



# Artificial intelligence in recommender systems

Qian Zhang<sup>1</sup> · Jie Lu<sup>1</sup> · Yaochu Jin<sup>2</sup>

Received: 26 June 2020 / Accepted: 28 September 2020 / Published online: 1 November 2020  
© The Author(s) 2020

## Abstract

Recommender systems provide personalized service support to users by learning their previous behaviors and predicting their current preferences for particular products. Artificial intelligence (AI), particularly computational intelligence and machine learning methods and algorithms, has been naturally applied in the development of recommender systems to improve prediction accuracy and solve data sparsity and cold start problems. This position paper systematically discusses the basic methodologies and prevailing techniques in recommender systems and how AI can effectively improve the technological development and application of recommender systems. The paper not only reviews cutting-edge theoretical and practical contributions, but also identifies current research issues and indicates new research directions. It carefully surveys various issues related to recommender systems that use AI, and also reviews the improvements made to these systems through the use of such AI approaches as fuzzy techniques, transfer learning, genetic algorithms, evolutionary algorithms, neural networks and deep learning, and active learning. The observations in this paper will directly support researchers and professionals to better understand current developments and new directions in the field of recommender systems using AI.

**Keywords** Recommender systems · Artificial intelligence · Computational intelligence

## Introduction

It is challenging for businesses in a competitive marketplace to offer products and services that appeal directly to an individual customer's needs. Personalized e-services help to solve a major problem—that of information overload—thereby making the decision process easier for customers and enhancing user experience. The recommender systems used in these personalized e-services were first established twenty years ago and were developed by employing techniques and theories drawn from other artificial intelligence (AI) fields for user profiling and preference discovery. The past few years have seen a huge increase in success-

ful AI-driven applications. Successes include Deepmind's AlphaGo, the AI-driven program that famously won the game 'Go' against a professional human player, and the self-driving car, as well as others in the areas of computer vision and speech recognition. These continuing advances in AI, data analytics and big data present a great opportunity for recommender systems to embrace the impressive achievements of AI.

Various AI techniques have more recently been applied to recommender systems, helping to enhance the user experience and increase user satisfaction. AI enables a higher quality of recommendation than conventional recommendation methods can achieve. This has propelled a new era for recommender systems, creating advanced insights into the relationships between users and items, presenting more complex data representations, and discovering comprehensive knowledge in demographical, textural, virtual and contextual data.

The aim of this paper is to review the most recent and cutting-edge theoretical and practical contributions to the field, to identify limitations, and to indicate new research directions in the development and application of AI in recommender systems. It will attempt to survey the issues related to recommender systems using AI, and the capacity of AI to

✉ Jie Lu  
Jie.Lu@uts.edu.au  
Qian Zhang  
qian.zhang-1@uts.edu.au  
Yaochu Jin  
yaochu.jin@surrey.ac.uk

<sup>1</sup> Decision Systems and e-Service Intelligence Laboratory, Australian Artificial Intelligence Institute, University of Technology Sydney, Sydney, NSW 2007, Australia

<sup>2</sup> Department of Computer Science, University of Surrey, Guildford, Surrey GU27XH, UK

aid the understanding of large data sets and convert data into knowledge. In this paper, we have reviewed the improvements AI has made to recommender systems, such as the inclusion of fuzzy techniques, transfer learning, neural networks and deep learning, active learning, natural language processing, computer vision and evolutionary computing. The main contributions of this paper are as follows:

1. A systematic review of eight fields of AI methods and their applications in recommender systems;
2. An overview of state-of-the-art AI in recommender systems including models, methods and applications;
3. A discussion of open research issues, revealing the directions of new trends and future development, expanding the scope of how AI techniques can be applied in recommender systems.

The remainder of this paper is as follows. Section 2 provides an introduction to the basics of recommender system models and methods; Section 3 examines the AI techniques currently used in recommender systems; Section 4 reviews how AI techniques are used in recommender systems and their areas of application; Section 5 considers the challenges and future directions of research on AI driven recommender systems. Finally, Section 6 concludes this paper.

## Recommender systems: main models and methods

The explosive growth in information on the World Wide Web and the rapid increase in e-services has presented users with a huge number of choices, which often lead to more complex decision-making. Recommender systems are primarily devised to assist individuals who are short on experience or knowledge to deal with the vast array of choices they are presented with [1]. Recommender systems take advantage of several sources of information to predict the preferences of users for items of interest [2]. This area of research has been the focus of great concern for the past twenty years in both academia and industry, and research in this field is often motivated by the potential profit that recommender systems can generate for businesses such as Amazon [3]. Recommender systems were first applied in e-commerce to solve the information overload problem caused by Web 2.0, and they were quickly expanded to the personalization of e-government, e-business, e-learning, and e-tourism [4]. Nowadays, recommender systems are an indispensable feature of Internet websites such as Amazon.com, YouTube, Netflix, Yahoo, Facebook, Last.fm, and Meetup. In brief, recommender systems are designed to estimate the utility of an item and predict whether it is worth recommending. The core element of a recommender system is [5]:

$$f : U \times I \rightarrow D.$$

This is a function to define the utility of a specific item  $i \in I$  to a user  $u \in U$ .  $D$  is the final recommendation list containing a set of items ranked according to the utility of all the items the user has not consumed. The utility of an item is presented in terms of user ratings. Recommender systems find an item for the user by maximizing the utility function, formulated as follows [5]:

$$\forall u \in U, \arg \max_{i \in I} f(u, i).$$

Predicting the utility of items for a particular user varies according to the recommendation algorithm selected. Referencing the classical taxonomies of previous research [4–6], recommendation techniques fall into three categories: content-based, collaborative filtering (CF)-based and knowledge-based approaches. These three categories will be reviewed in the following subsections.

### Content-based recommender systems

As the name suggests, content-based recommender systems make use of the content of an item's description to predict its utility based on a user's profile [7]. Content-based recommender systems aim to recommend items that are similar to items that have previously interested in a specific user. First, different item properties are extracted from documents/descriptions. For instance, a movie can be represented by attributes such as genre, the director, writer, actors, storyline, etc. These properties can be obtained directly from structured data, such as a table, or from unstructured data, such as an article or news. One of the most commonly used retrieval techniques in content-based recommender systems is a keyword-based model known as the vector space model with term frequency-inverse document frequency weighting [8]. Content-based recommender systems profile a user's preferences from items in that user's consumption records. The profile usually comprises information about what the user has liked or disliked in the past. Thus, the profiling process can be seen as a typical binary classification problem, which has been well studied in machine learning and data mining fields. Classic methods such as Naïve Bayes, nearest neighbor algorithms and decision trees are used in this step [9]. Once the user's profile has been established, the system compares the item's attributes with the user's profile and finds the most relevant items from which to form a recommendation list. Recommendation in a content-based recommender system is a filtering and matching process between the item representation and the user profile, based on the features acquired in the first two steps. The final result is to forward the matched items and remove those items the user tends to

dislike, so the relevance evaluation of the recommendation is clearly dependent on the accuracy of the item's representation and the user's profile [10].

The content-based recommender system has several advantages [11, 12]. First, content-based recommendation is based on item representation and is thus user independent. As a result, this kind of system does not suffer from the data sparsity problem. Second, content-based recommender systems are able to recommend new items to users, which solves the new item cold-start problem. Finally, content-based recommender systems can provide a clear explanation of the recommendation result. The transparency of this kind of system is a great advantage compared to other techniques in real-world applications. There are nevertheless several limitations to content-based recommender systems [5, 13]. Although such systems overcome the new item problem, they still suffer from the new user problem because the lack of user profile information seriously affects the accuracy of the recommendation result. Furthermore, content-based systems always choose similar items for users, leading to overspecialization in the recommendation. Users tend to become bored with these types of recommendation lists because most users want to learn about new and fashionable items rather than being limited to items similar to those they have previously used. Another issue is that items cannot always be easily represented in the specific form required by content-based recommender systems. This kind of system is, therefore, more suitable for recommending articles or news items rather than images or music.

### Collaborative filtering-based recommender systems

In contrast to content-based recommender systems, which are independent of other users but dependent on a user's personal historical records, CF-based recommender systems infer the utility of an item according to other users' ratings [13]. This technique has been widely researched in academia [14] and was quickly applied in the industry more than 20 years ago [15]. Today, CF is still the most popular technique applied in recommender systems [16]. The basic assumption underpinning the CF technique is that users who share similar interests will consume similar items, so a system using the CF technique relies on information provided by users who have similar preferences to the given user. A classic scenario in CF is to predict a user's ratings on unconsumed items from a user-item rating matrix, which is related to the matrix completion problem [17]. CF-based techniques are classified into two categories [18]: memory-based CF and model-based CF.

Memory-based CF is an early generation CF that uses heuristic algorithms to calculate similarity values between users or items, and can therefore be subdivided into two types: user-based CF and item-based CF [19]. The core algorithm used in the memory-CF technique is the nearest neighbor

algorithm. The recommendation calculates and ranks the rating of a target user on different items based on the neighbor ratings of a user or item. This algorithm is well accepted because of its simplicity, efficiency and ability to produce accurate results. Although memory-based CF is well known for its easy implementation and relatively effective and practical application, the technique still has some non-negligible drawbacks [5]. First, it is not able to deal with the cold-start problem. When a new user/item enters the system, there are no ratings for the system to use to make predictions. Second, if an item is not new but is unpopular with users, it will receive very few ratings from consumers. Memory-based CF is unlikely to recommend unpopular items to users; therefore, the recommendation coverage is limited. Third, it cannot provide a real-time recommendation. The heuristic process takes a long time to provide a recommendation result, especially when the dimension of the user-item rating matrix is high. This problem can be partially solved by a pre-calculated and pre-stored weighting matrix in item-based CF [19], but the scalability is still unable to meet practical needs.

