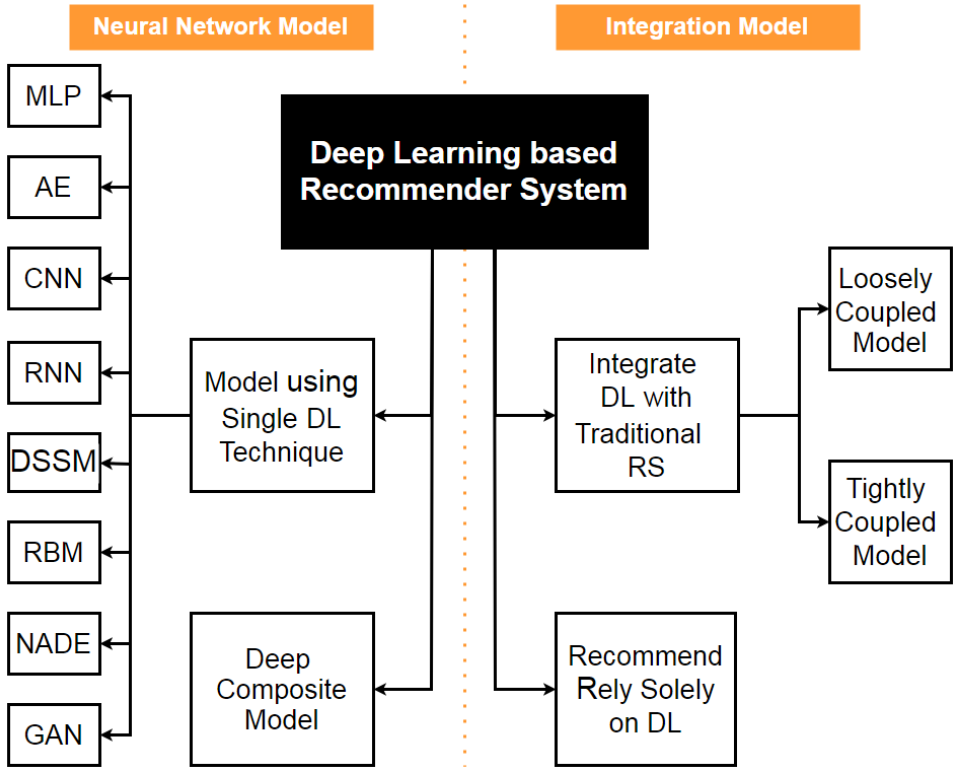


Fig. 1. Two-dimension scheme for classification of deep learning based recommender system. The left part illustrates the neural network models, and the right part illustrates the integration models.

- Neural Autoregressive Distribution Estimation (NADE) [57, 108] is an unsupervised neural network built atop autoregressive model and feedforward neural network. It is a tractable and efficient estimator for modelling data distributions and densities.
- Generative Adversarial Network (GAN) [34] 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.

## 3 CLASSIFICATION SCHEME AND ANALYSIS

In this section, we introduce our two-dimensional scheme for classifying deep learning based recommender system and conduct overall analysis on the listed studies.

### 3.1 Two-dimension Classification Scheme

To give a better birdview of current works, we propose a classification scheme to organize the existing works along two different perspectives (neural network model and integration model). Figure 1 summarizes these two classification criteria.

*3.1.1 Neural Network Model.* We classify the existing studies in accordance with the types of employed deep learning techniques. We divide deep learning based recommendation models into two broad categories: models

Table 2. Classification of shortlisted publications. "*" denotes that there are no papers on that category so far. The number in "()" indicates the total number of publications in that category.

| | | Recommend Rely Solely on Deep Learning | Integrate Deep Learning with Traditional Recommender System | |
|---|---|---|---|---|
| | | | *Tightly Coupled Model* | *Loosely Coupled Model* |
| Model Using Single Deep Learning Technique | *MLP (15)* | [46], [133], [2], [111] [118], [38], [9], [25] [12] | [10], [65], [17], [35] [39] | [66] |
| | *AE (25)* | [82], [90], [99], [129] [98], [135], [144], [102] [84] | [113], [63], [62], [24] [112] , [145], [3], [136] | [139], [122], [123], [106] [7], [146], [107], [22] |
| | *CNN (17)* | [32], [79], [120] | [51], [52], [140], [109] [91], [69] | [15], [93], [143], [73] [36], [37], [124]. [117] |
| | *RNN (22)* | [4], [53], [94], [18] [64], [104], [126], [49] [41], [128], [127], [40] [101], [103], [77], [130] [125], [23] | [19], [105], [81] | [96] |
| | *DSSM (3)* | [26], [8], [132] | * | * |
| | *RBM (7)* | [89], [30], [70], [131] [47] | * | [48], [119] |
| | *NADE (2)* | [142], [141] | * | * |
| | *GAN (1)* | * | [116] | * |
| Deep Composite Model (10) | | [59], [85], [138], [97] [25], [61], [58], [28] | [137], [114] | * |

using **single** deep learning technique and deep **composite** model (recommender system which involves two or more deep learning techniques).

- *Model using Single Deep Learning Technique.* In this category, models are divided into eight subcategories in conformity with the aforementioned eight deep learning models: MLP, AE, CNN, RNN, DSSM, RBM, NADE and GAN based recommender system. The deep learning technique in use determines the strengths and application scenarios of these recommendation models. For instance, MLP can easily model the non-linear interactions between users and items; CNN is capable of extracting local and global representations from heterogeneous data sources such as textual and visual information; RNN enables the recommender system to model the temporal dynamics of rating data and sequential influences of content information; DSSM is able to perform semantic matching between users and items.
- *Deep Composite Model.* Some deep learning based recommendation models utilize more than one deep learning technique. The motivation is that different deep learning techniques can complement one another and enable a more powerful hybrid model. There are many possible combinations of these eight deep learning techniques but not all have been exploited. We list the existing combinations in Section 5. Note that the deep composite model here is different from the hybrid deep networks in [21] which refer to the deep architectures that make use of both generative and discriminative components.

*3.1.2 Integration Model.* The second dimension, integration model, classifies the models by considering whether it integrates traditional recommendation models with deep learning or relies solely on deep learning technique.

- *Integrate Deep Learning with Traditional Recommendation Model.* Some studies attempt to combine deep learning methods with traditional recommendation techniques (e.g. matrix factorization (MF) [56], probabilistic matrix factorization (PMF) [75], factorization machine (FM) [86] or nearest neighbours algorithm etc.) in one way or another. Based on how tightly the two approaches are integrated. These models are further divided into *Loosely Coupled Model* and *Tightly Coupled Model*. For instance, when applying autoencoder to learn feature representations for feeding into latent factor model, we call it tightly coupled model if the parameters of autoencoder and latent factor model are optimized simultaneously. In this way, latent factor model and feature learning process mutually influence each other. If the parameters are learned separately, we name it loosely coupled model instead.
- *Recommend Rely Solely on Deep Learning.* In this case, the training and predicting steps of recommender system solely rely on deep learning techniques without any forms of help from traditional recommendation models.

Table 2 summarizes the surveyed works on the basis of the above classification scheme. Note that, for emerging models such as GAN based recommendation, there are not many works to fill up this table. However, future studies will fall into the proposed classification framework in our expectation.

## 3.2 Qualitative Analysis

Here, we conduct basic statistical analyses on the number of publications, experimental datasets, evaluation metrics and citations of publications.

Figure 2 (a) illustrates the number of yearly publications from the year 2007. The number increased exponentially in the last five years. According to Table 2, we find that AE, RNN, CNN and MLP based recommender systems have been studied widely, followed by deep composite models, RBM and DSSM based models. Recent studies attempt to apply GAN and NADE to recommendation tasks.

Figures 2(c) and 2(d) present the datasets and evaluation metrics used in the reviewed works. Two movie recommendation datasets: Movielens [3] and Netflix remain the most-used datasets. Other datasets such as Amazon [4], Yelp [5] and CiteUlike [6] are also frequently adopted. As for evaluation metrics, Root Mean Square Error (RMSE) and Mean Average Error (MAE) are usually used for rating prediction evaluation, while Recall, Precision, Normalized Discounted Cumulative Gain (NDCG) and Area Under the Curve (AUC) are often adopted for evaluating the ranking scores. Precision, Recall and F1-score are widely used for classification result evaluation.

Table 3 shows the most influential publications [7] with yearly citation greater than 10, From this table, we can clearly identify the most notable deep learning based recommendation models. Nevertheless, other promising models might stand out with time passing.

Another thing we would like to mention is the percentages of works on different recommendation tasks. Ranking prediction has gained the most popularity (66%), followed by the prevailing paradigm, rating prediction (28%), which is also extensively studied, while only few works (6%) convert the recommendation tasks into classification problems.

[3] https://grouplens.org/datasets/movielens/
[4] http://jmcauley.ucsd.edu/data/amazon/links.html
[5] https://www.yelp.com/dataset_challenge
[6] http://www.citeulike.org/faq/data.adp
[7] We collect the citation data from Google Scholar.

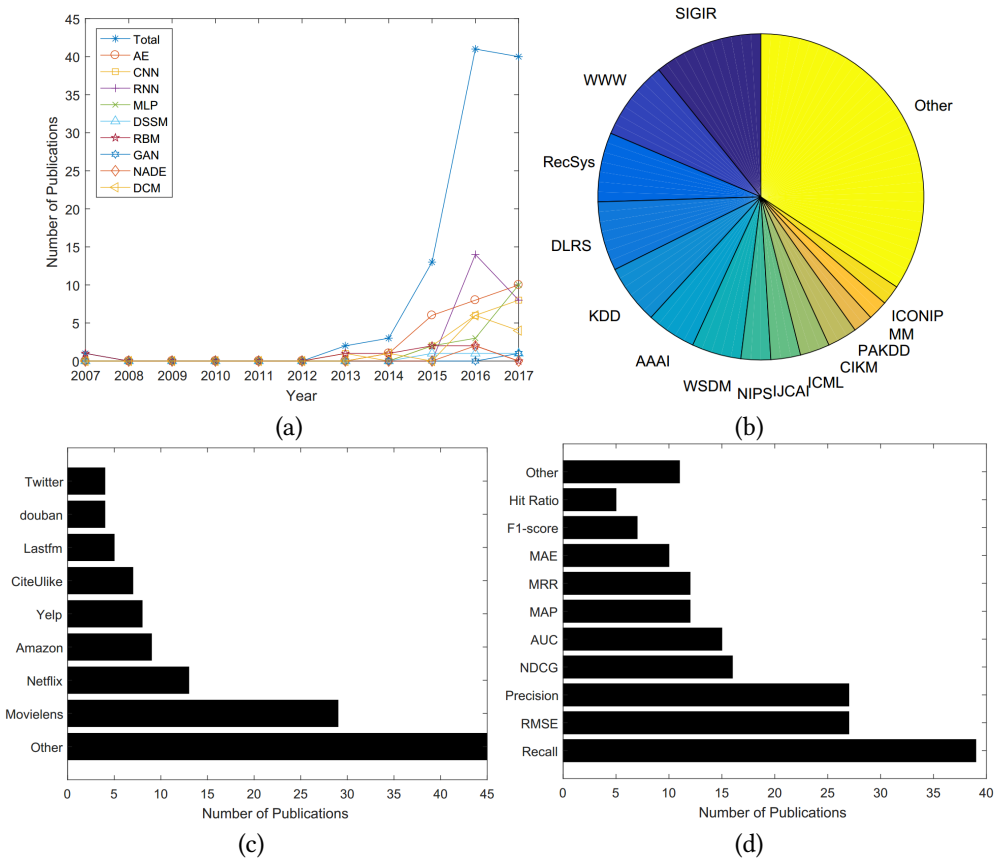


Fig. 2. Illustration of: (a) Number of publications in each year; (b) Venue of publications; (c) Datasets in use; (d) Evaluation metrics in use.