Model-based CF builds a model to predict a user's rating on items using machine learning or data mining methods rather than heuristic methods, as discussed in the previous section. This technique was originally designed to remedy the defects in memory-based CF, but it has been widely studied for solving problems in other domains. In addition to the user-item rating matrix, side information is used, such as location, tags and reviews [20]. The model-based CF technique is a good choice if this ancillary information is combined with the rating matrix. Matrix factorization was a product of the Netflix Prize competition of 2009 [21], and it is still one of the most popular algorithms in this field. It projects both user space and item space onto the same latent factor space so that they are comparable. Three advantages of matrix factorization contribute to its popularity. First, the dimension of the user-item rating matrix can be reduced significantly, so the scalability of the system employing matrix factorization is secured. Second, the factorization process makes a dense rating matrix, so that the sparsity problem can be alleviated [22]. Users who only have a few ratings can acquire relatively more accurate recommendation through matrix factorization, which is a significant improvement over memory-based methods. Third, matrix factorization is highly suitable for integrating a variety of side information [23]. This helps to profile user preferences and improves the performance of recommender systems.

### Knowledge-based recommender systems

In knowledge-based recommender systems, recommendations are based on existing knowledge or rules about user needs and item functions [6]. Unlike content-based and CF-based techniques, knowledge-based recommender systems

retain a knowledge base that is constructed with knowledge extracted from a user's previous records. This knowledge-base contains previous problems, constraints, and corresponding solutions. Knowledge in the knowledge base is referenced when the system encounters a new recommendation problem [24]. Case-based reasoning uses previous cases to solve the current problem [25] and is a commonly used technique for knowledge-based systems. In contrast to content-based recommender systems, finding the similarities between products requires more structured representations. In this process, a comparison of a previous case and the current case is made, along with solution adaptation.

The application of the knowledge-based recommendation technique is of particular value in house sales, financial services, and health decision support [26]. These services are characterized by highly specific domain knowledge, and each case presents a unique situation. One advantage of this technique is that the new item/user problem does not exist, since prior knowledge is acquired and stored in the knowledge base. Another advantage is that users can impose constraints on the recommendation results [27]. However, no advantage comes without a corresponding disadvantage, and in this case, the cost of system setup and management in building and maintaining the knowledge base is usually high.

## Artificial intelligence: main models and methods

Artificial intelligence is a fast-developing field in which applications range from playing chess to learning systems or diagnosing disease [28]. The goal of developing AI techniques is to achieve automation of intelligent behaviors which mainly cover six areas: knowledge engineering, reasoning, planning, communication, perception, and motion [29]. Specifically, knowledge engineering refers to techniques that are used for knowledge representation and modelling to enable machines to understand and process knowledge; Techniques for reasoning are developed for problem solving and logical deduction; Planning is to help machines to set and achieve a goal; Communication aims to understand natural language and communicate with human; Perception plays the role of analyzing and processing inputs such as images or speech; and finally motion is about movement and manipulation. Except for the motion, techniques in the first five areas can be applied to enhance and boost the development of recommender systems due to the huge information processing demands.

In this section, we will introduce eight main models and methodologies as shown in Fig. 1. Deep neural networks, transfer learning, active learning, and fuzzy techniques are representatives for knowledge and reasoning and are interconnected with each other. Evolutionary algorithms and

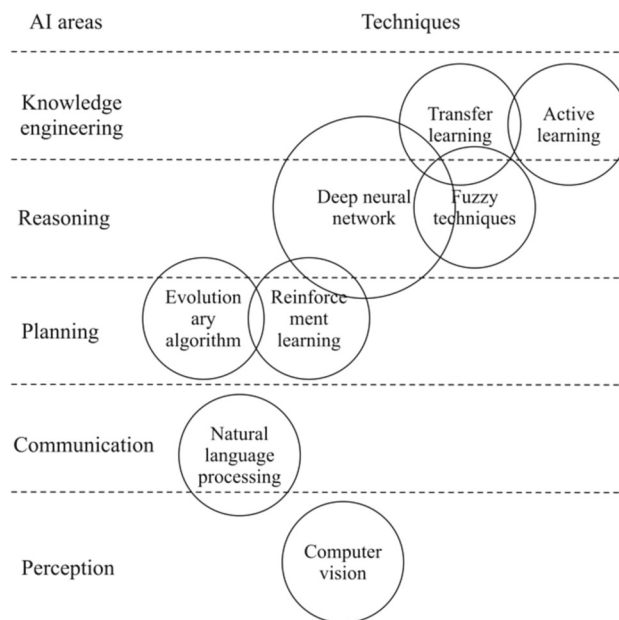


Fig. 1 AI areas and techniques

reinforcement learning are related to reasoning and planning, while natural language processing is the main technique for communication and perception, and computer vision is for the perception of images. Among the eight methods, natural language processing and computer vision are two application areas of AI techniques in recommender systems.

## Deep neural network

Neural network is inspired by the network of neurons in the human brain. A neural net consists of a set of neurons (or nodes) that receive and process signals from connected neurons/nodes. Each neuron can change its internal state (activation) according to the signal received so that activation weights and functions can be learned and modified in the learning process. In 1980s, neural nets were largely forsaken and ignored by the machine learning community. By the late of 1990, however, a particular type of deep feedforward network called convolutional neural network (CNN) was developed which is much easier to train [30]. CNN can also be much better generalized than traditional neural networks; they were thus quickly adopted in the areas of speech recognition and computer vision [31]. Deep learning includes the following diverse types [32]:

Multilayer perceptions (MLP) [33] are feed-forward neural networks consisting of three or more layers with a non-linear activation. It allows approximate solutions to be found for both regression and classification problems.

Autoencoders (AE) [34] are unsupervised neural networks for learning feature representations where the purpose is dimensionality reduction, data compression, or data denois-

ing. It usually consists of two parts, the encoder and the decoder, which reconstruct the input in the output.

Convolutional neural networks (CNN) [35] are capable of processing images and visual information. It consists of an input layer, an output layer and multiple hidden layers, in which convolutional layers, pooling layers, fully connected layers or normalization layers are usually contained.

Recurrent neural networks (RNN) [36] are designed to deal with sequence data since its node connections form a directed graph. It uses internal states as memory so that sequence processes can be remembered. Representative RNN is a long short-term memory (LSTM) network [37] which is suitable for time series prediction.

Generative adversarial networks (GAN) [38] are used for unsupervised learning tasks and is implemented by two sets of models. One is a generative model and the other is a discriminative model. These two models compete to generate samples that look like the original samples.

Graph neural networks (GNNs) [39] are motivated by CNN and graph embedding to model the graph structure between nodes with neighborhood information included. GNNs have advantages in graph structured data for representation learning, link prediction and node classification, due to their high performance and good interpretability.

## Transfer learning

Machine learning has attracted great attention because of the assumption that trained models can solve problems of prediction or classification, given that the training data and test data are under the same distribution. In practice, however, test data is usually dynamic and diverges from the training data. This results in the inapplicability of the current model and requires it to be rebuilt, which takes great effort. It is not always possible to retrain and build a new learning-based model since the newly collected data may be insufficient, and there are usually not enough labels accompanying the new data. This problem is extremely serious in many real-world scenarios.

Unlike traditional machine learning, transfer learning has developed as a means of transferring knowledge from a domain with relatively rich data (source domain) to a domain with scarce data (target domain) [40]. In this definition, transfer learning aims to extract knowledge from one or more source data to assist a learning task with target data. Transfer learning techniques can be divided into three main categories [41]. (1) Inductive transfer learning. The target task is different from the source task. When labeled data are available in the target domain, inductive transfer learning is similar to multi-task learning [42]. On the other hand, if there are no labeled data in the target domain, it is known as self-taught learning. (2) Transductive transfer learning. The source and target tasks are the same, but the source and target

domains are different. Transductive transfer learning is also used interchangeably with domain adaptation [43]. For this type of transfer learning technique, the discrepancy between the source domain and the target domain can be caused by the existence of different feature spaces, or the different marginal distribution of feature spaces [44]. (3) Unsupervised transfer learning. The setting is similar to inductive transfer learning, but the target tasks are unsupervised learning tasks. Unsupervised transfer learning is similar to semi-supervised learning [45], except that there are no labeled data for either the source domain or the target domain. In the literature, domain adaptation, covariate shift, sample selection bias, multi-task learning, robust learning, and concept drift are all terms which have been used to describe the related scenarios.

## Active learning

The basic idea of active learning is to selectively choose from training data to enable machine learning to perform better with less information. A system with an active learning strategy may query users to provide labels for unlabeled instances [46]. As the labeling process may be expensive, time-consuming and sometimes impossible, active learning can usefully be applied to many areas in AI and is especially suitable for online systems. Many AI areas related to classification or regression problems, such as speech recognition, information retrieval and computational biology, benefit from active learning [47].

Active learning strategies can be roughly divided into several groups according to their evaluation criteria on unlabeled instances. They include uncertainty sampling, query-by-committee, expected model change, expected error reduction, variance reduction, and density-weighted methods [48]. Uncertainty sampling queries instances that are least confident to be labeled. Query-by-committee is a framework that aims to minimize the inconsistency of the query to current labeled training data. Expected model change selects those instances that maintain the least change to the established model. Expected error reduction measures global error and reduces potential risk to include the queried instance. Variance reduction follows a similar direction as expected error reduction but cuts down on variance to increase the stability of the established model. Density-weighted methods search for representative instances which are important for boundary decisions or representing controversial situations.

## Reinforcement learning

Reinforcement learning aims to maximize reward in a sequence of actions of a learning agent to achieve a goal, while the next situation (input) will be affected by the actions in an interactive way [49]. Different from supervised learning which relies on a labeled training set, reinforcement learning

is to train an agent that can act in a situation that is not shown in the training set. It is also different from unsupervised learning, which mine patterns from unlabeled data whereas reinforcement learning is to achieve the long-term goal by interaction with the environment. The generality of reinforcement learning makes it widely applied in various aspects such as game theory [50], optimal control [51], swarm intelligence [52] and other areas such as healthcare [53] and psychology [54].

Usually, reinforcement learning follows the definition of Markov decision process [55] to describe how the agent interacts with the environment: at a step, the agent receives a state, selects an action according to a policy and receives a reward for this step, then transit to the next step. A value function will define the long-term reward accumulated during the whole process containing a series of steps. A unique challenge that exists in reinforcement learning is the dilemma between exploration and exploitation [56]. The learning agent is facing a choice to take actions that it has experienced in the past or try new actions that may bring more rewards. The balance of the dilemma lies in whether to exploit actions that in the historical records or explore new actions that finally come to a reward maximization. The methods of reinforcement learning can be divided according to value function, policy, and model in value-based or policy-based, off-policy or on-policy, model-based or model-free and hybrids of the above [57]. Recently, the combination of deep neural networks and reinforcement learning becomes popular with two well-known and successful works: deep Q-network [58] and AlphaGo [59]. Deep neural networks significantly boosted reinforcement learning in dealing with high dimensional states or/and actions and make it as an indispensable component in future AI systems.

### Fuzzy techniques

Fuzzy techniques can be used to model real-world concepts that cannot be represented in a precise way; thus, it is widely used in the AI area. Fuzzy techniques have attracted considerable attention in the literature; for example, researchers have applied fuzzy sets to represent linguistic variables when feature values cannot be precisely described in numerical values, and to describe fuzzy distance for the retrieval of similar cases [60]. Knowledge extracted from data is hidden and uncertain by nature, so using fuzzy logic and fuzzy rule theory to handle the associated vagueness and uncertainty is apt and can improve the accuracy of both classification and regression [61]. Fuzzy techniques facilitate data and knowledge sharing between businesses where knowledge can be used to build data analytics models efficiently [62]. This has the advantage of significantly reducing the computational expense incurred by businesses, particularly in data-shortage

and rapidly-changing environments, and provides outstanding benefit to their business intelligence systems.

### Evolutionary algorithms

Evolutionary algorithms (EAs) are a sub-area of AI research that form a class of nature-inspired, population-based search algorithms for global optimization. An evolutionary algorithm starts with an initial population, known as the parent population, which is a set of candidate solutions to a problem to be solved. New solutions, called offspring, are generated by applying genetic operators such as crossover and mutation to parent individuals. Offspring individuals are selected according to their fitness to become the parents of the next generation. This process continues until certain termination conditions are met.

There are three independently developed streams of evolutionary algorithms: the genetic algorithm [63], evolution strategies [64], and genetic programming [65]. Other popular EAs include estimation of distribution algorithms [66] and differential evolution [67]. Several other nature-inspired meta-heuristic algorithms have also been developed, such as particle swarm optimization [68] and ant colony optimization [69], which are sometimes categorized as EAs in a very loose sense. Although they were designed to solve a wide range of problems, EAs have been shown to be very powerful in solving complex optimization problems that are difficult for traditional mathematical programming techniques to solve. Evolutionary algorithms (EAs) are divided into single-objective and multi-objective EAs [70] according to the number of objectives to be optimized. Multi-objective EAs that have more than three objectives are also termed many-objective EAs [71].

### Natural language processing

Natural language processing is a traditional research area in AI that dates back to the 1950s. Its origins lie in the recognition of hand-written image analysis, and it entered a new era with the development of machine learning [72]. Text data are different from other kinds of structured data; their most important characteristics are sparsity and high dimensionality. They can be analyzed at different levels of representation, such as bag-of-words, topics or embedded vectors. Many machine learning algorithms, such as support vector machine and Bayesian network [73], can be applied to a wide range of natural language processing areas, as detailed below.

To illustrate the broad reach of natural language processing, the various tasks are clustered but not limited to the following aspects. Information extraction aims to extract structured information from unstructured text and includes entity extraction and relationship extraction [74]. Text summarization analyzes the importance of sentences, then scores

and selects the set of best sentences to compose a summary. Text classification is widely used in data mining research to label text and relate it to multiple applications, such as customer segmentation, document organization, and CF [75]. Sentiment analysis extracts hidden opinion, sentiment and subjective information from the text to assist with classification or prediction [76]. Dimensionality reduction techniques such as latent semantic indexing, topic modeling, and latent Dirichlet allocation are widely used in natural language processing to reduce the number of variables and obtain a set of principal variables [77]. The evolution of text corpus and its interactions with other context data or heterogeneous data have also been well researched in AI.

### Computer vision

Humans can directly recognize an object by discerning its shape, color, motion and related characteristics. As increasing amounts of data with images and video accumulate, it is desirable for machines to obtain high-level understanding from vision through such techniques as object capture, recognition or tracking [78]. A number of models have been established that describe and process images or videos to effectively contribute to classification, detection, and segmentation problems. Recent developments in deep learning have revolutionized the computer vision research area, given the ability of deep learning methods to extract features [79]. This has prompted their use in computer vision tasks for analyzing, processing and describing digital images and videos. In particular, CNN has been widely adopted for recognition and detection tasks [80], which has resulted in huge changes being made in image processing, not only in academia but also in industry.

### Recommender systems with artificial intelligence

Multiple artificial intelligent techniques have been introduced and applied to recommender systems to meet the increased recommendation demands of the big data information explosion. In this section, we highlight six AI techniques that have enhanced recommender systems.

#### Deep neural networks in recommender systems

Neural network is rarely used in recommender systems since the task of recommendation concerns the ranking of items rather than classification. In an early work, Salakhutdinov et al. proposed a two-layer restricted Boltzmann machine (RBM) to explore the ordinal property of ratings. This method attracted great attention in the 2009 Netflix Prize competition [81], but there has been little follow-up work

apart from research by Truyen et al., who extended this work by studying the parameterization options of RBM in recommendation [82]. In contrast, deep learning has achieved great success in the fields of natural language processing, speech recognition and computer vision [31]. With the availability of more data (e.g., user-generated comments or visual photos of items), the need to integrate all the information and provide recommendation for multi-media items, such as images or videos, prompted the development of deep learning-based recommender systems [83]. In this sub-section, we divide deep learning-based recommender systems according to the different types of deep neural networks applied in recommender systems.

#### Multi-layer perceptron-based recommender systems

Multi-layer perceptron is used in factorization machines to help with feature engineering. It combines the advantages of linear and non-linear modeling in one recommendation framework [84]. Guo et al. improved the wide and deep model in [84] as the proposed factorization machines can be trained without feature engineering [85]. He et al. proposed neural collaborative filtering (NCF) to model the non-linear relationship between users and items in conjunction with matrix factorization to model the linear relationship [86]. NCF, which is based on multi-layer perceptrons, is widely used in recommender systems as a general model for user-item interactions.

#### Autoencoder-based recommender systems

AutoRec integrates an autoencoder with matrix factorization with the aim of learning non-linear latent representations of users or items [87]. AutoSVD++ is a hybrid method that fuses a contractive autoencoder and matrix factorization to generate item feature representations from item content [88]. Strub et al. improved AutoRec by boosting its robustness through the use of denoising techniques and integrating such side information as item content or user-contributed tags [89]. Autoencoder serves as a basic building block for representation learning which is well suited for user profiling and item representation learning in recommender systems.

#### Convolutional neural network-based recommender systems

By integrating two parallel neural networks, DeepCoNN jointly models users and items through reviews [90]. The two CNNs are connected by a shared layer facilitated by factorization machines. To exploit the information in user-contributed reviews and address the data sparsity problem, ConvMF integrates CNN into matrix factorization to improve rating prediction accuracy [91]. CNN has also been used for

the hashtag recommendation task in microblogging by introducing the attention mechanism in the process of selecting the hashtags [92].

### Recurrent neural network-based recommender systems

Since RNN is suitable for sequential data, it is mainly used to model and analyze the evolution of user interests or item features. Dai et al. applied RNN and proposed a co-evolutionary latent feature process for modeling the temporal dynamics of user-item interactions [93]. Wu et al. used an LSTM-based model to capture the dynamics of user behavior to predict whether or not to inherit existing user behavior in the future [94]. LSTM is also used in recommender systems to make in-time music recommendations, to predict when users will return to a music system and what their interest will be at that time [95].

RNNs have emerged as a new direction known as session-based recommender systems or sequential recommender systems where the real-time recommendation is refined according to the historical sequential data [96, 97]. In [98], the most recent states are modelled by an RNN to predict the next item that may attract the interests of users. The early works did not take into consideration of the short-term and long-term user interests in the sequence. Later, the current state is modelled as a short-term user preference and the session state is modelled by RNNs with an attention mechanism as the long-term preference. They are equally integrated and matched with an item through a bi-linear scheme [99]. The short-term user preference is enhanced in [100] and user preference drift is also taken into consideration. Further, the two kinds of preferences are fine-tuned by a hierarchical attention network [101]. Sequential recommender systems are gaining more attention in research dealing with the relationship between short-term and long-term interests as well as integrating contextual information and preference dynamics.

### Generative adversarial network-based recommender systems

Wang et al. integrated GAN to a unified information retrieval framework. It contains a generative retrieval model that learns the distribution over documents and try to generate relevant documents that look like the ground truth to fool the discriminative model, and a discriminative model that aims to classify the ground-truth documents from the generated ones as an opponent to the generative model [102]. This approach shows that GAN-based information retrieval systems offer promise, and further effort is needed specifically in the recommender system area. He et al. introduced perturbations on the user and item embedding as an adversarial regularizer under the framework of Bayesian personalized ranking [103]. A GAN is used to learn robust user/item representations not

only from user-item interactions but also from knowledge graph [104], tags and images [105].

### Graph neural network-based recommender systems

The ability of GNNs to learn feature for nodes from the information of neighborhoods in the graph is highly desired for recommender systems, as the user-item relationships are usually represented as a bipartite graph. The feature embedding by a GNN and random walk are incorporated in [106] and a highly scalable and efficient recommendation method is proposed and deployed in Pinterest. This work shows the great potential of GNNs to improve the productivity of recommender systems. A generalized graph neural network-based CF framework is proposed in [107] with attention-based message-passing method for information propagation. GNN is also suited for sequential recommender systems to model the item sequences as a graph [108]. It is superior as user-item interactions are considered in the sequence while an RNN can only model one-side item information. GNN-based recommender systems are just emerging and more studies in social recommendation, sequential recommendation and cross-domain recommendation are expected.

Current trends of application of deep neural networks in recommender systems are towards addressing more complex situations such as dynamic environments, multiple data sources and heterogeneous data representations. They aim to develop methods and build models with hybrids of different types of deep neural networks to comprehensively model the user preferences.