Table 3. Most influential works with yearly citation greater than 10

| Work | DL in use | Yearly Citation | Work | DL in use | Yearly Citation |
|---|---|---|---|---|---|
| Salakhutdinov et al. [89] | RBM | 80 | Wang et al. [113] | AE | 44 |
| Oord et al. [109] | CNN | 44 | McAuley et al. [73] | CNN | 39 |
| Covington et al. [17] | MLP | 22 | Hidasi et al. [40] | RNN | 19 |
| Elkahky et al. [26] | DSSM | 17 | Wu et al. [129] | AE | 14 |
| Wang et al. [119] | RBM | 13 | Sedhain et al. [90] | AE | 12 |
| Cheng et al. [12] | MLP | 12 | He et al. [37] | CNN | 10 |

## 3.3 Application Fields

Among all the reviewed publications, some of the recommendation models have specified their application fields. General recommender system may perform unsatisfactorily in certain domains. These domains usually require to be investigated independently due to their unique characteristics. In this part, we focus on the specific application fields of reviewed publications and catalogue them into: image recommendation, music recommendation, point

of interest (POI) recommendation, news recommendation, hashtag recommendation, quote recommendation and citation recommendation. Table 3 summarizes the corresponding publications.

Table 4. Recommendation models in specific application fields.

| Image | Music | POI | News | HashTag | Quote | Citation |
|---|---|---|---|---|---|---|
| [59, 124] [130] | [49, 109] [66, 119] | [117, 133] [106, 107] | [8, 120] [81, 97] | [85, 138] [32, 112] | [58, 103] | [25, 46] |

## 4 DEEP LEARNING BASED RECOMMENDER SYSTEM

In this section, we highlight important research prototypes within the proposed classification frameworks. We aim to identify the most notable and promising advancements rather than offer an exhaustive list.

### 4.1 Multilayer Perceptron based Recommender System

Multilayer Perceptron is a concise but effective model. As such, it is widely used in many areas, especially in industry areas [12, 17]. Multilayer feedforward networks are demonstrated to be able to approximate any measurable function to any desired degree of accuracy [42]. It is the basis of many advanced models.

#### *4.1.1 Recommend Rely Solely on MLP.*

*Neural Collaborative Filtering.* In most cases, recommendation is considered to be a two-way interaction between users preferences and items features. For example, matrix factorization decomposes the rating matrix into a low-dimensional latent user space and a low-dimensional latent item space. Content-based recommender system generates recommendation lists based on the similarities between user profiles and item features [71]. Thus, it is natural to build a dual network for modelling the two-way interaction between users and items. Neural Collaborative Filtering (NCF) [38] is such a framework aiming to capture the non-linear relationship between users and items. Figure 3(a) presents 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 $i$ respectively. The prediction rule of NCF is formulated as follows:

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

where function $f(\cdot)$ defines the multilayer perceptron, and $\theta$ is the parameters of this network. The loss function for predicting explicit rating is defined as the weighted square error:

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

where $w_{ui}$ denotes the weight of training instance $(u, i)$. For binary ratings or implicit feedback (e.g. *1 represents like* and *0 represents dislike*). The authors proposed a probabilistic approach (e.g. Logistic or Probit function) to constrain the output $\hat{r}_{ui}$ in the range of [0,1], and revised the loss to a cross-entropy form:

$$\mathcal{L} = -\sum_{(u,i) \in \mathcal{O} \cup \mathcal{O}^-} r_{ui} \log \hat{r}_{ui} + (1 - r_{ui}) \log(1 - \hat{r}_{ui}) \tag{3}$$

As there are a large number of unobserved instances, NCF utilizes *negative sampling* to reduce the training data size, which greatly improves the learning efficiency. Traditional matrix factorization can be viewed as a special case of NCF. Therefore, it is convenient to fuse matrix factorization with NCF to formulate a more general model which makes use of both linearity of MF and non-linearity of MLP to enhance recommendation quality.

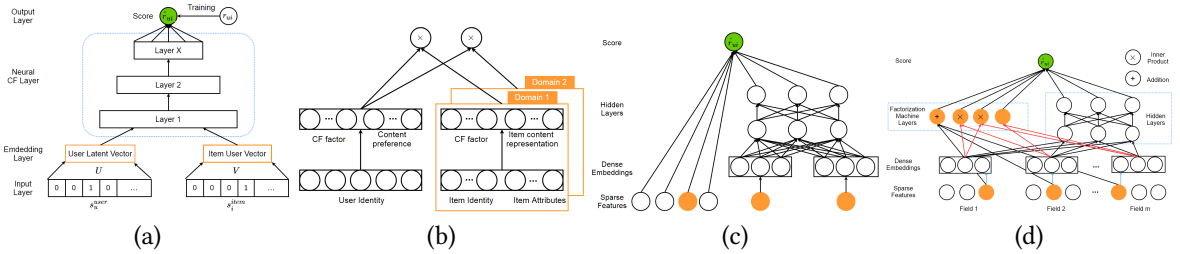


Fig. 3. Illustration of: (a) Neural Collaborative Filtering; (b) CCCFNet; (c) Wide & Deep Learning; (d) DeepFM.

He et al. [118] extended the NCF model to cross-domain social recommendations, i.e., recommending items of information domains to potential users of social networks, and presented a neural social collaborative ranking recommender system. Another extension is CCCFNet (Cross-domain Content-boosted Collaborative Filtering neural Network). The basic composite of CCCFNet is also a dual network (for users and items respectively) [65]. It models the user-item interactions in the last layer with dot product. To embed content information, the authors further decomposed each network of the dual net into two components: collaborative filtering factor (user and item latent factor) and content information (user's preferences on item features and item features). A multi-view neural framework is built on this basic model to perform cross-domain recommendations.

*Wide & Deep Learning*. This general model (shown in Figure 3(c)) can solve both regression and classification problems, but initially introduced for App recommendation in Google play [12]. The wide learning component is a single layer perceptron which can also be viewed as a generalized linear model. The deep learning component is multilayer perceptron. The rationale of combining these two learning techniques is that it enables the recommender system to capture both memorization and generalization. Memorization achieved by the wide learning component represents the capability of catching the direct features from historical data. Meanwhile, the deep learning component catches the generalization by producing more general and abstract representations. This model can improve the accuracy as well as the diversity of recommendation.

Formally, the 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 $l^{th}$ layer, and $f(\cdot)$ is the activation function. $W_{deep}^{(l)}$ and $b^{(l)}$ are weight and bias terms. The wide & deep learning model is attained by fusing these two models:

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

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

By extending this model, Chen et al. [9] devised a locally-connected wide & 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 & deep learning is selecting features for wide and deep parts. In other word, the system should be able to determine which features are memorized or generalized. Moreover, the cross-product transformation also requires to be manually designed. These pre-steps will greatly affect the utility of this model. To alleviate manual efforts in feature engineering, Guo et al. [35] proposed the Deep Factorization Machine (DeepFM).

*Deep Factorization Machine.* DeepFM [35] is an end-to-end model which seamlessly integrates factorization machine and MLP. It is able to model the high-order feature interactions via deep neural network and low-order interactions via factorization machine. Factorization machine utilizes addition and inner product operations to capture the linear and pairwise interactions between features (refer to Equation (1) in [86] for more details). MLP leverages the non-linear activations and deep structure to model the high-order interactions. The way of combining these two models is enlightened by wide & deep network. Compared to wide & deep model, DeepFM does not require tedious feature engineering. It replaces the wide component with a neural interpretation of factorization machine. Figure 3(d) 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 output of FM and MLP are denoted as $y_{FM}(x)$ and $y_{MLP}(x)$ respectively. The prediction score is calculated by:

$$\hat{r}_{ui} = \sigma(y_{FM}(x) + y_{MLP}(x)) \tag{5}$$

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

#### 4.1.2 Integrate MLP with Traditional Recommender System.

*Attentive Collaborative Filtering.* Attention mechanism is motivated by human visual attention. For example, people only need to focus on specific parts of the visual inputs to understand or recognize them. Attention mechanism is capable of filtering out the uninformative features from raw inputs and reduce the side effects of noisy data. Attention-based model has shown promising results on tasks such as speech recognition [13] and machine translation [72]. Recent works [10, 32, 91] demonstrate its capability in enhancing recommendation performance by incorporating it into deep learning techniques (e.g. MLP, CNN and RNN) for recommendation models. Chen et al. [10] proposed an attentive collaborative filtering model by introducing a two-level attention mechanism to latent factor model. The attention model is a MLP consisting 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.

Alashkar et al. [2] proposed a 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. It demonstrates the efficacy of adopting expert knowledge to guide the learning process of the recommendation model in a MLP framework. Although expertise acquisition needs a multitude of human involvements, it is highly precise.

Covington et al. [17] 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 neighbors scores from the candidates. We notice that the industrial world cares more about feature engineering (e.g. transformation, normalization, crossing) and scalability of recommendation models.

### 4.2 Autoencoder based Recommender System

There exist two general ways of applying autoencoder to recommender system: (1) using autoencoder to learn lower-dimensional feature representations at the bottleneck layer; or (2) filling the blanks of rating matrix directly in the reconstruction layer.

#### 4.2.1 Recommend Rely Solely on Autoencoder.

*AutoRec.* AutoRec [90] 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. Apparently, 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

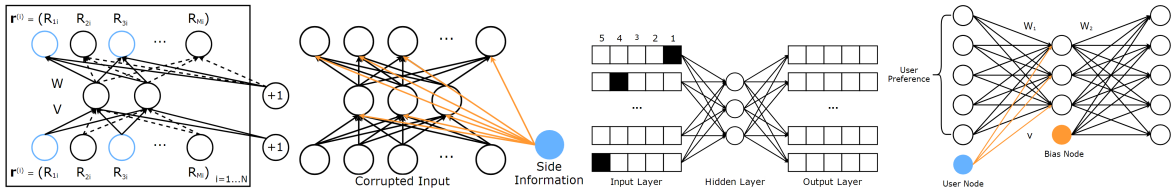


Fig. 4. Illustration of: (a) I-AutoRec; (b) CFN; (c) ACF; (d) CDAE.

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:

$$\arg\min_{\theta} \sum_{i=1}^{N} \| \mathbf{r}^{(i)} - h(\mathbf{r}^{(i)};\theta) \|_{\mathcal{O}}^2 + \lambda \cdot Regularization \tag{6}$$

Here $\| \cdot \|_{\mathcal{O}}^2$ means that it only considers observed ratings. The objective function can be optimized by resilient propagation (converges faster and produces comparable results) or L-BFGS (Limited-memory Broyden Fletcher Goldfarb Shanno algorithm). There are four important points about AutoRec that worth noticing before deployment:

- I-AutoRec performs better than U-AutoRec, which may be due to the higher variance of user partial observed vectors.
- Different combination of activation functions $f(\cdot)$ and $g(\cdot)$ will influence the performance considerably.
- 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.
- Adding more layers to formulate a deep network further improves the performance.

*Collaborative Filtering Neural network (CFN).* CFN [98, 99] is an extension of AutoRec, and posses the following two advantages: (1) it deploys the denoising techniques, which makes CFN more robust; (2) it incorporates the side information such as user profiles and item description to mitigate the sparsity and cold start influence. The input of CFN is also partial observed vectors, so it has two variants: I-CFN and U-CFN, taking $\mathbf{r}^{(i)}$ and $\mathbf{r}^{(u)}$ as input respectively.

The authors introduced three widely used corruption approaches to corrupt the input: Gaussian noise, masking noise and salt-and-pepper noise. To better deal with missing elements (their values are zero), masking noise is imposed as a strong regularizer in CFN [99].

Let $\tilde{\mathbf{r}^{(i)}}$ denotes the corrupted input. The training loss is defined as follows:

$$\mathcal{L} = \alpha\left(\sum_{i \in I(\mathcal{O}) \cap I(\mathcal{C})} [h(\tilde{\mathbf{r}}^{(i)}) - \mathbf{r}^{(i)}]^2\right) + \beta\left(\sum_{i \in I(\mathcal{O}) \setminus I(\mathcal{C})} [h(\tilde{\mathbf{r}}^{(i)}) - \mathbf{r}^{(i)}]^2\right) + \lambda \cdot Regularization \tag{7}$$

In this equation, $I(\mathcal{O})$ and $I(\mathcal{C})$ are the indices of observed and corrupted elements. $\alpha$ and $\beta$ are two hyper-parameters which balance the influence of two components. $h(\tilde{\mathbf{r}}^{(i)})$ represents the reconstruction of corrupted input.

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:

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

Table 5. Comparison of five autoencoder based recommendation models

| Models | Recommendation Task | Input | Corrupted Input | Side Information | Stacked | Pretrained |
|---|---|---|---|---|---|---|
| AutoRec | Rating Prediction | $\mathbf{r}^{(i)}$ or $\mathbf{r}^{(u)}$ | No | No | Yes | No |
| CFN | Rating Prediction | $\mathbf{r}^{(i)}$ or $\mathbf{r}^{(u)}$ | Yes | Yes | Yes | No |
| ACF | Rating Prediction | $\mathbf{r}^{(i)}$ or $\mathbf{r}^{(u)}$ | No | No | Yes | Yes |
| CDAE | Ranking Prediction | $\mathbf{r}^{(u)}$ | Yes | No | No | No |

where $\mathbf{s}_i$ is side information, $\{\tilde{\mathbf{r}}^{(i)}, \mathbf{s}_i\}$ 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.

*Autoencoder-based Collaborative Filtering (ACF).* To the extent of our knowledge, ACF [82] 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 integer in the range of [1-5], each $\mathbf{r}^{(i)}$ will be divided into five partial vectors.

Figure 4(c) presents an example of ACF, where rating scale is [1-5]. The shaded entries indicate that the user has rated that movie as the corresponding rating (e.g. the user gives 1 for item 1, 4 for item 2). Similar to AutoRec and CFN, the cost function of ACF aims at reducing the mean squared error. The rating prediction of ACF is calculated by summarizing each entry of the five vectors, then scaled by the maximum rating $K$ (5 in this example). It uses RBM to pretrain the parameters as well as to avoid local optimum. Stacking several autoencoders together also enhances the accuracy slightly. However, there are two demerits of ACF: (1) it fails to deal with non-integer ratings; (2) the decomposition of partial observed vectors increases the sparseness of input data and leads to worse prediction accuracy.

*Collaborative Denoising Auto-Encoder (CDAE).* The three models reviewed ealier are mainly designed for rating prediction, while CDAE [129] is principally used for ranking prediction. The input of CDAE is user partial observed implicit feedback $\mathbf{r}^{(u)}_{pref}$. The entry value is 1 if the user likes the movie, otherwise 0. It can also be regarded as a preference vector which reflects user's interests to items. Figure 4(d) 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:

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

where $V_u \in \mathbb{R}^K$ denotes the weight matrix for user node (see figure 4(d)). This weight matrix is unique for each user and proven to greatly improve the performance. Parameters are learned by minimizing the reconstruction error:

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

The loss function $\ell(\cdot)$ can be square loss or logistic loss. It adopts $\ell_2$ norm rather than Frobenius norm to regularize both weight and bias terms.

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.

*4.2.2 Integrate Autoencoder with Traditional Recommender System.*

Tightly coupled model learns the parameters of autoencoder and recommender model simultaneously, which enables recommender model to provide guidance for autoencoder to learn more semantic features. Loosely coupled model is performed in two steps: learning salient feature representations via autoencoders, and then feeding these feature representations to recommender system. Both forms have their own strengths and shortcomings. For example, tightly coupled model requires carefully design and optimization to avoid the local optimum, but recommendation and feature learning can be performed at once; loosely coupled method can be easily extended to existing advanced models, but they require more training steps.

**I. Tightly Coupled Model.**

*Collaborative Deep Learning (CDL).* CDL [113] is a hierarchical Bayesian model which integrates stacked denoising autoencoder (SDAE) into probabilistic matrix factorization. To seamlessly combine deep learning and recommendation model, the authors proposed a general Bayesian deep learning framework [115] consisting of two tightly hinged components: perception component (deep neural network) and 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 ratings. 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: $V_i = \epsilon_i + X^T_{\frac{L}{2},i*}$.
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 biases vector for layer $l$, $X_l$ represents layer $l$. $\lambda_w$, $\lambda_s$, $\lambda_n$, $\lambda_v$, $\lambda_u$ are hyper-parameters, $C_{ui}$ is a confidence parameter for measuring the confidence to observations [44]. Figure 5(a, 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 [115] to avoid the local optimum.

Before CDL, Wang et al. [112] proposed a similar model, relational stacked denoising autoencoders (RSDAE), for tag recommendation. The difference of CDL and RSDAE is that RSDAE replaces the PMF with a relational information matrix. Another extension of CDL is collaborative variational autoencoder (CVAE) [63], 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 [136] is devised specifically in a pairwise framework for top-n recommendation. CDL is a point-wise model initially proposed for rating prediction. However, studies have demonstrated that pairwise model is more suitable for ranking lists generation [87, 129, 136]. Experimental results also demonstrated that CDR outperformed CDL in terms of ranking prediction. Figure 5(a, right) presents the structure of CDR. The first and second generative process steps of CDR are the same with CDL. The third and fourth steps are replaced by the following step:

3. 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 pair-wise preference $(i, j) \in P_i$, where $P_i = \{(i, j) : r_{ui} - r_{uj} > 0\}$, draw the

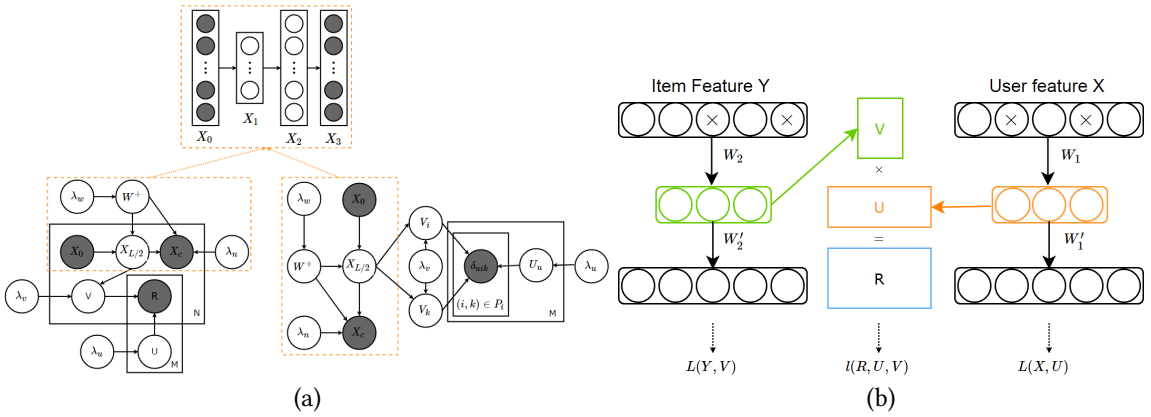


Fig. 5. Illustration of: (a) Graphical model of collaborative deep learning (left) and collaborative deep ranking (right); (b) Deep collaborative filtering framework.

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 $j$, $C_{uij}^{-1}$ 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.* It is a general framework for unifying deep learning approaches with collaborative filtering model [62]. This framework makes it more easily to utilize deep feature learning techniques to aid collaborative recommendation. The aforementioned works such as [109, 113, 119] can be easily interpreted into this general framework. Formally, the deep collaborative filtering framework is formulated as follows:

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

where $\beta$, $\gamma$ and $\delta$ are trade-off parameters to balance the influences of these three components, $X$ and $Y$ are side information, $\ell(\cdot)$ 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 model (mDA-CF). Compared to CDL, mDA-CF explores a more computationally efficient variants of autoencoder: marginalized denoising autoencoder [11]. It saves the computational costs for searching sufficient corrupted version of input by marginalizing out the corrupted input, which makes mDA-CF more scalable than CDL. In addition, mDA-CF embeds content information of items and users while CDL only considers the effects of item features.

**II. Loosely Coupled Model.**

AutoSVD++ [139] makes use of contractive autoencoder [88] to learn item feature representations, then integrates them into the classic recommendation model, SVD++ [54]. The proposed model posses the following advantages: (1) compared to other autoencoders variants, contractive autoencoder captures the infinitesimal input variations; (2) it models the implicit feedback to further enhance the accuracy; (3) an efficient training algorithm is designed to reduce the training time.

HRCD [122, 123] is a hybrid collaborative model based on autoencoder and timeSVD++ [55]. It is a time-aware model which uses SDAE to learn item representations from raw features and aims at solving the cold item problem. However, the similarity based method for cold item recommendation is computationally expensive

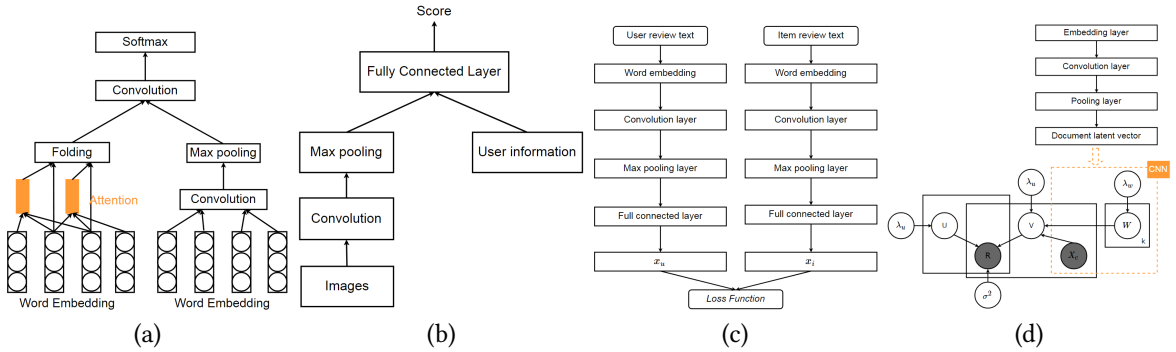


Fig. 6. Illustration of: (a) Attention based CNN; (b) Personalized CNN tag recommendation; (c) DeepCoNN ; (d)ConvMF.

## 4.3 Convolutional Neural Network based Recommender System

Convolution Neural Network is powerful in processing visual, textual and audio information. Most of the CNN based recommender systems utilize CNN for feature extraction.

### 4.3.1 *Recommend Rely Solely on CNN.*

*Attention based CNN.* Gong et al. [32] proposed an attention based CNN system for hashtag recommendation in microblog. It treats hashtag recommendation as a multi-class 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 $h$ and threshold $\eta$ to select informative words (known as trigger words in this work). Hence, only trigger words are at play in the following layers. Here, we emphasize some details about the local attention channel. Let $w_{i:i+h}$ denote the concatenation of words $w_i$, $w_{i+1}$,...,$w_{i+h}$. The score of the central word ($w_{(2i+h-1)/2}$) in the window is calculated by:

$$s_{(2i+h-1)/2} = g(W * w_{i:i+h} + b) \tag{12}$$

where $W$ is the weight matrix, $g(\cdot)$ is the non-linear activation. Only the words with score greater than $\eta$ are kept. Figure 6(a) illustrates the attention-based CNN model. The left and right parts represent the local attention channel and CNN global channel respectively. In the follow-up work [91], Seo et al. made use of two neural networks same as [32] (without the last two layers) to learn feature representations from user and item review texts, and predict rating scores with dot product in the final layer.

*Personalized CNN Tag Recommendation.* Nguyen et al. [79] proposed a personalized tag recommender system based on CNN. Figure 6(b) presents the overall architecture. It utilizes convolutional and max-pooling layer to get visual features from patches of images. User information is injected for generating personalized recommendation. To optimize this network, a Bayesian Personalized Ranking (BPR) algorithm [87] is adopted to maximize the differences between the relevant and irrelevant tags.

### 4.3.2 *Integrate CNN with Traditional Recommender System.*

Same as AE, CNN can also be incorporated into traditional models. However, the existing integration types are not as abundant as those built on AE. The former reviewed AE based models should have suggested some possible extensions for CNN.

**I. Tightly Coupled Model.**

*Deep Cooperative Neural Network (DeepCoNN).* DeepCoNN [140] adopts two parallel convolutional neural networks to model user behaviors and item properties from review texts. In the final layer, the factorization machine is applied to capture their interactions for rating prediction. It alleviates the sparsity problem and

enhances the model interpretability by exploiting rich semantic representations of review texts. It utilizes a word embedding technique to map the review texts into a lower-dimensional semantic spaces as well as keep the words sequences information. The extracted review representations then pass through a convolutional layer with different kernels, a max-pooling layer, and a full-connected layer consecutively. The output of the user network $x_u$ and item network $x_i$ are finally concatenated as the input of factorization machine.