### Transfer learning in recommender systems

Transfer learning has demonstrated great success and a promising future in the machine learning field. In the field of recommender systems, transfer learning extends recommendation requests from a single domain to multiple domains. By exploiting the correlation of several domains, all domains can benefit from mining user preferences that cannot be found with single domain data. For example, an active user in a movie domain is likely to be interested in books and music related to movies they like. Another reason to exploit multiple domains is to solve the data sparsity or cold-start problem, as there may be insufficient data in one domain but relatively rich data in another domain. For example, a user may have few records in a book category in an online review and rating system but may have a large number of movie ratings, thus an abundance of data in a secondary domain can assist recommendation in the target domain. This demand for a rich and diverse recommendation, together with the ability to alleviate the data sparsity problem, has driven the development of cross-domain recommender systems (CDRS).



The biggest difference between CDRS and other transfer learning methods is that there is no explicit feature space in CDRS. This means that CDRS cannot be classified as a single type of transfer learning method, because they involve the practical application of multiple transfer learning techniques. From the practical perspective, CDRS provide multi-domain recommendation for online shopping retailers selling a variety of goods while at the same time offering a solution to the data sparsity problem. Some methods connect two domains through auxiliary information other than preference data [20], while CDRS based on preference data can be strategically designed according to the overlap of users and items, the form the data takes, or the tasks the system needs to handle [109]. We classify CDRS according to these three different scenarios and review them below.

### CDRS with side information

For this type of recommender system, it is assumed that some side information on entities is available, such as user-generated information, social information or item attributes. Collective matrix factorization (CMF) is designed for scenarios in which a user-item rating matrix and an item-attribute matrix for the same group of items are available [110]. CMF collectively factorizes these two matrixes by sharing item parameters, since the items are the same. Other methods have since been developed that exploit social network information to assist cross-domain recommender systems. Yang et al. used a bipartite graph to represent the relationships between entities across heterogeneous domains and exploit hidden similarity to help recommendations in two domains [111]. Excluding social network information, many user-generated tags in online systems provide auxiliary data for CDRS. Abel et al. used both a form-based user profile and a tag-based profile to investigate how the social web can be connected with recommender systems to assist with cross-system user modeling [112]. Tag-informed collaborative filtering (TagiCoFi) is a proposed method in which a user-item rating matrix and a user-tag matrix for the same group of users are used [113]. User similarities extracted from shared tags are used to assist the matrix factorization of the original rating matrix. Tag cross-domain CF (TagCDCF) extends TagiCoFi to two domain scenarios each containing data from these two matrixes [114]. By simultaneously integrating intra-domain and inter-domain correlations to matrix factorization, TagCDCF improves recommender system performance in the target domain.

### CDRS with non-overlapping entities

Methods that handle two domains with non-overlapping entities transfer knowledge at group-level. Users and items are clustered into groups and knowledge is shared through group-

level rating patterns; for example, codebook transfer (CBT) clusters users and items into groups and extracts group-level knowledge as a “codebook” [115]. A probabilistic model named rating matrix generated model (RMGM) was extended from CBT which relaxes the hard group membership to soft membership [116]. However, these two methods are unable to ensure that the information in the two groups from two different domains is consistent, and the effectiveness of the knowledge transfer is not guaranteed. Zhang et al. [117] used a domain adaptation technique to extract consistent knowledge from the source domain, which proved to be a more superior method, especially when the statistics between the source domain data and the target domain data are divergent. Zhang et al. [118] extended RMGM with an active learning strategy in a multi-domain scenario, which enables queries to be made across several domains by considering both domain-specific and domain-independent knowledge and benefits recommendation in each of these domains.

### CDRS with partially or fully overlapping entities

Given the assumption that entities between two domains overlap, the source domain and target domain are bridged by constraints on the overlapping entities. Methods to handle data where the user and/or item in both domains partially or fully corresponds usually collectively factorize two matrixes in each domain by sharing some part of the factorization parameters. Transfer collective factorization (TCF) [119] has been developed to use implicit data in the source domain to help the prediction of explicit feedback, i.e., ratings in the target domain. Cross-domain triadic factorization (CDTF) models a user-item-domain tensor to integrate both explicit and implicit user feedback [120]. Users are fully overlapped, and the user factor matrix is the same, thus bridging all the domains. Cluster-based matrix factorization (CBMF) tries to boost CDTF to partially-overlapping entities [121]. Since entity correspondence is not always fully available, some strategies have been developed that match users or items in two domains. Unknown user/item mappings are identified in [122] using latent space matching. The identification of the mapping is time-consuming, so an active-learning framework is sometimes developed to identify the most valuable entity correspondences in the source domain [123]. Zhang et al. proposed a kernel-induced knowledge transfer method for cross-domain recommender systems with partially overlapped entities where alignment on heterogeneous latent feature spaces between two domains is taken into consideration [124].

The above mentioned CDRSs are mainly based on shallow learning methods. The recent developments of deep neural networks are also applied in knowledge transfer and cross-domain recommendation. A framework for CDRS on partially overlapping entities with a deep neural net-

work is proposed in [125]. Knowledge transfer between two domains in this framework is achieved by mapping the user/item features in the target domain with the combined features obtained from both domains. Hu et al. also propose a cross-domain recommendation method by sharing the hidden layers between two domains [126]. GAN is applied with an additional objective function to discriminate user/item embedding features into different domains [127]. A general CDRS framework with a GAN is proposed in [128] to deal with all the three scenarios above. The application of deep neural networks in CDRS is well received due to their power of robust feature extraction and their capability of sharing knowledge in different levels of granularity. Knowledge is transferred through the overlapped entities as a bridge with both rating and content information and benefits both the source and the target domains in [129]. As the data are accumulated from multiple sources, further studies of CDRS that is able to deal with multi-domain knowledge transfer are needed.

### Active learning in recommender systems

Each user-item correlation in a recommender system—especially one based on explicit ratings or implicit interactions between users and items—is crucial for profiling user preferences and substantially affects system performance. The challenge of data sparsity in recommendation reveals that the greater the number of ratings acquired from users, the better a system will perform in providing a recommendation. However, it is time-consuming, labour-intensive, and therefore almost impossible to query users to rate all, or most, items. Active learning has been introduced to help recommender systems select the most representative items and deliver them to users to rate [130]. As user experience is valued and user interactions with systems are desirable in the information era, active learning techniques have been adopted that improve both the efficiency and the accuracy of recommender systems.

Active strategies that used pre-computed bounds on the value of information were employed in early works to reduce the online computation time in recommender systems [131], but academics soon found that the item selection greatly influences rating prediction. There are many different active learning strategies, such as rating impact analysis [132] and bootstrapping [133], and such active learning strategies have been integrated with common recommendation models such as the aspect model [134], decision trees [135], and matrix factorization [136]. Complex factors such as naturally acquired ratings by users [137], the probability of a user being able to provide a rating for the system query [138], the influence of items [139] and the item attributes [140] have been added to the active learning strategy. The active learning strategies are also brought to a multi-domain rec-

ommendation scenario in rating selection [141] and entity correspondence selection [123].

Active learning is mostly used in the early work for item selection in recommender systems. Its combination with more advanced model-based recommendation methods may lead to novel directions. Although many factors have been considered as we reviewed above, still active learning for contextual information selection is rare. The combination of active learning and reinforcement learning is another direction that worth more attention, as its application in recommender systems will further enhance their performance.

### Reinforcement learning in recommender systems

The nature of using recommender system is an interactive process between the user and the system with a series of states and action, which is in accordance with reinforcement learning. Different from traditional recommender systems, which usually focus on predicting interests of users at a specific time point, the reinforcement learning-based recommender systems aim to maximize the engagement and satisfaction of users in a long term. Under the framework of reinforcement learning, the recommender system is treated as a learning agent, the user behaviours correspond to the states and the actions are recommendations generated by the system. The reward is the feedback of the users on the recommendation results, such as the click through the rate or the time duration on the webpage. The target is to find a policy or a value function for the users to maximize the long-term rewards. The challenge of reinforcement learning lies in the large number of items that are available to users, which creates a large action space for learning agents and increases the complexity of the system.

The early work studies mainly the balance of exploration and exploitation, which is also known as bandit problems [142]. A direct implementation of MDP to recommender systems without considering the balance is proposed in [143] to recommend the next item with the previous  $k$  consumed items. Later, the trade-off between exploration and exploitation is addressed with linear reinforcement learning with theoretical guarantee [144]. There is also some work which treats the interactive process between the user and the recommender system as a multi-arm bandit problem [145] and later extended with contextual information [146, 147].

Researches reviewed above mostly focus on the immediate rewards and ignores the long-term rewards. Recently, deep reinforcement learning has gained more attention with the breakthrough of deep Q-network and deep deterministic policy gradient, which have advantages in addressing the immediate and long-term rewards simultaneously [148]. The challenge of large and dynamic actions is tackled in [149] with Actor-Critic architecture to reduce the computational complexity. Negative feedback of the user is taken into

consideration to boost deep reinforcement learning-based recommendation with a pair-wise regularization [150]. The current trend in this direction is to take into account complex user behaviours and knowledge graph information to achieve high efficiency with a large amount of data and large number of items [151]. The application of reinforcement learning techniques in industrial recommender systems is also prevalent, such as in YouTube [152] and Alibaba [153]. The development of deep reinforcement learning-based recommender systems will continue to be a hot area and will be more heavily driven by real-world industrial applications.

### Fuzzy techniques in recommender systems

Item features and user behaviors in real-world recommender systems are usually subjective, incomplete and vague. Fuzzy set and fuzzy relation theories offer an effective way to deal with information uncertainty problems, and can also be adopted in recommender systems [154]. In this section, three groups of fuzzy recommendation approaches are discussed based on the classification of recommender system methods: (1) Content-based recommender systems with fuzzy techniques, (2) memory-based CF recommender systems with fuzzy techniques, and (3) model-based CF recommender systems with fuzzy techniques.

In content-based recommender systems, fuzzy techniques are applied to two phases of the process: profiling and the matching of appropriate items. Fuzzy sets are used to express the uncertainty in item features, especially vague and incomplete item descriptions, as well as the subjective user feedback on those items. Recommendation approaches are developed using fuzzy set theories to discover user preferences and create item representations [155, 156]. As product information often takes the form of tree-structured content information, and because user preferences are vague and fuzzy, a number of fuzzy tree-based recommender systems have been developed for e-commerce [157], business-to-business e-services [158] and e-learning systems [158].