*ConvMF.* ConvMF [51] combines CNN with PMF in a similar way as CDL. CDL uses autoencoder to learn the item feature representations, while ConvMF employs CNN to learn high level item representations. The main advantage of ConvMF over CDL is that CNN is able to capture more accurate contextual information of item via world embedding and convolutional kernels. The graphical model of ConvMF is shown in Figure 6(d). The probabilistic model can be easily formulated by replacing $X^T_{\frac{L}{2},i*}$ in CDL with the output of CNN $cnn(W, X_i)$, where $W$ is the parameters of CNN. The objective function of ConvMF is defined as:

$$\mathcal{L} = \sum_i^N \sum_u^M \frac{I_{ui}}{2}(R_{ui} - U_u^T V_i)^2 + \frac{\lambda_u}{2}\sum_u^M \parallel U_u \parallel + \frac{\lambda_v}{2}\sum_i^N \parallel V_i - cnn(W, X_i) \parallel + \lambda_W \cdot Regularization \tag{13}$$

where $\lambda_u$, $\lambda_v$ and $\lambda_W$ are hyper-parameters. $U$ and $V$ are learned by coordinate descent. The parameters of CNN are obtained by fixing $U$ and $V$.

**II. Loosely Coupled Model.** We classify the loosely coupled models into the following three categories according to the feature types CNN dealing with.

*CNN for Image Feature Extraction.* Wang et al. [117] investigated the influences of visual features to Point-of-Interest (POI) recommendation, and proposed a visual content enhanced POI recommender system (VPOI). VPOI adopts CNN to extract image features. The recommendation model is built on PMF by exploring the interactions between: (1) visual content and latent user factor; (2) visual content and latent location factor. Chu et al. [15] 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 joint with the text representation are input into MF, BPRMF and FM to test their effects. Results show that visual information improves the performance to some degree but not significant. He et al. [37] designed a visual Bayesian personalized ranking (VBPR) algorithm by incorporating visual features (learned via CNN) into matrix factorization. He et al. [36] extended VBPR with exploring user's fashion awareness and the evolution of visual factors that user considers when selecting items.

*CNN for Audio Feature Extraction.* Van et al. [109] proposed using CNN 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. Let $y'$ denote the output of CNN, $y$ denote the item latent factors learned from weighted matrix factorization. The aim of CNN is to minimize the square error between $y$ and $y'$.

*CNN for Text Feature Extraction.* Shen et al. [93] built an e-learning resources recommendation model. It uses CNN to extract item features from text information of learning resources such as introduction and content of learning material, and follows the same procedure of [109] to perform recommendation.

## 4.4 Recurrent Neural Network based Recommender System

Recurrent neural network is specifically suitable for coping with the temporal dynamics of ratings and sequential features in recommender system.

### *4.4.1 Recommend Rely Solely on RNN.*

*Session-based Recommendation with RNN.* In many real word applications or websites, the system usually do not bother users to log in so that it has no access to user's long period consumption habits or long-term interests. However, the session or cookie mechanisms enable those systems to get user's short term preferences. This is

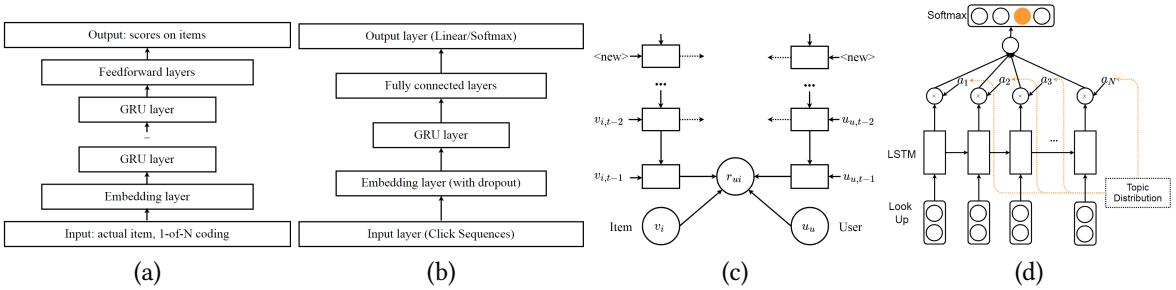


Fig. 7. Illustration of: (a) Session-based recommendation with RNN; (b) Improved session-based recommendation with RNN; (c) Recurrent recommender network; (d) Attention based RNN for tag recommendation.

a relatively unappreciated task in recommender system due to the extreme sparsity of training data. Recent advancements have demonstrated the efficacy of RNN in solving this issue [40, 104, 128]. Hidasi et al. [40] proposed a session-based recommendation model based GRU (shown in Figure 7(a)). The input is the actual state of 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 is formulated as follows:

$$\mathcal{L}_s = \frac{1}{S}\sum_{j=1}^{S}\sigma(\hat{r}_{sj} - \hat{r}_{si}) + \sigma(\hat{r}_{sj}^2) \tag{14}$$

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 $s$, $\sigma$ is the logistic sigmoid function. The last term is used as a regularization.

The follow-up work [104] proposed several strategies to further improve this model:

- Augment the click sequences with sequence preprocessing and dropout regularization;
- Adapt to temporal changes by pre-training with full training data and fine-tuning the model with more recent click-sequences;
- Distillation the model with *privileged information* with a teacher model;
- Using item embedding to decrease the number of parameters for faster computation.

Wu et al. [128] designed a session-based recommendation model for real-world e-commerce website. It utilizes the basic RNN to predict what user will buy next based on the click histories. To minimize the computation costs, it only keeps a finite number of the latest states while collapsing the old states into a single history state. This method helps to balance the trade-off between computation costs and prediction accuracy.

The aforementioned three session-based models do not consider any side information. Two extensions [41, 94] demonstrate that side information have effect on enhancing session recommendation quality. Hidasi et al. [41] 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 weightedly concatenated and fed into a non-linear activation to predict the next items in that session. Smirnova et al. [94] proposed a context-aware session-based recommender system based on conditional RNN. It injects context information into input and output layers. Experimental results of these two models indicate that models incorporated additional information outperform those solely based on historical interactions.

*Recurrent Recommender Network (RRN).* RRN [127] is a non-parametric recommendation model built on RNN. It is capable of modelling the seasonal evolutions of items and changes of user preferences over time. RRN

uses two LSTM networks as the building block to model dynamic user state $u_{ut}$ and item state $v_{it}$. In the meantime, considering the 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:

$$\hat{r}_{ui|t} = f(u_{ut}, v_{it}, u_u, v_i) \tag{15}$$

where $u_{ut}$ and $v_{it}$ are learned from LSTM, $u_u$ 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. [126] further improved the RRN model by modelling text reviews and ratings simultaneously. Unlike most text review enhanced recommendation models [91, 140], this model aims to generate reviews by 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 can not generate coherent and readable review texts. The deep composite model NRT [61] which will be introduced in Section 5 can generate readable review tips.

*Neural Survival Recommender.* Jing et al. [49] proposed a multi-task 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 survival of patients. The authors modified this model by using LSTM to estimate the returning time of costumers. The item recommendation is also performed via LSTM from user's past session actions. Unlike aforementioned session-based recommendations which focus on recommending in the same session, this model aims to provide inter-session recommendations. The objective function for NSR is defined as:

$$\textbf{arg min} \sum_u (1-\alpha)\mathcal{L}_{\text{sur}}(S_u, T) + \alpha\mathcal{L}_{\text{rec}}(S_u, T) \tag{16}$$

where $T$ denotes time point, $S_u$ denotes past session actions for user $u$, $\mathcal{L}_{\text{sur}}$ and $\mathcal{L}_{\text{rec}}$ are loss functions for survival and recommendation model respectively, $\alpha$ acts as a controller for balancing these two components.

*Attention based RNN.* We have discussed the attention based MLP and CNN. Similarly, attention mechanism can also be applied to RNN based recommendation. Li et al. [64] proposed such an attention-based LSTM model for hashtag recommendation. This work takes the advantages of both RNN and attention mechanism to capture the sequential property and recognize the informative words from microblog posts. First, the model uses LSTM to learn hidden states $[h_1, h_2, h_3, ..., h_N]$ for microblog posts. Meanwhile, a topic model, LDA, is employed to learn the posts' topic distribution. The topic attention $[a_1, a_2, a_3, ..., a_N]$ is attained from this distribution after a series of non-linear transformation and softmax normalization. The output after attention filtering is $vec = \sum_{j=1}^{P} a_j h_j$, where $P$ is the length of the posts. This model is trained by minimizing the cross-entropy.

#### 4.4.2 Integrate RNN with Traditional Recommender System.

*GRU Multitask Learning.* Bansal et al. [4] proposed to use GRU to encode the text sequences into latent factor model: $\hat{r_{ui}} = b_i + b_u + U_u^T f(X_i)$, where $b_i$ and $b_u$ are the bias for items and users, $X_i$ is the text embedding of item content information:

$$f(X_i) = g(X_i) + \widetilde{V}_i \tag{17}$$

where function $g(X_i)$ represents GRU for modelling word order information, $\widetilde{V}_i$ is an item specific embedding for capturing behaviors that can not be modelled by content information. This hybrid model solves both warm-start and cold-start problems. In case of cold-start item, $V_i$ is simply set to 0. Furthermore, the authors adopted a multi-task 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.

Dai et al. [19] presented a co-evolutionary latent model to capture the co-evolution nature of users' and items' latent features. The interactions between users and items play an important role in driving the changes of user

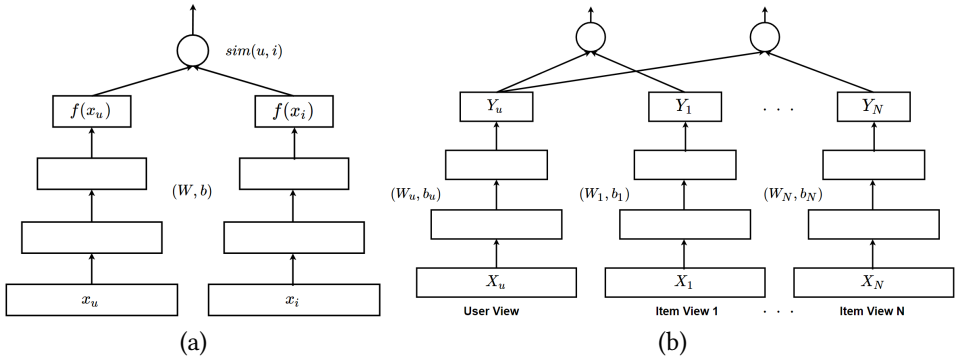


Fig. 8. Illustration of: (a) Deep semantic similarity based personalized recommendation; (b) Multi-view deep neural network.

preferences and item status. To model the historical interactions, the author proposed using RNN to automatically learn representations of the influences from drift, evolution and co-evolution of user and item features.

Okura et al. [81] proposed to use GRU to learn more expressive aggregation for user browsing histories, and recommend news articles with 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 serving over ten million unique users everyday.

### 4.5 Deep Semantic Similarity Model based Recommender System

Deep Semantic Similarity Model (DSSM) [45] is a deep neural network widely used in information retrieval area. It is supremely suitable for top-n recommendation [26, 132]. DSSM projects different entities into a common low-dimensional space, and computes their similarities with cosine function. Basic DSSM is made up of MLP, while more advanced neural layers such as convolution and max-pooling layers can be easily added in.

*Deep Semantic Similarity based Personalized Recommendation (DSPR).* DSPR [132] shown in Figure 8(a) is a tag-aware personalized recommender system where each user $x_u$ and item $x_i$ are represented by tag annotations and mapped into a common tag space. Cosine similarity and softmax function are applied to decide the relevance of items and users (or user's preference to item):

$$sim(u,i) = cosine(f(x_u), f(x_i)) = \frac{f(x_u)^T f(x_i)}{\| f(x_u) \| \cdot \| f(x_i) \|} \tag{18}$$

where $f(\cdot)$ denotes the feedforward neural network. The similarity is transformed into probabilistic form by softmax. Here, $u$ and $i$ are very much alike to the query and document corpus in web search problem. $P(i|u)$ measures the probability of user $u$ interacting with (e.g. browsing, buying, etc.) item $i$.

$$P(i|u) = \frac{exp(\gamma \cdot sim(u,i))}{\sum_{i' \in I} exp(\gamma \cdot sim(u,i'))} \tag{19}$$

Here $\gamma$ is a smoothing factor of softmax function. The loss function of DSPR is:

$$\mathcal{L} = -\log \prod_{(u,i^*)} P(i^*|u) \tag{20}$$

where $(u, i^*)$ is the training set consisting of pairs of users and the items they like and some randomly sampled items they dislike.

*Multi-View Deep Neural Network (MV-DNN).* MV-DNN [26] is designed for cross domain recommendation. It treats users as the pivot view and each domain (suppose we have $Z$ domains) as auxiliary view. Apparently, there

are $Z$ similarity scores for $Z$ user-domain pairs. MV-DNN is similar to the former reviewed MLP based model, CCCFNet, but CCCFNet does not involve any similarity and posterior probability estimation.

$$\mathcal{L} = \arg\min_{\theta} \sum_{j=1}^{Z} \frac{exp(\gamma \cdot cosine(Y_u, Y_{a,j}))}{\sum_{X' \in R^{da}} exp(\gamma \cdot cosine(Y_u, f_a(X')))} \tag{21}$$

where $\theta$ is the model parameters, $\gamma$ is the smoothing factor, $Y_u$ is the output of user view, $a$ is the index of active view. $R^{da}$ 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 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.

### 4.6 Restricted Boltzmann Machine based Recommender System

*Restricted Boltzmann Machine Collaborative Filtering (RBM-CF).* Salakhutdinov et al. [89] proposed a restricted Boltzmann machine based recommender system. To the best of our knowledge, it is the first recommendation model that built atop deep learning. 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 4 to this item. Let $h_j, j = 1, ..., F$ denote the hidden units with fixed size $F$. Each user has a unique RBM with shared parameters. Suppose a user rated $m$ movies, the number of visible units is $m$, Let $X$ be a $K \times m$ matrix where $x_i^y = 1$ if user $u$ rated movie $i$ as $y$ and $x_i^y = 0$ otherwise. Then:

$$p(v_i^y = 1|h) = \frac{exp(b_i^y + \sum_{j=1}^{F} h_j W_{ij}^y)}{\sum_{l=1}^{K} exp(b_i^l + \sum_{j=1}^{F} h_j W_{ij}^l)} \quad , \quad p(h_j = 1|X) = \sigma(b_j + \sum_{i=1}^{m} \sum_{y=1}^{K} x_i^y W_{ij}^y) \tag{22}$$

where $W_{ij}^y$ represents the weight on the connection between the rating $y$ of movie $i$ and the hidden unit $j$, $b_i^y$ is the bias of rating $y$ for movie $i$, $b_j$ is the bias of hidden unit $j$. RBM is not tractable, but the parameters can be learned via the Contrastive Divergence (CD) algorithm [33].

The authors further proposed using a conditional RBM to incorporate the implicit feedback. Here, let a binary vector $t$ of length $M$ represent the implicit feedback ($S$). The entry will be 1 if the user rated that movie, otherwise 0. The essence here is that users implicitly tell their preferences by giving ratings, regardless of how they rate items (same with the implicit feedback in SVD++ [54]). Then the conditional probability of hidden units will be:

$$p(h_j = 1|X, S) = \sigma(b_j + \sum_{i=1}^{m} \sum_{k=1}^{K} x_i^k W_{ij}^k + \sum_{i=1}^{M} t_i D_{ij}) \tag{23}$$

where $D_{ij}$ is the weight matrix which controls the influence of $t$ to the hidden units. Figure 9(b) illustrates the conditional RBM-CF.

The above two reviewed models are user-based RBM-CF models which clamp a given user's ratings on the visible layer. Similarity, we can easily design an item-based RBM-CF if we clamp a given item's ratings on the visible layer. Georgiev et al. [30] proposed to combine the user-based and item-based RBM-CF in a unified framework. In the case, the visible units are determined both by user and item hidden units. Let $x_{ui}$ be the visible unit for user $u$ on item $i$. The reconstruction value is given by the following formula:

$$x_{ui} = \frac{1}{2}(\underbrace{b_u^U + \sum_{p=1}^{F^U} h_{up}^U W_{up}^U}_{\text{user-based RBM}} + \underbrace{b_i^I + \sum_{q=1}^{F^I} h_{iq}^I W_{iq}^I}_{\text{item-based RBM}}) \tag{24}$$

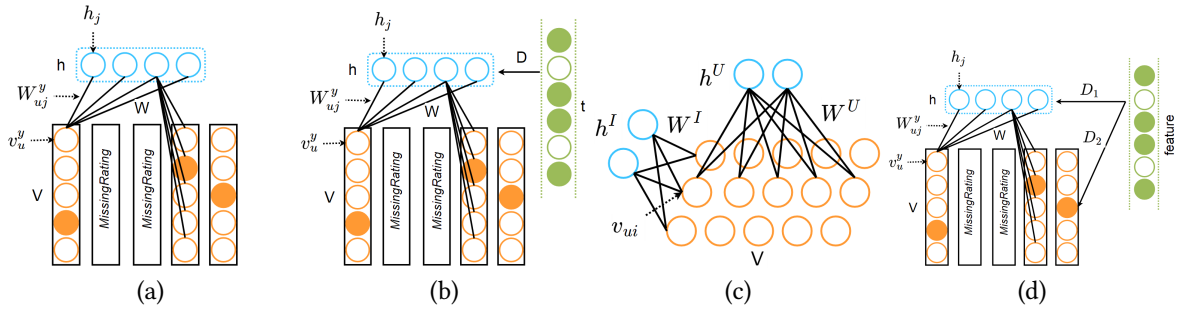


Fig. 9. Illustration of: (a) RBM-CF; (b) Conditional RBM-CF with implicit feedback; (c) Combination of user-based and item-based RBM-CF; (d) Hybrid RBM-CF.

The equation consists of the user-based and item-based influences. We eliminate the superscript $K$ for simplicity. Figure 9(c) shows the combined model.

*Hybrid RBM-CF.* Liu et al. [70] designed a hybrid RBM-CF which incorporates item features (item categories) into RBM collaborative filtering. This model is also based on conditional RBM. There are two differences between this hybrid model with the conditional RBM-CF with implicit feedback: (1) the conditional layer here is modelled with the binary item genres; (2) the conditional layer affects both the hidden layer and the visible layer with different connected weights. The model formulation can be easily formulated.

$$p(v_u^y = 1|h) = \frac{exp(b_u^y + \sum_{j=1}^{F} h_j W_{uj}^y + \sum_{q=1}^{C} f_q D'_{qu})}{\sum_{l=1}^{K} exp(b_u^l + \sum_{j=1}^{F} h_j W_{uj}^l + \sum_{q=1}^{C} f_q D'_{qu})} \quad , \quad p(h_j = 1|X, S) = \sigma(b_j + \sum_{i=1}^{m}\sum_{y=1}^{K} x_i^y W_{ij}^y + \sum_{q=1}^{C} f_q D_{qj}) \tag{25}$$

where $f_q$ is the genre feature vector, $C$ is the size of total feature sets, $D$ and $D'$ are connected weight matrices for hidden units and visible units. The parameters can also be learned via CD algorithm.

## 4.7 Emerging Methods: NADE and GAN

Here, we elaborate two emerging methods: NADE and GAN based recommender systems. NADE presents a tractable method for approximating the real distribution of source data and produces state-of-the-art recommendation accuracy in terms of rating prediction (compared with other deep learning based recommendation models) on several experimental datasets; GAN is capable of fusing discriminative model with generative model together and posses the advantages of these two schools of thinking.

*4.7.1 Neural AutoRegressive based Recommender System.* As mentioned above, RBM is not tractable, thus we usually use the Contrastive Divergence algorithm to approximate the log-likelihood gradient on the parameters [57], which also limits the usage of RBM-CF. The so-called Neural Autoregressive Distribution Estimator (NADE) is a tractable distribution estimator which provides a desirable alternative to RBM. Inspired by RBM-CF, Zheng et al. [142] proposed a NADE based collaborative filtering model (CF-NADE). CF-NADE models the distribution of user ratings. Here, we present a detailed example shown in Figure 10(a) to illustrate how the CF-NADE works.

Suppose we have 4 movies: 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:

$$p(\mathbf{r}) = \prod_{i=1}^{D} p(r_{m_{o_i}} | \mathbf{r}_{m_{o_{<i}}}) \tag{26}$$

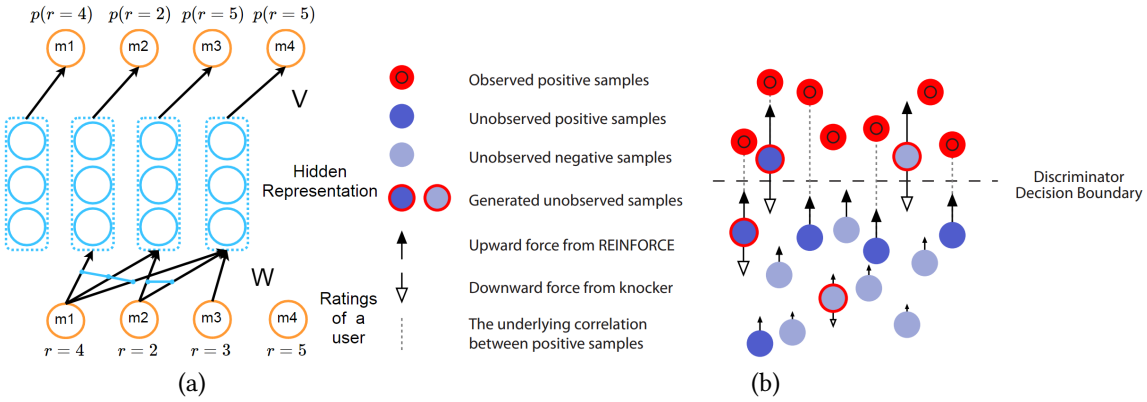


Fig. 10. Illustration of: (a) Neural autoregressive based recommender system; (b) IRGAN.

where $D$ is the number of items that the user has rated, $o$ is the $D$-tuple in the permutations of $(1, 2, ..., D)$, $m_i$ is the index of the $i^{th}$ rated item, $r_{m_{o_i}}$ is the rating that the user gives to item $m_{o_i}$.

More specifically, the procedure goes as follows: (1) the probability that the user gives $m1$ 4-star conditioned on nothing; (2) the probability that the user gives $m2$ 2-star conditioned on giving $m1$ 4-star; (3) the probability that the user gives $m3$ 3-star conditioned on giving $m1$ 4-star and $m2$ 2-star; (4) the probability that the user gives $m4$ 5-star conditioned on giving $m1$ 4-star, $m2$ 2-star and $m3$ 3-star.

Ideally, the order of movies should follow the time-stamps of ratings. However, 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. [141] proposed to incorporate implicit feedback to overcome the sparsity problem of rating matrix.

*4.7.2 Generative Adversarial Network based Recommender System.* IRGAN [116] is the first model which applies GAN to 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.

Firstly, we introduce the general framework of IRGAN. Traditional GAN consists of a discriminator and a generator. Likely, there are two schools of thinking in information retrieval, that is, 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 labelled relevant query-document pairs. The aim of IRGAN is to combine these two thoughts into a unified model, and make them to play a minimax game like 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 documents and non-relevant documents. Similar to the objective function of GAN, the overall objective is formulated as follows:

$$J^{G^*,D^*} = \min_\theta \max_\phi \sum_{n=1}^{N} (\mathbb{E}_{d\sim p_{true}(d|q_n,r)}[logD(d|q_n)] + \mathbb{E}_{d\sim p_\theta(d|q_n,r)}[log(1 - D(d|q_n))]) \quad (27)$$

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

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

$$p_\theta(d_i|q, r) = \frac{exp(g_\theta(q, d_i))}{\sum_{d_j} exp(g_\theta(q, d_j))} \tag{28}$$

$g_\theta(q, d)$ is the chance of document $d$ being generated from query $q$. In 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 modelled them with the same function for convenience, and define 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 $s(\cdot)$.

$$s(u, i) = b_i + U_u^T V_i \tag{29}$$

where $b_i$ is the bias for item $i$, $U_u$ and $V_i$ are the latent vectors for user $u$ and item $i$. It can be substituted with other advanced models such as factorization machine or neural network.