In memory-based CF recommender systems, fuzzy set theories are used to profile the uncertainty in customer preferences [159]. By matching customer interests with the service provided and managing the natural noise of uncertainty, these methods can improve accuracy in certain areas [160]. Cornelis et al. [161] extended the CF framework to make one-and-only item recommendation for personalized e-government by modeling user preferences and similarities with fuzzy relationships. Son et al. [162] used intuitionistic fuzzy recommender systems to enhance diagnoses in clinical medicine. Zhang et al. [163] built a fuzzy user-interest drift detection approach to deal with dynamic user preferences in rapidly changing big data, using fuzzy relationships to measure user-interest consistency.

Several different techniques have been applied in model-based CF recommender systems, including fuzzy network, fuzzy clustering, and fuzzy Bayesian. In fuzzy network techniques, fuzzy rules are extracted using the adaptive neuro-fuzzy inference system (ANFIS) to alleviate the data sparsity issue in CF and predict user preferences, especially for multi-criteria CF [164]. Nilashi et al. [165] used ANFIS for recommender systems with a hybrid of self-organizing map (SOM), based on several fuzzy-based distance measures and similarities. In fuzzy clustering, compared with CF methods with singular value decomposition (SVD) which only allows hard membership clustering, fuzzy C-means is a soft clustering and allows users/items to belong to several groups [166]. Xu et al. transformed user profiles by fuzzifying rating records and clustering them to exclude the noise of uncertainty to improve the accuracy and scalability of item-based CF recommender systems [167]. With regard to fuzzy Bayesian technique, Kant et al. proposed a fuzzy naïve Bayesian classifier which was extended with CF-based, reclusive-based and hybrid recommendation methods [168]. Campos et al. modeled uncertainty in the probability of related users and the description of ratings, combining Bayesian network, soft computing and CF techniques [169]. Fuzzy-based recommendation methods have also been developed for new applications. For example, a recommender system for digital libraries has been developed that suggests useful resources for researchers by using Google Wave technology and integrating fuzzy linguistic modeling [170]. In addition, Bedi et al. used fuzzy logic to measure the agreement of arguments and enhance recommendation with trust, as well as adding an explanation of the recommendation results [171].

Fuzzy techniques are well suited for handling imprecise user preference descriptions (e.g. linguistic terms), knowledge description, and the gradual accumulation of user preference profiles. A future trend is to integrate fuzzy profiling and fuzzy relationship into advanced recommendation methods, including the development of fuzzy neural networks to enhance the performance of recommender systems.

### Evolutionary algorithms in recommender systems

Evolutionary algorithms (EAs) are used to combine the outputs of multiple recommendation algorithms when the recommendation is treated as a multi-objective optimization problem. They are also used to generate user/item profiles and are employed to handle ratings in the recommendation. The application of EAs in recommender systems can be broadly divided into the following three categories.

### Multi-objective recommender systems

Evolutionary algorithms (EAs) are used to optimize these recommender systems by considering multiple performance indicators, e.g., accuracy, novelty and diversity [172–174]. To achieve accurate and diverse recommendations, Karabadjji et al. [175] improved a memory-based CF method by using multi-objective optimization to find neighbors. A new probabilistic multi-objective evolutionary algorithm was proposed in [118] that strikes a good balance between accuracy and diversity, in which a new crossover operator called multi-parent probability genetic operator and a new topic diversity indicator were introduced.

### Evolutionary optimization of user/item profiles

To achieve accurate personalized recommendation, Mu et al. [176] proposed a novel EA with elite population to find the information core, i.e., core users. In the proposed algorithm, an elite population with a new crossover, termed “ordered crossover”, is adopted to accelerate the evolution. To address changing user profiles in recommender systems, Rana and Jain [177] developed a dynamic recommender system that uses an evolutionary clustering algorithm to identify similar users. Chen et al. [178] proposed an interactive estimation of distribution algorithm to offer users recommendations in an interactive manner. The algorithm quantitatively expresses user preference based on human–computer interactions and trains an RBF neural network as the preference surrogate.

### Evolutionary optimization of ratings

Adomavicius et al. [5, 179] discussed how to integrate multi-criteria ratings into recommender systems. This category of algorithms engages multi-criteria ratings in recommendations, which leverages more sophisticated user preferences. Like evolutionary optimization, multi-criteria approach supports decision-making by aggregating a multi-objective optimization problem into a single-objective problem, by searching for Pareto optimal recommendations, or by taking the multiple criteria as the constraints. To handle the data sparsity problem, Hu et al. [65] utilized a genetic algorithm to optimize the weights of the domains to weight their influences within the framework called generalized cross-domain triadic factorization model over the triadic relation user-item-domain.

One future trend of EA applications will be to develop secure federated recommender systems and interactive recommender systems. Federated learning [180] is able to preserve privacy by sending model parameters to a server instead of storing data in a central server. To reduce com-

munication overheads, it is important to reduce the number of parameters in a model, thus EAs can be used to optimize models in federated learning. Additionally, they can play an important role in creating secure recommender systems in which the model is less vulnerable to adversarial attacks, e.g., malicious manipulation of the data [181], because they can be used to generate models that are less sensitive to malicious data manipulation. Due to its capability of handling multiple objectives, new requirements can be taken into account in designing recommender systems, in addition to accuracy and diversity [182]. These requirements can also be produced from an interactive process, where EAs can be used to fulfill user requirements in each state.

### Natural language processing in recommender systems

Recent developments in deep neural networks exploit the structure of natural language and vision, especially in the RNN, CNN and GNN-based methods. In addition to the reviews, we did in Sect. 4.1, the following two sections will introduce how recommender systems can benefit from natural language processing and computer vision with the integration of free text (e.g. reviews) and visual images (e.g. photo of items).

Recommender systems in the movie and star rating domains are well developed, but a huge amount of text information such as item metadata, item description text, user-generated tags or reviews is not taken into account. Many fine-grained opinion mining and topic modeling methods have already been established in natural language processing, and efforts are increasingly being made to connect these two areas to extract information from the text and incorporate it into the recommendation process. Most recommender systems benefit from review information extracted by natural language processing to complement the rating matrix and alleviate the data sparsity problem. In extreme conditions when ratings are not available, virtual ratings are generated by sentiment polarity gained from review classification [183]. Item metadata in “bag-of-words” representation are analyzed by topic models, which are integrated with matrix factorization methods to manage both cold-start and warm-start scenarios [184]. By mining feature-based product descriptions from reviews, Dong et al. enhanced recommendation with feature sentiment and product experience to provide superior products according to user query [185]. In a similar case, user expertise was evaluated and the evolution of user experience was tracked through online reviews, suggesting that similar users with an equivalent level of experience are likely to respond similarly to the same product [186].

Free-text information is still of great value even when data are not sparse. User reviews are required to discover and interpret latent user features and improve the quality of recommendation in both accuracy and transparency [187]. Ling et al. extended this method to make the learnt latent topic interpretable, thus enabling the recommendation of completely “cold” items [188]. Review text has been incorporated in cross-domain recommendation methods where user vectors are mapped through non-linear functions [189]. The neural embedding algorithm, which has recently become popular in natural language processing, has also been linked with a CF framework to infer item similarity correlations [190], and multi-level item organization has been learnt and applied to personalized ranking [191].

Previous works mostly focus on static data of reviews, text content or item descriptions. As the digital voice systems such as Siri, Google home are becoming more and more mature [192], an interactive recommender system with voice feedback is a new direction where natural language processing techniques will play an important role.

### Computer vision in recommender systems

Recommender systems have benefited from the development of computer vision technologies, especially in the areas of fashion analysis and products that are highly related to visual appearance, such as clothes, jewellery, and images. The combination of image recognition and deep learning neural networks in recommender systems produces outstanding results.

One direct application is used in image recommendation. A dual-net deep network was proposed in [193] that directly applies computer vision to image recommendation to map images and user preferences. Early works in other e-commerce recommendation areas take advantage of the features extracted from images using deep neural networks and integrate them with existing methods for clothing recommendation [194]. Extended research in this area has added low-level features that mimic aspects of the human vision system, such as color characteristics, into this framework [195]. Zhao et al. integrated the visual features extracted from movie posters and still frames with a matrix factorization model to understand user preferences in movie recommendation from a new aspect [196]. Visual content has also been used in point of interest recommendations since photos and user-posted images contain large numbers of landmarks [197]. To reveal evolving fashion trends among users, He et al. modeled non-visual and visual dimensions with temporal dynamics and deep convolutional networks [198]. Jaradat proposed the transfer of knowledge between domains using two convolutional neural networks, one each for image and text, thus exploiting user preferences hidden in social media platforms such as Instagram [199].

Recommender system is required to be capable of profiling users from multimedia data, where visual information will be a significant component. Applications of multi-model fusion and multi-task learning in recommender systems are needed to comprehensively model user preferences. New functions such as cloth design and collocation are highly demanded in future fashion recommender systems.

### Future directions

Current developments in recommender systems focus on providing decision support with a wide range of information related to the metadata of items, images, social networks, and user-contributed reviews. In this paper, we have reviewed the various areas of AI that relate to such systems and chronicled their development. Given that the anticipated recommendation should always meet user requirements while also gaining a better understanding of what interests a broad range of users, we identify several emerging research aspects that will benefit from future research on recommender systems.

### Concept drift detection and reaction in recommender systems

Although recommender systems have achieved great success in the past, the complex and dynamic characteristics that are a feature of big data are not handled well in these systems [200]. Traditional recommender systems assume that user preference is relatively static over a period of time, so users' history records are weighted equally. However, user preferences change because of the gradual evolution of individual tastes, personal experiences or popularity-driven influences. This is a phenomenon commonly seen in Big Data streams and widely known as concept drift [201]. As a user's history records accumulate, older records may be inconsistent with the user's new requests. Using all the available data indiscriminately jeopardizes prediction accuracy, and recommender systems that fail to take this into consideration run the risk of performance degradation.

Time-aware recommender systems were developed to address this issue [202]. Most of the methods used in time-aware recommender systems tried to accommodate user-preference drift in their models without detecting the drift. Time-window and instance decay approach determine the weights of data instances along the timeline according to the principle that old data weighs less [203]. Besides penalizing the old data, some methods used dynamic matrix factorization, in which time is considered to be one more dimension of the data [204]. However, since these methods fail to detect the change, they cannot determine the direction of the change either, resulting in bias in the proposed adaptation and weighting decay. In the big data era, methods that

can manage temporal dynamics and can describe changes are required.