These two approaches both show promising results but are largely underexplored. We suggest that follow-up works investigate more on incorporating abundant auxiliary source information to these two models, or integrate them to other powerful deep composite models.

## 5 DEEP COMPOSITE MODELS FOR RECOMMENDATION

As this survey involves eight types of deep learning techniques, if we define the deep composite models by the combination of any arbitrary two techniques, then we have $C_8^2 = 28$ forms of composite models. There will be more if combinations with three or four deep learning techniques are allowed. Despite the abundant possible ways of combination, every deep composite model should be reasonable and carefully designed for the specific tasks. In this section, we summarize the existing models that has been proven to be effective in some application fields. Figure 11 illustrates the existing combinations of deep composite models.

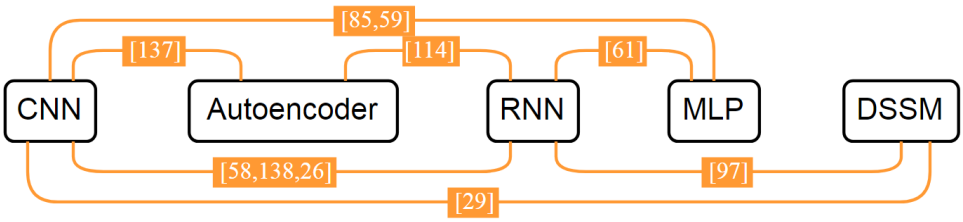


Fig. 11. Existing Deep composite models

To put more emphasis on the ways of combination, the following text is organized according to the ways of combination.

### 5.1 CNN and Autoencoder

Collaborative Knowledge Based Embedding (CKE) [137] combines CNN with 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 movies), 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 the TransR [67], 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 auto-encoders (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.

### 5.2 CNN and RNN

Lee et al. [58] proposed a composite model with RNN and CNN for quotes 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 CNN 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. The overall architecture is shown in Figure 12(a).

Zhang et al. [138] proposed a CNN and RNN composite model for hashtag recommendation. Given a tweet with corresponding images, the authors utilized CNN 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. [25] presented a neural citation network which integrates CNN with RNN in a encoder-decoder framework for citation recommendation. In this model, CNN acts as the encoder that captures the long-term dependencies from citation context. The RNN works 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 CNN.

### 5.3 CNN and MLP

Lei et al. [59] proposed a comparative deep leaning model combining CNN and MLP for image recommendation. This network consists of two CNNs which are used for image representation learning and a MLP for user preferences modelling. It compares two images (one positive image user likes and one negative image user dislikes) against a user. The training data is made up of triplets: $t$ (user $U_t$, positive image $I_t^+$, negative image $I_t^-$). Assuming 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). For generating better ranking lists, it maximizes the differences between negative images and positive images:

$$\mathop{\mathbf{arg\,min}}_{\pi,\phi} \sum_t -\mathcal{I} \log[\sigma(p)] - (1-\mathcal{I}) \log[1-\sigma(p)] \tag{30}$$

where $p = D(\pi(U_t), \phi(i)) - D(\pi(U_t), \phi(j))$, $\pi$ and $\phi$ correspond to MLP and CNN mapping, $\sigma$ is the Simoid function. $\mathcal{I}$ is a indicator, where $\mathcal{I} = 0$ if $(i = I_t^+, j = I_t^-)$, and $\mathcal{I} = 1$ if $(i = I_t^-, j = I_t^+)$. Figure 12(b) illustrates the overall architecture of comparative deep learning model. We omit some details for simplicity.

ConTagNet [85] is a context-aware tag recommender system. The image features are learned by CNN. The context representations are processed by a two layers fully-connected feedforward neural network. The outputs of two neural networks are concatenated and fed into a softmax normalization for predicting the probability of candiate tags.

### 5.4 RNN and Autoencoder

The former mentioned collaborative deep learning model is lack of robustness and incapable of modelling the sequences of text information. Wang et al. [114] further exploited integrating RNN and denoising autoencoder to overcome this limitations. The authors first designed a generalization of RNN named 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

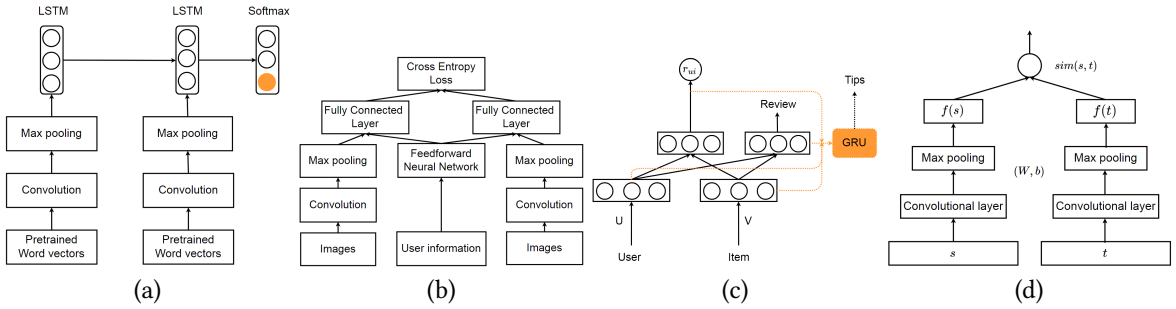


Fig. 12. Illustration of: (a) Quote recommendation with CNN and RNN; (b) Comparative deep learning model; (c) NRT; (d) DSSM with CNN.

RNN, which enables CRAE to capture the sequential information of item content information. Furthermore, the authors designed a wildcard denoising and a bet-pooling technique to prevent the model from overfitting.

## 5.5 RNN and MLP

NRT. [61] is a composite model which combines MLP and RNN in a multitask learning framework (shown in Figure 12(c)). NRT is capable of predicting ratings as well as generating textual tips for users simultaneously. The generated tips provide concise suggestions and anticipate user's experience and feelings on certain products. The rating prediction task is modelled by MLP with non-linear transformation 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 network for tips generation. Here, $r_{ui}$ is used as context information to decide the sentiment of the generated tips. The multi-task learning framework enables the whole model to be trained efficiently in an end-to-end paradigm.

## 5.6 CNN and DSSM

Gao et al. [28] proposed an interest-aware recommender system with DSSM and CNN to recommend users target document ($t$) according to the source documents ($s$) they are reading. Unlike the traditional DSSM and DSPR, the authors add convolutional and max pooling operations to the hidden layers. The convolutional and max-pooling layers are used to capture the local and global features respectively. Another important change is that, instead of maximizing the likelihood, the authors present a pairwise training technique. It computes the difference of interestingness scores, $\Delta = sim(s, t_1) - sim(s, t_2)$, where $t_1$ is preferred and has higher interestingness score then $t_2$, and then maximizes $\Delta$. These two strategies lead to great improvement over traditional DSSM.

## 5.7 RNN and DSSM

Song et al. [97] designed a temporal DSSM model which integrates RNN into DSSM for recommendation. Based on traditional DSSM, TDSSM replace the left network with item static features, and the right network with two sub-networks to modelling user static features (with MLP) and user temporal features (with RNN). The loss function of TDSSM is formulated as follows:

$$L(\Theta) = -\log \prod_{(u,i^*)} \frac{exp(sim(E(u,t_i), E_{i^+}))}{\sum_{\forall item} exp(sim(E(u,t_i), E_i))} \tag{31}$$

where $t_i$ represents the time spot $i$. The incorporation of temporal dynamics increases the model parameters, so the author proposed to use pre-training to speedup the parameters learning process.

## 6 FUTURE RESEARCH DIRECTIONS AND OPEN ISSUES

Current deep learning technologies help establish a solid foundation to enhance recommender system. The expected large number of various deep learning techniques to be deployed on the recommender system would trigger a significant paradigm shift from traditional recommender system to deep recommender system. Here, we identify several emerging research trends for the future evolution of deep learning based recommendation.

### 6.1 Deep Understanding of Users and Items

Making accurate recommendations requires deep understanding of item characteristics and user's real demands [1, 60]. This can be achieved by exploiting the abundant auxiliary information. For example, context information tailors services and products according to user's circumstances and surroundings [107], and mitigate cold start influence; Implicit feedback indicates users' implicit intention and is easier to collect while gathering explicit feedback is a resource-demanding task. Although existing works have investigated the efficacy of deep learning model in mining user and item profiles [65, 139], implicit feedback [37, 136, 139, 141], contextual information [25, 51, 85, 105, 107], and review texts [61, 91, 126, 140] for recommendation, they do not utilize these various side information in a comprehensive manner and 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 [43] and physical world (e.g. Internet of things) [134]. One can infer user's temporal interests or intentions from these side data resources while deep learning method is a desirable and powerful tool for integrating these additional information. The capability of deep learning in processing heterogeneous data sources also brings more opportunities in recommending diverse items with unstructured data such as textual, visual, audio and video features.

Another aspect of deep understanding towards users and items is through feature engineering. Feature engineering has not been fully studied in the recommendation research community, but it is essential and widely employed in industrial applications [12, 17]. However, most of the existing models require manually crafted and selected features, which is time-consuming and tedious. Deep neural network is a promising tool for feature crafting by reducing manual intervention [92]. More intensive studies on deep feature engineering specific for recommender system are expected to save human efforts as well as improve recommendation quality.

### 6.2 Deep Composite Models

As we indicate in Sections 2 and 5, deep composite model combining multiple deep neural networks enables a more powerful tool for modelling the heterogeneous characteristics of the determining factors (e.g. user, item, context, etc.) in recommender system. There already have been a few studies which integrate different deep learning techniques together for enhancing performances. Nevertheless, the attempts are limited compared to the possible extensions. For example, autoencoder could be fused with DSSM to capture the similarities of the salient features. Note that one needs to follow the principle that the model should be designed in a sensible way rather than arbitrarily and tailored for practical requirements.

### 6.3 Temporal Dynamics

The first promising extension on temporal dynamics in recommendation is session modelling. Although session-based recommender system is not a novel research topic, it is largely underinvestigated. Compared to traditional recommendation based on static preferences, session-based recommender system is more suitable for capturing dynamic and temporal user demands. Tracking user's long term interaction is inapplicable for many websites and mobile applications [40], but short term session can be usually collected and utilized for aiding the recommendation process. In recent years, RNN has demonstrated its superiority in session modelling [40, 104]. Extension works

such as incorporating auxiliary information to RNN or applying other deep learning models to session-based recommendation can be explored for future improvement.

Second, the evolution and co-evolution of items and users are also important aspects of temporal influences. As user and item features can evolve independently and co-evolve dependently over time [19, 121]. For instance, app changes along time with newer versions in the domain of app recommendation. Deep sequential model would be an ideal tool for modelling these evolutionary and coevolutionary influences and reflecting the underlying temporal nature.

### 6.4 Cross Domain Recommendation

Nowadays, many large companies offer diversified products or services to customers. For example, Google provides us with web searches, mobile applications and news services; We can buy books, electronics and clothes from Amazon. Single domain recommender system only focuses on one domain while ignores the user interests on other domains, which also exacerbates sparsity and cold start problems [50]. 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 [27, 83]. Deep learning is well suited to transfer learning as it learn high-level abstractions that disentangle the variation of different domains. Several existing works [26, 65] indicate the efficacy of deep learning in catching the generalizations and differences across different domains and generating better recommendations on cross-domain platforms. Therefore, it is a promising but largely under-explored area where mores studies are expected.

### 6.5 Multi-Task Learning

Multi-task learning has led to successes in many deep learning tasks, from computer vision to natural language processing [16, 21]. Among the reviewed studies, several works [4, 49, 61, 135] also applied multi-task learning to recommender system in a deep neural framework and achieved some improvements over single task learning. The advantages of applying deep neural network based multi-task learning are three-fold: (1) learning several tasks at a time can prevent overfitting by generalizing the shared hidden representations; (2) auxiliary task provides interpretable output for explaining the recommendation; (3) multi-task provides an implicit data augmentation for alleviating the sparsity problem. Multitask can be utilized in traditional recommender system [80], while deep learning enables them to be integrated in a tighter fashion. Apart from introducing side tasks, we can also deploy the multitask learning for cross domain recommendation with each specific task generating recommendation for each domain.

### 6.6 Attention Mechanism

Attention mechanism is an intuitive but effective technique, which can be applied to MLP, RNN, CNN and many other deep neural networks. For example, integrating attention mechanism into RNN enables the RNN to process long and noisy inputs [14]. Although LSTM can solve the long memory problem theoretically, it is still problematic when dealing with long-range dependencies. Attention mechanism provides a better solution and helps the network to better memorize inputs. Attention-based CNN is capable of capturing the most informative elements of the inputs [91]. By applying attention mechanism to recommender system, one could leverage attention mechanism to filter out uninformative content and select the most representative items [10] while providing good interpretability.

### 6.7 Scalability

The increasing data volumes in the big data era poses 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 demonstrated to be very effective and promising in big data analytics [78]. However, more future works should be studied on how to recommend efficiently by exploring the following problems: (1) incremental learning for non-stationary and streaming data such as large volume of incoming users and items; (2) computation efficiency for high-dimensional tensors and multimedia data sources; (3) balancing of the model complexity and scalability with the exponential growth of parameters.

### 6.8 Novel Evaluation Metrics

Most former researches focus on accuracy improvement, either aims at improving the recall/precision or reducing the prediction error. However, being accurate is far from enough for a high-quality recommender system in the long term [74], and can even lead to over-specialization. Apart from accuracy, other evaluation metrics such as, diversity [3, 102], novelty, serendipity, coverage, trustworthiness, privacy, interpretability etc. also matter [29, 50, 76, 110]. Recommender system will bring customers with unclear and uncertain intents more values by encouraging diversity and serendipity [95]; increasing the trustworthiness and privacy will release user's worries and give them more freedom in exploring their interested items; good interpretability provides evidence to support each recommendation [31] and presents more convincing results for users. Therefore, recommender system should not just perform accurate historical modelling, but offer a holistic experience to the users.

Table 6 outlines the existing works on the basis of aforementioned research directions. Despite several attempts having been made, they are not fully explored and more studies are needed.

Table 6. Existing works on the mentioned promising research directions

| Session-based Recommender System | Cross-domain Recommender System | Multi-Task Learning | Attention Mechanism | Novel Evaluation Metrics |
|---|---|---|---|---|
| [40, 41, 49, 104, 105] | [26, 65, 118] | [4, 49, 61, 126, 135] | [32, 64, 120, 138] [10, 91] | [3, 102] (Diversity) [22] (Coverage) |

## 7 CONCLUSION

In this article, we provided an extensive review of the most notable works to date on deep learning based recommender system. We proposed a two dimensional classification scheme for organizing and clustering existing publications. We also conduct a brief statistical analysis on these works to identify the contributions and characteristics of these studies. We highlight a bunch of influential research prototypes and analyze their advantages/disadvantages and applicable scenarios. Moreover, we detail some of the most pressing open problems and promising future extensions. Both deep learning and recommender system are ongoing hot research topics in the recent decades. There are a large number of new developing techniques and emerging models each year, here, we provide an inclusive framework for comprehensive understanding towards the key aspects of this field, clarify the most notable advancements 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 *AAAI*. 941–947.

[3] 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).

[4] 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*. ACM, 107–114.

[5] 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), 17–22. https://doi.org/10.5120/ijca2017913361

[6] Robin Burke. 2002. Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction* 12, 4 (2002), 331–370.

[7] S. Cao, N. Yang, and Z. Liu. 2017. Online news recommender based on stacked auto-encoder. In *2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS)*. 721–726. https://doi.org/10.1109/ICIS.2017.7960088

[8] 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.

[9] Cen Chen, Peilin Zhao, Longfei Li, Jun Zhou, Xiaolong Li, and Minghui Qiu. 2017. Locally Connected Deep Learning Framework for Industrial-scale Recommender Systems. In *Proceedings of the 26th International Conference on World Wide Web Companion (WWW '17 Companion)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 769–770. https://doi.org/10.1145/3041021.3054227

[10] 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. *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval* (2017).

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

[12] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. 2016. Wide & deep learning for recommender systems. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*. ACM, 7–10.

[13] 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).

[14] 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*. 577–585.

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

[16] 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*. ACM, 160–167.

[17] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM, 191–198.

[18] 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).

[19] Hanjun Dai, Yichen Wang, Rakshit Trivedi, and Le Song. 2016. Recurrent coevolutionary latent feature processes for continuous-time recommendation. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*. ACM, 29–34.

[20] 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 Fourth ACM Conference on Recommender Systems (RecSys '10)*. ACM, New York, NY, USA, 293–296. https://doi.org/10.1145/1864708.1864770

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

[22] 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.

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

[24] 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 *AAAI*. 1309–1315.
[25] Travis Ebesu and Yi Fang. 2017. Neural Citation Network for Context-Aware Citation Recommendation. *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval* (2017).
[26] Ali Mamdouh Elkahky, Yang Song, and Xiaodong He. 2015. A multi-view deep learning approach for cross domain user modeling in recommendation systems. In *Proceedings of the 24th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 278–288.
[27] Ignacio Fernández-Tobías, Iván Cantador, Marius Kaminskas, and Francesco Ricci. 2012. Cross-domain recommender systems: A survey of the state of the art. In *Spanish Conference on Information Retrieval*. 24.
[28] 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.
[29] Mouzhi Ge, Carla Delgado-Battenfeld, and Dietmar Jannach. 2010. Beyond accuracy: evaluating recommender systems by coverage and serendipity. In *Proceedings of the fourth ACM conference on Recommender systems*. ACM, 257–260.
[30] Kostadin Georgiev and Preslav Nakov. 2013. A non-iid framework for collaborative filtering with restricted boltzmann machines. In *International Conference on Machine Learning*. 1148–1156.
[31] Carlos A Gomez-Uribe and Neil Hunt. 2016. The netflix recommender system: Algorithms, business value, and innovation. *ACM Transactions on Management Information Systems (TMIS)* 6, 4 (2016), 13.
[32] Yuyun Gong and Qi Zhang. 2016. Hashtag Recommendation Using Attention-Based Convolutional Neural Network.. In *IJCAI*. 2782–2788.
[33] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press. http://www.deeplearningbook.org.
[34] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems*. 2672–2680.
[35] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: A Factorization-Machine based Neural Network for CTR Prediction. In *IJCAI*. 2782–2788.
[36] 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 25th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 507–517.
[37] Ruining He and Julian McAuley. 2016. VBPR: Visual Bayesian Personalized Ranking from Implicit Feedback. In *AAAI*. 144–150.
[38] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 173–182.
[39] Xiangnan He and Chua Tat-Seng. 2017. Neural Factorization Machines for Sparse Predictive Analytics. *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval* (2017).
[40] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2015. Session-based recommendations with recurrent neural networks. *International Conference on Learning Representations* (2015).
[41] 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 10th ACM Conference on Recommender Systems*. ACM, 241–248.
[42] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. 1989. Multilayer feedforward networks are universal approximators. *Neural networks* 2, 5 (1989), 359–366.
[43] 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 25th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 51–62.
[44] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative Filtering for Implicit Feedback Datasets. In *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining (ICDM '08)*. IEEE Computer Society, Washington, DC, USA, 263–272. https://doi.org/10.1109/ICDM.2008.22
[45] 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 22nd ACM international conference on Conference on information & knowledge management*. ACM, 2333–2338.
[46] Wenyi Huang, Zhaohui Wu, Liang Chen, Prasenjit Mitra, and C Lee Giles. 2015. A Neural Probabilistic Model for Context Based Citation Recommendation. In *AAAI*. 2404–2410.
[47] X. Jia, X. Li, K. Li, V. Gopalakrishnan, G. Xun, and A. Zhang. 2016. Collaborative restricted Boltzmann machine for social event recommendation. In *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. 402–405. https://doi.org/10.1109/ASONAM.2016.7752265
[48] 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 *2015 IEEE International Conference on Big Data (Big Data)*. IEEE, 837–842.
[49] How Jing and Alexander J Smola. 2017. Neural survival recommender. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. ACM, 515–524.

[50] Muhammad Murad Khan, Roliana Ibrahim, and Imran Ghani. 2017. Cross Domain Recommender Systems: A Systematic Literature Review. *ACM Comput. Surv.* 50, 3, Article 36 (June 2017), 34 pages. https://doi.org/10.1145/3073565
[51] Donghyun Kim, Chanyoung Park, Jinoh Oh, Sungyoung Lee, and Hwanjo Yu. 2016. Convolutional matrix factorization for document context-aware recommendation. In *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM, 233–240.
[52] 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).
[53] Young-Jun Ko, Lucas Maystre, and Matthias Grossglauser. 2016. Collaborative recurrent neural networks for dynamic recommender systems. In *Asian Conference on Machine Learning*. 366–381.
[54] Yehuda Koren. 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 426–434.
[55] Yehuda Koren. 2010. Collaborative filtering with temporal dynamics. *Commun. ACM* 53, 4 (2010), 89–97.
[56] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009).
[57] Hugo Larochelle and Iain Murray. 2011. The neural autoregressive distribution estimator. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. 29–37.
[58] 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 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 957–960.
[59] 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.
[60] Jure Leskovec. 2015. New Directions in Recommender Systems. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining (WSDM '15)*. ACM, New York, NY, USA, 3–4. https://doi.org/10.1145/2684822.2697044
[61] Piji Li, Zihao Wang, Zhaochun Ren, Lidong Bing, and Wai Lam. 2017. Neural Rating Regression with Abstractive Tips Generation for Recommendation. *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval* (2017).
[62] Sheng Li, Jaya Kawale, and Yun Fu. 2015. Deep collaborative filtering via marginalized denoising auto-encoder. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. ACM, 811–820.
[63] Xiaopeng Li and James She. 2017. Collaborative Variational Autoencoder for Recommender Systems. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. ACM.
[64] Yang Li, Ting Liu, Jing Jiang, and Liang Zhang. 2016. Hashtag recommendation with topical attention-based LSTM. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics*.
[65] 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 26th International Conference on World Wide Web Companion*. International World Wide Web Conferences Steering Committee, 817–818.
[66] Dawen Liang, Minshu Zhan, and Daniel PW Ellis. 2015. Content-Aware Collaborative Music Recommendation Using Pre-trained Neural Networks.. In *ISMIR*. 295–301.
[67] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning Entity and Relation Embeddings for Knowledge Graph Completion. In *AAAI*. 2181–2187.
[68] Juntao Liu and Caihua Wu. 2017. *Deep Learning Based Recommendation: A Survey*. Springer Singapore, Singapore, 451–458. https://doi.org/10.1007/978-981-10-4154-9_52
[69] Qiang Liu, Shu Wu, and Liang Wang. 2017. DeepStyle: Learning User Preferences for Visual Recommendation. *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval* (2017). https://doi.org/10.1145/3077136.3080658
[70] Xiaomeng Liu, Yuanxin Ouyang, Wenge Rong, and Zhang Xiong. 2015. Item Category Aware Conditional Restricted Boltzmann Machine Based Recommendation. In *International Conference on Neural Information Processing*. Springer, 609–616.
[71] Pasquale Lops, Marco De Gemmis, and Giovanni Semeraro. 2011. Content-based recommender systems: State of the art and trends. In *Recommender systems handbook*. Springer, 73–105.
[72] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025* (2015).
[73] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 43–52.
[74] Sean M. McNee, John Riedl, and Joseph A. Konstan. 2006. Being Accurate is Not Enough: How Accuracy Metrics Have Hurt Recommender Systems. In *CHI '06 Extended Abstracts on Human Factors in Computing Systems (CHI EA '06)*. ACM, New York, NY, USA, 1097–1101. https://doi.org/10.1145/1125451.1125659
[75] Andriy Mnih and Ruslan R Salakhutdinov. 2008. Probabilistic matrix factorization. In *Advances in neural information processing systems*. 1257–1264.