### Long tail in recommender systems (imbalanced data)

Long-tail items are items that are unpopular and seldom noticed by users. More attention should be paid by recommender systems to long-tail items, to help users discover them. Long-tail items are noticed less by users precisely because fewer data about them are collected, which results in these items being forgotten by users and e-commerce companies. When exploited, however, long-tail items can bring huge benefits to both customers and companies [205]. Cross-domain recommender systems offer a potential means to solve the long tail item problem because of their ability to transfer knowledge from related but different data from one domain to another domain even when the data are scarce. Therefore, recommender systems for long-tail items present great opportunities for future study.

### Privacy-preserving and secure recommender systems

The use of recommender systems grows widely into various application areas, which lead users to more concerns about their privacy. As a result, users are reluctant to provide authentic information and preferences when using the system, which on the other hand, impairs the performance of the recommender systems. The capability of evolutionary algorithms of covering multiple objectives enables its application in developing privacy-preserving recommender systems. One way to implement privacy by encryptions on the user profile, such as a distributed CF model with encrypted data [206]. The main concern of this method is its high computational cost. Another way is to transform user profiles and prevent the possible inference of user data. In [207] randomness is added to user data by perturbation so that privacy is preserved while keeping the accuracy of recommendation. How to preserve privacy is also studied on the CF method where similar users are clustered by data-independent hashing [208]. With more cross-platform systems developed, the development of privacy-preserving and secure recommender systems is intensively needed. The application of recommender systems in domains with high privacy risks such as healthcare or banking will prompt the development of privacy-preserving techniques.

### Recommender system visualization

Many recommender systems focus on methods and accuracy but lack adequate explanation. Although the performance of recommender systems is very good, users find them difficult

to trust due to opacity and privacy concerns. This is a challenging limitation in many recommender systems, especially those that are combined with complex artificial intelligence techniques such as deep learning or natural language processing.

Visualization is incorporated into recommender systems to provide a means for users to quickly and easily understand and interact with the system. Interactive and non-interactive strategies are compared in [209], illustrating how a visual interface can improve user satisfaction by providing explanatory notes. Several works have discussed possible options for visualizing and explaining the recommendation entity or process to users in traditional recommendation methods [210, 211], but the interpretation of how a system works for hybrid methods in which AI techniques are integrated is still lacking. It is necessary for systems to include a deeper illustration of the process and enhanced user interaction so that more works on recommender system visualization can be developed in the future.

### Summary

In this position paper, we review eight fields of AI, introduce their applications in recommender systems, discuss the open research issues, and give directions of possible future research on how AI techniques will be applied in recommender systems. This paper highlights how the recommender system can be enhanced by AI techniques and aims to provide guidance for researchers and practitioners in the area of recommender systems.

**Acknowledgements** The work presented in this paper was supported by the Australian Research Council (ARC) under the Australian Laureate Fellowship [FL190100149] and the UTS Distinguished Visiting Scholars (DVS) Scheme.

### Compliance with ethical standards

**Conflict of interest** On behalf of all authors, the corresponding author states that there is no conflict of interest.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

1. Shapira B, Ricci F, Kantor PB, Rokach L (2011) Recommender systems handbook. Springer, New York
2. Bobadilla J, Ortega F, Hernando A, Gutiérrez A (2013) Recommender systems survey. *Knowl Based Syst* 46:109–132
3. Ben Schafer J, Konstan J, Riedl J (1999) Recommender systems in e-commerce. In: *Proceedings of the 1st ACM Conference on Electronic Commerce, 1999*, pp 158–166
4. Lu J, Wu D, Mao M, Wang W, Zhang G (2015) Recommender system application developments: a survey. *Decis Support Syst* 74:12–32
5. Adomavicius G, Tuzhilin A (2005) Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Trans Knowl Data Eng* 17(6):734–749
6. Burke R (2002) Hybrid recommender systems: survey and experiments. *User Model User-adapt Interact* 12(4):331–370
7. Shardanand U, Maes P (1995) Social information filtering: algorithms for automating ‘word of mouth’. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 1995*, pp 210–217
8. Salton G, Wong A, Yang C-S (1975) A vector space model for automatic indexing. *Commun ACM* 18(11):613–620
9. Sebastiani F (2002) Machine learning in automated text categorization. *ACM Comput Surv* 34(1):1–47
10. Herlocker JL, Konstan JA, Terveen LG, Riedl JT (2004) Evaluating collaborative filtering recommender systems. *ACM Trans Inf Syst* 22(1):5–53
11. Lops P, De Gemmis M, Semeraro G (2011) Content-based recommender systems: state of the art and trends. *Recommender systems handbook*. Springer, Berlin, pp 73–105
12. Shambour Q, Lu J (2012) A trust-semantic fusion-based recommendation approach for e-business applications. *Decis Support Syst* 54(1):768–780
13. Balabanović M, Shoham Y (1997) Fab: content-based, collaborative recommendation. *Commun ACM* 40(3):66–72
14. Resnick P, Iacovou N, Suchak M, Bergstrom P, Riedl J (1994) GroupLens: an open architecture for collaborative filtering of news. In: *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work, 1994*, pp 175–186
15. Linden G, Smith B, York J (2003) Amazon.com recommendations: item-to-item collaborative filtering. *IEEE Internet Comput* 7(1):76–80
16. Liu H, Hu Z, Mian A, Tian H, Zhu X (2014) A new user similarity model to improve the accuracy of collaborative filtering. *Knowl Based Syst* 56:156–166
17. Hu Y, Zhang D, Ye J, Li X, He X (2013) Fast and accurate matrix completion via truncated nuclear norm regularization. *IEEE Trans Pattern Anal Mach Intell* 35(9):2117–2130
18. Su X, Khoshgoftar TM (2009) A survey of collaborative filtering techniques. *Adv Artif Intell* 2009:1–19
19. Deshpande M, Karypis G (2004) Item-based top-n recommendation algorithms. *ACM Trans Inf Syst* 22(1):143–177
20. Shi Y, Larson M, Hanjalic A (2014) Collaborative filtering beyond the user-item matrix: a survey of the state of the art and future challenges. *ACM Comput Surv* 47(1):3
21. Koren Y, Bell R, Volinsky C (2009) Matrix factorization techniques for recommender systems. *Computer* 42(8):30–37
22. Luo X, Zhou M, Li S, You Z, Xia Y, Zhu Q (2016) A nonnegative latent factor model for large-scale sparse matrices in recommender systems via alternating direction method. *IEEE Trans Neural Netw Learn Syst* 27(3):579–592
23. Liu B, Xiong H, Papadimitriou S, Fu Y, Yao Z (2015) A general geographical probabilistic factor model for point of interest recommendation. *IEEE Trans Knowl Data Eng* 27(5):1167–1179
24. Smyth B (2007) Case-based recommendation. *The adaptive web*. Springer, Berlin, pp 342–376
25. Aamodt A, Plaza E (1994) Case-based reasoning: foundational issues, methodological variations, and system approaches. *AI Commun* 7(1):39–59
26. Felfernig A, Friedrich G, Jannach D, Zanker M (2011) Developing constraint-based recommenders. *Recommender systems handbook*. Springer, Berlin, pp 187–215
27. Felfernig A, Burke R (2008) Constraint-based recommender systems: technologies and research issues. In: *Proceedings of the 10th International Conference on Electronic Commerce, 2008*, p 3
28. Luger GF (2005) Artificial intelligence: structures and strategies for complex problem solving. Pearson Education, London
29. Russell SJ, Norvig P (2016) Artificial intelligence: a modern approach. Pearson Education Limited, Malaysia
30. LeCun Y, Bengio Y (1995) Convolutional networks for images, speech, and time series. *Handb Brain Theor Neural Netw* 3361(10):1995
31. Lecun Y, Bengio Y, Hinton G (2015) Deep learning. *Nature* 521(7553):436–444
32. Goodfellow I, Bengio Y, Courville A (2016) Deep learning. MIT press, Cambridge
33. Murtagh F (1991) Multilayer perceptrons for classification and regression. *Neurocomputing* 2(5–6):183–197
34. Wang Y, Yao H, Zhao S (2016) Auto-encoder based dimensionality reduction. *Neurocomputing* 184:232–242
35. Lawrence S, Giles CL, Tsoi AC, Back AD (1997) Face recognition: a convolutional neural-network approach. *IEEE Trans Neural Netw* 8(1):98–113
36. Rumelhart DE, Hinton GE, Williams RJ (1986) Learning representations by back-propagating errors. *Nature* 323(6088):533–536
37. Hochreiter S, Schmidhuber J (1997) LSTM can solve hard long time lag problems. *Advances in neural information processing systems*. MIT Press, Cambridge, pp 473–479
38. Goodfellow I et al (2014) Generative adversarial nets. *Advances in neural information processing systems*. MIT Press, Cambridge, pp 2672–2680
39. Zhou J et al (2018) Graph neural networks: a review of methods and applications. *arXiv Prepr. arXiv1812.08434*
40. Lu J, Zuo H, Zhang G (2019) Fuzzy multiple-source transfer learning. *IEEE Trans. Fuzzy Syst*
41. Lu J, Behbood V, Hao P, Zuo H, Xue S, Zhang G (2015) Transfer learning using computational intelligence: a survey. *Knowl Based Syst* 80:14–23
42. Kang Z, Grauman K, Sha F (2011) Learning with whom to share in multi-task feature learning. In: *The 28th International Conference on Machine Learning*, pp 521–528
43. Arnold A, Nallapati R, Cohen WW (2007) A comparative study of methods for transductive transfer learning. In: *The 7th IEEE International Conference on Data Mining Workshops, 2007*, pp 77–82
44. Lu J, Xuan J, Zhang G, Luo X (2018) Structural property-aware multilayer network embedding for latent factor analysis. *Pattern Recogn* 76:228–241
45. Zhu X, Lafferty J, Rosenfeld R (2005) Semi-supervised learning with graphs. Carnegie Mellon University, Language Technologies Institute, School of Computer Science, Pittsburgh
46. Aghdam HH, Gonzalez-Garcia A, van de Weijer J, López AM (2019) Active learning for deep detection neural networks. In: *Proceedings of the IEEE International Conference on Computer Vision, 2019*, pp 3672–3680
47. Settles B (2011) From theories to queries: active learning in practice. In: *Active Learning and Experimental Design workshop in conjunction with AISTATS 2010, 2011*, pp 1–18

48. Settles B (2010) Active learning literature survey. University of California, Santa Cruz
49. Sutton RS, Barto AG (2011) Reinforcement learning: An introduction. MIT Press, Cambridge
50. Peng et al P (2017) Multiagent bidirectionally-coordinated nets: emergence of human-level coordination in learning to play starcraft combat games. arXiv Prepr. arXiv1703.10069
51. Bai W, Li T, Tong S (2020) NN reinforcement learning adaptive control for a class of nonstrict-feedback discrete-time systems. IEEE Trans Cybern
52. Hüttenrauch M, Adrian S, Neumann G (2019) Deep reinforcement learning for swarm systems. J Mach Learn Res 20(54):1–31
53. Neftci EO, Averbek BB (2019) Reinforcement learning in artificial and biological systems. Nat Mach Intell 1(3):133–143
54. Botvinick M, Ritter S, Wang JX, Kurth-Nelson Z, Blundell C, Hassabis D (2019) Reinforcement learning, fast and slow. Trends Cogn Sci 23(5):408–422
55. Bellman R (1957) A Markovian decision process. J Math Mech 679–684
56. Henderson P, Islam R, Bachman P, Pineau J, Precup D, Meger D (2017) Deep reinforcement learning that matters. arXiv Prepr. arXiv1709.06560
57. Kaelbling LP, Littman ML, Moore AW (1996) Reinforcement learning: a survey. J Artif Intell Res 4:237–285
58. Mnih V et al (2015) Human-level control through deep reinforcement learning. Nature 518(7540):529–533
59. Silver D et al (2016) Mastering the game of Go with deep neural networks and tree search. Nature 529(7587):484–489
60. Tran L, Duckstein L (2002) Comparison of fuzzy numbers using a fuzzy distance measure. Fuzzy Sets Syst 130(3):331–341
61. Roubos JA, Setnes M, Abonyi J (2003) Learning fuzzy classification rules from labeled data. Inf Sci (Ny) 150(1–2):77–93
62. Chen S-M, Wang C-Y (2013) Fuzzy decision making systems based on interval type-2 fuzzy sets. Inf Sci (Ny) 242:1–21
63. Holland JH (1975) Adaption in natural and artificial systems
64. Beyer H-G, Beyer H-G, Schwefel H-P, Schwefel H-P (2002) Evolution strategies: a comprehensive introduction. Nat Comput 1(1):3–52
65. Koza JR (1992) Genetic programming: on the programming of computers by means of natural selection. The MIT Press, Cambridge
66. Larrañaga P, Lozano JA (2001) Estimation of distribution algorithms: a new tool for evolutionary computation, vol 2. Springer Science & Business Media, Berlin
67. Storn R, Price K (1997) Differential evolution: a simple and efficient heuristic for global optimization over continuous spaces. J Glob Optim 11(4):341–359
68. Eberhart R, Kennedy J (1995) A new optimizer using particle swarm theory. In: Proceedings of the 6th International Symposium on Micro Machine and Human Science, 1995, pp 39–43
69. Dorigo M, Gambardella LM (1997) Ant colony system: a cooperative learning approach to the traveling salesman problem. IEEE Trans Evol Comput 1(1):53–66
70. Miettinen K (1999) Nonlinear multiobjective optimization. Kluwer Academic Publishers, Dordrecht
71. Li B, Li J, Tang K, Yao X (2015) Many-objective evolutionary algorithms: a survey. ACM Comput Surv 48(1):1–35
72. Chowdhary KR (2020) Natural language processing. Fundamentals of artificial intelligence. Springer, Berlin, pp 603–649
73. Chien J-T (2019) Deep Bayesian natural language processing. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts, 2019, pp 25–30
74. Collobert R, Weston J, Bottou L, Karlen M, Kavukcuoglu K, Kuksa P (2011) Natural language processing (almost) from scratch. J Mach Learn Res 12:2493–2537
75. Zhang W, Yoshida T, Tang X (2008) Text classification based on multi-word with support vector machine. Knowl Based Syst 21(8):879–886
76. Yi J, Nasukawa T, Bunescu R, Niblack W (2003) Sentiment analyzer: Extracting sentiments about a given topic using natural language processing techniques. In: Third IEEE international conference on data mining, 2003, pp 427–434
77. Blei DM, Ng AY, Jordan MI (2003) Latent dirichlet allocation. J. Mach. Learn. Res. 3:993–1022
78. Forsyth DA, Ponce J (2002) Computer vision: a modern approach. Prentice Hall Professional Technical Reference, Upper Saddle River
79. Vouliodimos A, Doulamis N, Doulamis A, Protopapadakis E (2018) Deep learning for computer vision: a brief review. Comput Intell Neurosci 2018:7068349
80. Khan S, Rahmani H, Shah SAA, Bennamoun M (2018) A guide to convolutional neural networks for computer vision. Synth Lect Comput Vis 8(1):1–207
81. Salakhutdinov R, Mnih A, Hinton G (2007) Restricted Boltzmann machines for collaborative filtering. In: Proceedings of the 24th International Conference on Machine Learning, 2007, pp 791–798
82. Truyen TT, Phung DQ, Venkatesh S (2009) Ordinal Boltzmann machines for collaborative filtering. In: Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence, 2009, pp 548–556
83. Zhang S, Yao L (2017) Deep learning based recommender system: a survey and new perspectives. ACM J Comput Cult Herit Artic 1(35):1–35
84. Cheng et al. HT (2016) Wide and deep learning for recommender systems. arXiv Prepr. pp 1–4
85. Guo H, Tang R, Ye Y, Li Z, He X (2017) DeepFM: a factorization-machine based neural network for CTR prediction. In: International Joint Conference on Artificial Intelligence, 2017, pp 1725–1731
86. He X, Liao L, Zhang H, Nie L, Hu X, Chua TS (2017) Neural collaborative filtering. In: Proceedings of the 26th International Conference on World Wide Web, 2017, pp 173–182
87. Sedhain S, Menon AK, Sanner S, Xie L (2015) AutoRec: autoencoders meet collaborative filtering. In: Proceedings of the 24th International Conference on World Wide Web, 2015, pp 111–112
88. Zhang S, Yao L, Xu X (2017) AutoSVD++: an efficient hybrid collaborative filtering model via contractive auto-encoders. In: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2017, pp 2–5
89. Strub F, Gaudel R, Mary J (2016) Hybrid recommender system based on autoencoders. In: Proceedings of the 1st Workshop on Deep Learning for Recommender Systems, 2016, pp 1–5
90. Diao Q, Qiu M, Wu CY, Smola AJ, Jiang J, Wang C (2014) Jointly modeling aspects, ratings and sentiments for movie recommendation. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2014, pp 193–202
91. Kim D, Park C, Oh J, Lee S, Yu H (2016) Convolutional matrix factorization for document context-aware recommendation. In: RecSys 2016—Proceedings of the 10th ACM Conference on Recommender Systems, 2016, pp 233–240
92. Yuyun G, Qi Z (2016) Hashtag recommendation using attention-based convolutional neural network. In: International Joint Conference on Artificial Intelligence, 2016, pp 2782–2788
93. Dai H, Wang Y, Trivedi R, Song L (2016) Recurrent coevolutionary feature embedding processes for recommendation. In: Proceedings of the 1st Workshop on Deep Learning for Recommender Systems, 2016, pp 1–11
94. Wu CY, Ahmed A, Beutel A, Smola AJ, Jing H (2017) Recurrent recommender networks. In: Proceedings of the 10th ACM Inter-