[76] Bamshad Mobasher, Robin Burke, Runa Bhaumik, and Chad Williams. 2007. Toward trustworthy recommender systems: An analysis of attack models and algorithm robustness. *ACM Transactions on Internet Technology (TOIT)* 7, 4 (2007), 23.
[77] 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 *IIR*.
[78] 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.
[79] 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, Cham, 186–197. https://doi.org/10.1007/978-3-319-57454-7_15
[80] Xia Ning and George Karypis. 2010. Multi-task learning for recommender system. In *Proceedings of 2nd Asian Conference on Machine Learning*. 269–284.
[81] Shumpei Okura, Yukihiro Tagami, Shingo Ono, and Akira Tajima. 2017. Embedding-based News Recommendation for Millions of Users. In *Proceedings of the 23th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM.
[82] Yuanxin Ouyang, Wenqi Liu, Wenge Rong, and Zhang Xiong. 2014. Autoencoder-based collaborative filtering. In *International Conference on Neural Information Processing*. Springer, 284–291.
[83] Weike Pan, Evan Wei Xiang, Nathan Nan Liu, and Qiang Yang. 2010. Transfer Learning in Collaborative Filtering for Sparsity Reduction. In *AAAI*, Vol. 10. 230–235.
[84] Yiteng Pana, Fazhi Hea, and Haiping Yua. 2017. Trust-aware Collaborative Denoising Auto-Encoder for Top-N Recommendation. *arXiv preprint arXiv:1703.01760* (2017).
[85] 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, 1102–1106.
[86] S. Rendle. 2010. Factorization Machines. In *2010 IEEE International Conference on Data Mining*. 995–1000. https://doi.org/10.1109/ICDM.2010.127
[87] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*. AUAI Press, 452–461.
[88] Salah Rifai, Pascal Vincent, Xavier Muller, Xavier Glorot, and Yoshua Bengio. 2011. Contractive auto-encoders: Explicit invariance during feature extraction. In *Proceedings of the 28th international conference on machine learning (ICML-11)*. 833–840.
[89] Ruslan Salakhutdinov, Andriy Mnih, and Geoffrey Hinton. 2007. Restricted Boltzmann machines for collaborative filtering. In *Proceedings of the 24th international conference on Machine learning*. ACM, 791–798.
[90] Suvash Sedhain, Aditya Krishna Menon, Scott Sanner, and Lexing Xie. 2015. Autorec: Autoencoders meet collaborative filtering. In *Proceedings of the 24th International Conference on World Wide Web*. ACM, 111–112.
[91] 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 *3rd International Workshop on Machine Learning Methods for Recommender Systems (MLRec)(SDM'17)*.
[92] Ying Shan, T Ryan Hoens, Jian Jiao, Haijing Wang, Dong Yu, and JC Mao. 2016. Deep Crossing: Web-scale modeling without manually crafted combinatorial features. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 255–262.
[93] Xiaoxuan Shen, Baolin Yi, Zhaoli Zhang, Jiangbo Shu, and Hai Liu. 2016. Automatic Recommendation Technology for Learning Resources with Convolutional Neural Network. In *International Symposium on Educational Technology (ISET)*. IEEE, 30–34.
[94] Elena Smirnova and Flavian Vasile. 2017. Contextual Sequence Modeling for Recommendation with Recurrent Neural Networks. (2017).
[95] Brent Smith and Greg Linden. 2017. Two Decades of Recommender Systems at Amazon. com. *IEEE Internet Computing* 21, 3 (2017), 12–18.
[96] 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*. ACM, 589–593.
[97] Yang Song, Ali Mamdouh Elkahky, and Xiaodong He. 2016. Multi-rate deep learning for temporal recommendation. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 909–912.
[98] 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*. ACM, 11–16.
[99] Florian Strub and Jeremie Mary. 2015. Collaborative Filtering with Stacked Denoising AutoEncoders and Sparse Inputs. In *NIPS Workshop on Machine Learning for eCommerce*.
[100] Xiaoyuan Su and Taghi M Khoshgoftaar. 2009. A survey of collaborative filtering techniques. *Advances in artificial intelligence* 2009 (2009), 4.
[101] 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, 202–211.

[102] Yosuke Suzuki and Tomonobu Ozaki. 2017. Stacked Denoising Autoencoder-Based Deep Collaborative Filtering Using the Change of Similarity. In *Advanced Information Networking and Applications Workshops (WAINA), 2017 31st International Conference on*. IEEE, 498–502.

[103] 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, 65–74.

[104] Yong Kiam Tan, Xinxing Xu, and Yong Liu. 2016. Improved recurrent neural networks for session-based recommendations. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*. ACM, 17–22.

[105] Bartlomiej Twardowski. 2016. Modelling Contextual Information in Session-Aware Recommender Systems with Neural Networks. In *Proceedings of the 10th ACM Conference on Recommender Systems (RecSys '16)*. ACM, New York, NY, USA, 273–276.

[106] Moshe Unger. 2015. Latent Context-Aware Recommender Systems. In *Proceedings of the 9th ACM Conference on Recommender Systems*. ACM, 383–386.

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

[108] 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.

[109] Aaron Van den Oord, Sander Dieleman, and Benjamin Schrauwen. 2013. Deep content-based music recommendation. In *Advances in neural information processing systems*. 2643–2651.

[110] Saúl Vargas and Pablo Castells. 2011. Rank and relevance in novelty and diversity metrics for recommender systems. In *Proceedings of the fifth ACM conference on Recommender systems*. ACM, 109–116.

[111] Jeroen B. P. Vuurens, Martha Larson, and Arjen P. de Vries. 2016. Exploring Deep Space: Learning Personalized Ranking in a Semantic Space. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems (DLRS 2016)*. ACM, New York, NY, USA, 23–28. https://doi.org/10.1145/2988450.2988457

[112] Hao Wang, Xingjian Shi, and Dit-Yan Yeung. 2015. Relational Stacked Denoising Autoencoder for Tag Recommendation.. In *AAAI*. 3052–3058.

[113] Hao Wang, Naiyan Wang, and Dit-Yan Yeung. 2015. Collaborative deep learning for recommender systems. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1235–1244.

[114] Hao Wang, SHI Xingjian, and Dit-Yan Yeung. 2016. Collaborative recurrent autoencoder: Recommend while learning to fill in the blanks. In *Advances in Neural Information Processing Systems*. 415–423.

[115] Hao Wang and Dit-Yan Yeung. 2016. Towards Bayesian deep learning: A framework and some existing methods. *IEEE Transactions on Knowledge and Data Engineering* 28, 12 (2016), 3395–3408.

[116] Jun Wang, Lantao Yu, Weinan Zhang, Yu Gong, Yinghui Xu, Benyou Wang, Peng Zhang, and Dell Zhang. 2017. IRGAN: A Minimax Game for Unifying Generative and Discriminative Information Retrieval Models. *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval* (2017).

[117] 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 26th International Conference on World Wide Web (WWW '17)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 391–400.

[118] Xiang Wang, Xiangnan He, Liqiang Nie, and Tat-Seng Chua. 2017. Item Silk Road: Recommending Items from Information Domains to Social Users. *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval* (2017).

[119] Xinxi Wang and Ye Wang. 2014. Improving content-based and hybrid music recommendation using deep learning. In *Proceedings of the 22nd ACM international conference on Multimedia*. ACM, 627–636.

[120] 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 Editorsfi Demonstration. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. ACM.

[121] Yichen Wang, Nan Du, Rakshit Trivedi, and Le Song. 2016. Coevolutionary latent feature processes for continuous-time user-item interactions. In *Advances in Neural Information Processing Systems*. 4547–4555.

[122] 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 *Dependable, Autonomic and Secure Computing, 14th Intl Conf on Pervasive Intelligence and Computing, 2nd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech), 2016 IEEE 14th Intl C*. IEEE, 874–877.

[123] 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.

[124] Jiqing Wen, Xiaopeng Li, James She, Soochang Park, and Ming Cheung. 2016. Visual background recommendation for dance performances using dancer-shared images. In *2016 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. IEEE, 521–527.

[125] 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.

[126] Chao-Yuan Wu, Amr Ahmed, Alex Beutel, and Alexander J Smola. 2016. Joint Training of Ratings and Reviews with Recurrent Recommender Networks. *Workshop track, ICLR 2017* (2016).

[127] Chao-Yuan Wu, Amr Ahmed, Alex Beutel, Alexander J Smola, and How Jing. 2017. Recurrent recommender networks. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. ACM, 495–503.

[128] Sai Wu, Weichao Ren, Chengchao Yu, Gang Chen, Dongxiang Zhang, and Jingbo Zhu. 2016. Personal recommendation using deep recurrent neural networks in NetEase. In *2016 IEEE 32nd International Conference on Data Engineering (ICDE)*. IEEE, 1218–1229.

[129] Yao Wu, Christopher DuBois, Alice X Zheng, and Martin Ester. 2016. Collaborative denoising auto-encoders for top-n recommender systems. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*. ACM, 153–162.

[130] Ruobing Xie, Zhiyuan Liu, Rui Yan, and Maosong Sun. 2016. Neural Emoji Recommendation in Dialogue Systems. *arXiv preprint arXiv:1612.04609* (2016).

[131] Weizhu Xie, Yuanxin Ouyang, Jingshuai Ouyang, Wenge Rong, and Zhang Xiong. 2016. User Occupation Aware Conditional Restricted Boltzmann Machine Based Recommendation. In *2016 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. IEEE, 454–461.

[132] 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 25th ACM International on Conference on Information and Knowledge Management*. ACM, 1921–1924.

[133] Carl Yang, Lanxiao Bai, Chao Zhang, Quan Yuan, and Jiawei Han. 2017. Bridging Collaborative Filtering and Semi-Supervised Learning: A Neural Approach for POI Recommendation. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM.

[134] 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.

[135] Baolin Yi, Xiaoxuan Shen, Zhaoli Zhang, Jiangbo Shu, and Hai Liu. 2016. Expanded autoencoder recommendation framework and its application in movie recommendation. In *Software, Knowledge, Information Management & Applications (SKIMA), 2016 10th International Conference on*. IEEE, 298–303.

[136] Haochao Ying, Liang Chen, Yuwen Xiong, and Jian Wu. 2016. Collaborative deep ranking: a hybrid pair-wise recommendation algorithm with implicit feedback. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 555–567.

[137] 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 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. ACM, 353–362.

[138] Qi Zhang, Jiawen Wang, Haoran Huang, Xuanjing Huang, and Yeyun Gong. [n. d.]. Hashtag Recommendation for Multimodal Microblog Using Co-Attention Network. In *The 26th International Joint Conference on Artificial Intelligence (IJCAI 2017)*.

[139] Shuai Zhang, Lina Yao, and Xiwei Xu. 2017. AutoSVD++: An Efficient Hybrid Collaborative Filtering Model via Contractive Auto-encoders. *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval* (2017).

[140] Lei Zheng, Vahid Noroozi, and Philip S. Yu. 2017. Joint Deep Modeling of Users and Items Using Reviews for Recommendation. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining (WSDM '17)*. ACM, New York, NY, USA, 425–434. https://doi.org/10.1145/3018661.3018665

[141] Yin Zheng, Cailiang Liu, Bangsheng Tang, and Hanning Zhou. 2016. Neural Autoregressive Collaborative Filtering for Implicit Feedback. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems (DLRS 2016)*. ACM, New York, NY, USA, 2–6. https://doi.org/10.1145/2988450.2988453

[142] Yin Zheng, Bangsheng Tang, Wenkui Ding, and Hanning Zhou. 2016. A Neural Autoregressive Approach to Collaborative Filtering. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48 (ICML'16)*. JMLR.org, 764–773. http://dl.acm.org/citation.cfm?id=3045390.3045472

[143] Jiang Zhou, Cathal Gurrin, and Rami Albatal. 2016. Applying visual user interest profiles for recommendation & personalisation. (2016).

[144] Fuzhen Zhuang, Dan Luo, Nicholas Jing Yuan, Xing Xie, and Qing He. 2017. Representation Learning with Pair-wise Constraints for Collaborative Ranking. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. ACM, 567–575.

[145] 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.

[146] Yi Zuo, Jiulin Zeng, Maoguo Gong, and Licheng Jiao. 2016. Tag-aware recommender systems based on deep neural networks. *Neurocomputing* 204 (2016), 51–60.