- national Conference on Web Search and Data Mining, 2017, pp 495–503
95. Jing H, Smola AJ (2017) Neural survival recommender. In: Proceedings of the 10th ACM International Conference on Web Search and Data Mining, 2017, pp 515–524
  96. Wang S, Hu L, Wang Y, Cao L, Sheng QZ, Orgun M (2019) Sequential recommender systems: challenges, progress and prospects. arXiv Prepr. arXiv2001.04830
  97. Hidasi B, Karatzoglou A, Baltrunas L, Tikk D (2016) Session-based recommendations with recurrent neural networks. In: 4th Int. Conf. Learn. Represent, pp 1–10, 2016
  98. Wu S, Ren W, Yu C, Chen G, Zhang D, Zhu J (2016) Personal recommendation using deep recurrent neural networks in NetEase. In: Proceeding of the 32nd International Conference on Data Engineering, 2016, pp 1218–1229
  99. Li J, Ren R, Chen Z, Ren Z, Lian T, Ma J (2017) Neural attentive session-based recommendation. In: Int. Conf. Inf. Knowl. Manag. Proc., vol. Part F1318, pp 1419–1428, 2017
  100. Liu Q, Zeng Y, Mokhosi R, Zhang H (2018) STAMP: Short-term attention/memory priority model for session-based recommendation. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2018, pp. 1831–1839
  101. Ying H et al. (2018) Sequential recommender system based on hierarchical attention network. In: International Joint Conference on Artificial Intelligence, 2018.
  102. Wang J et al. (2017) IRGAN: a minimax game for unifying generative and discriminative information retrieval models. In: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2017, pp 515–524
  103. He X, He Z, Du X, Chua TS (2018) Adversarial personalized ranking for recommendation. In: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, 2018, pp 355–364
  104. Yang D, Guo Z, Wang Z, Jiang J, Xiao Y, Wang W (2018) A knowledge-enhanced deep recommendation framework incorporating GAN-based models. In: 2018 IEEE International Conference on Data Mining, 2018, pp 1368–1373
  105. Tang J, Du X, He X, Yuan F, Tian Q, Chua T-S (2019) Adversarial training towards robust multimedia recommender system. IEEE Trans Knowl Data Eng 32(5):855–867
  106. Ying R, He R, Chen K, Eksombatchai P, Hamilton WL, Leskovec J (2018) Graph convolutional neural networks for web-scale recommender systems. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2018, pp 974–983
  107. Yin R, Li K, Zhang G, Lu J (2019) A deeper graph neural network for recommender systems. Knowl Based Syst 185:105020
  108. Wu S, Tang Y, Zhu Y, Wang L, Xie X, Tan T (2019) Session-based recommendation with graph neural networks. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2019, vol. 33, pp 346–353
  109. Cantador I, Fernández-Tobías I, Berkovsky S, Cremonesi P (2015) Cross-domain recommender systems. Recommender systems handbook. Springer, Berlin, pp 919–959
  110. Singh AP, Gordon GJ (2008) Relational learning via collective matrix factorization. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2008, pp 650–658
  111. Yang D, He J, Qin H, Xiao Y, Wang W (2015) A graph-based recommendation across heterogeneous domains, In: Proceedings of the 24th ACM International Conference on Information and Knowledge Management, 2015, pp 463–472
  112. Abel F, Herder E, Houben G-J, Henze N, Krause D (2013) Cross-system user modeling and personalization on the social web. User Model. User-adapt. Interact, pp 1–41
  113. Zhen Y, Li WJ, Yeung DY (2009) TagiCoFi: Tag informed collaborative filtering. In: RecSys'09—Proceedings of the 3rd ACM Conference on Recommender Systems, 2009, pp 69–76
  114. Hao P, Zhang G, Martinez L, Lu J (2017) Regularizing knowledge transfer in recommendation with tag-inferred correlation. IEEE Trans Cybern
  115. Li B, Yang Q, Xue X (2009) Can movies and books collaborate? Cross-domain collaborative filtering for sparsity reduction. In: Proceedings of the 21st International Joint Conference on Artificial Intelligence, 2009, vol 9, pp 2052–2057
  116. Li B, Yang Q, Xue X (2009) Transfer learning for collaborative filtering via a rating-matrix generative model. In: Proceedings of the 26th International Conference on Machine Learning, ICML 2009, 2009, pp 617–624
  117. Zhang Q, Wu D, Lu J, Liu F, Zhang G (2017) A cross-domain recommender system with consistent information transfer. Decis Support Syst 104:49–63
  118. Zhang Y, Cao B, Yeung DY (2010) Multi-domain collaborative filtering. In: Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence, 2010, pp 725–732
  119. Pan W, Yang Q (2013) Transfer learning in heterogeneous collaborative filtering domains. Artif Intell 197:39–55
  120. Hu L, Cao J, Xu G, Cao L, Gu Z, Zhu C (2013) Personalized recommendation via cross-domain triadic factorization. In: Proceedings of the 22nd International Conference on World Wide Web, 2013, pp 595–606
  121. Mirbakhsh N, Ling CX (2015) Improving top-n recommendation for cold-start users via cross-domain information. ACM Trans Knowl Discov Data 9(4):33
  122. Li CY, Lin SD (2014) Matching users and items across domains to improve the recommendation quality. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2014, pp 801–810
  123. Zhao L, Pan SJ, Yang Q (2017) A unified framework of active transfer learning for cross-system recommendation. Artif Intell 245:38–55
  124. Zhang Q, Lu J, Wu D, Zhang G (2019) A cross-domain recommender system with kernel-induced knowledge transfer for overlapping entities. IEEE Trans Neural Netw Learn Syst 30(7):1998–2012
  125. Zhu F, Wang Y, Chen, Liu G, Orgun M, Wu (2018) A deep framework for cross-domain and cross-system recommendations. In: IJCAI International Joint Conference on Artificial Intelligence, 2018
  126. Hu G, Zhang Y, Yang Q (2018) Conet: collaborative cross networks for cross-domain recommendation. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, 2018, pp 667–676
  127. Wang C, Niepert M, Li H (2019) Recsys-dan: discriminative adversarial networks for cross-domain recommender systems. IEEE Trans Neural Networks Learn Syst
  128. Yuan F, Yao L, Benatallah B (2019) DAREc: Deep domain adaptation for cross-domain recommendation via transferring rating patterns. In: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, 2019
  129. Zhu F, Chen C, Wang Y, Liu G, Zheng X (2019) Dtcdr: a framework for dual-target cross-domain recommendation. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, 2019, pp 1533–1542
  130. Elahi M, Ricci F, Rubens N (2016) A survey of active learning in collaborative filtering recommender systems. Comput Sci Rev 20:29–50

131. Boutilier C, Zemel RS, Marlin B (2003) Active collaborative filtering. In: Proceedings of the 19th Conference on Uncertainty in Artificial Intelligence, 2003, pp 98–106
132. Mello CE, Aufaure MA, Zimbrao G (2010) Active learning driven by rating impact analysis. In: Proceedings of the 4th ACM Conference on Recommender Systems, 2010, pp 341–344
133. Golbandi N, Koren Y, Lempel R (2010) On bootstrapping recommender systems. In: Proceedings of the 19th ACM International Conference on Information and Knowledge Management, 2010, pp 1805–1808
134. Karimi R, Freudenthaler C, Nanopoulos A, Schmidt-Thieme L (2011) Active learning for aspect model in recommender systems. In: IEEE Symposium on Computational Intelligence and Data Mining, 2011, pp 162–167
135. Golbandi N, Koren Y, Lempel R (2011) Adaptive bootstrapping of recommender systems using decision trees. In: Proceedings of the 4th ACM International Conference on Web Search and Data Mining, 2011, pp 595–604
136. Karimi R, Freudenthaler C, Nanopoulos A, Schmidt-Thieme L (2011) Non-myopic active learning for recommender systems based on matrix factorization. In: IEEE International Conference on Information Reuse & Integration, 2011, pp 299–303
137. Wiesner M, Pfeifer D (2010) Adapting recommender systems to the requirements of personal health record systems. In: Proceedings of the 1st ACM International Health Informatics Symposium, 2010, pp 410–414
138. Elahi M, Ricci F, Rubens N (2012) Adapting to natural rating acquisition with combined active learning strategies. In: International Symposium on Methodologies for Intelligent Systems, 2012, pp 254–263
139. Rubens N, Sugiyama M (2007) Influence-based collaborative active learning. In: Proceedings of the 1st ACM Conference on Recommender Systems, 2007, pp 145–148
140. He L, Liu NN, Yang Q (2011) Active dual collaborative filtering with both item and attribute feedback. In: Proceedings of the National Conference on Artificial Intelligence, 2011, vol. 2, pp 1186–1191
141. Zhang Z, Jin X, Li L, Ding G, Yang Q (2016) Multi-domain active learning for recommendation. In: AAAI, 2016, pp 2358–2364
142. Berry DA, Fristedt B (1985) Bandit problems: sequential allocation of experiments (Monographs on statistics and applied probability). London Chapman Hall 5(71–87):7
143. Shani G, Heckerman D, Brafman RI (2005) An MDP-based recommender system. *J Mach Learn Res* 6:1265–1295
144. Warlop R et al (2018) Fighting boredom in recommender systems with linear reinforcement learning. No. NeurIPS, 2018
145. Wang H, Wu Q, Wang H (2017) Factorization bandits for interactive recommendation. *AAAI* 17:2695–2702
146. Li L, Chu W, Langford J, Schapire RE (2010) A contextual-bandit approach to personalized news article recommendation. In: Proc. 19th Int. Conf. World Wide Web, pp. 661–670, 2010
147. Zeng C, Wang Q, Mokhtari S, Li T (2016) Online context-aware recommendation with time varying multi-armed bandit. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp 2025–2034
148. Zheng G et al (2018) DRN: a deep reinforcement learning framework for news recommendation. *Proc World Wide Web Conf* 2:167–176
149. Zhao X, Xia L, Zhang L, Ding Z, Yin D, Tang J (2018) Deep reinforcement learning for page-wise recommendations. In: 12th ACM Conf Recomm Syst, pp 95–103, 2018
150. Zhao X, Xia L, Zhang L, Tang J, Ding Z, Yin D (2018) Recommendations with negative feedback via pairwise deep reinforcement learning. In: Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min., pp. 1040–1048, 2018
151. Zhou S et al (2020) Interactive recommender system via knowledge graph-enhanced reinforcement learning. pp 179–188
152. Ie E et al (2019) SLateq: a tractable decomposition for reinforcement learning with recommendation sets. In: *Int Jt Conf Artif Intell*, vol. 2019-Augus, pp 2592–2599, 2019
153. Hu Y, Da Q, Zeng A, Yu Y, Xu Y (2018) Reinforcement learning to rank in E-commerce search engine: Formalization, analysis, and application. In: *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, pp 368–377, 2018
154. Chung F, Rhee H (2007) “Uncertain fuzzy clustering: insights and recommendations. *IEEE Comput Intell Mag* 2(1):44–56
155. Yager RR (2003) Fuzzy logic methods in recommender systems. *Fuzzy Sets Syst* 136(2):133–149
156. Zenebe A, Zhou L, Norcio AF (2010) User preferences discovery using fuzzy models. *Fuzzy Sets Syst* 161(23):3044–3063
157. Mao M, Lu J, Zhang G, Zhang J (2015) A fuzzy content matching-based e-commerce recommendation approach. In: *IEEE International Conference on Fuzzy Systems*, 2015
158. Wu D, Zhang G, Lu J (2015) A fuzzy preference tree-based recommender system for personalized business-to-business e-services. *IEEE Trans Fuzzy Syst* 23(1):29–43
159. Zhang Z, Lin H, Liu K, Wu D, Zhang G, Lu J (2013) A hybrid fuzzy-based personalized recommender system for telecom products/services. *Inf Sci (Ny)* 235:117–129
160. Yera R, Castro J, Martínez L (2016) A fuzzy model for managing natural noise in recommender systems. *Appl Soft Comput J* 40:187–198
161. Cornelis C, Lu J, Guo X, Zhang G (2007) One-and-only item recommendation with fuzzy logic techniques. *Inf Sci (Ny)* 177(22):4906–4921
162. Son LH, Thong NT (2015) Intuitionistic fuzzy recommender systems: an effective tool for medical diagnosis. *Knowl Based Syst* 74:133–150
163. Zhang Q, Wu D, Zhang G, Lu J (2016) Fuzzy user-interest drift detection based recommender systems. In: *International Conference on Fuzzy Systems*, 2016, pp 1274–1281
164. Nilashi M, Bin-Ibrahim O, Ithnin N (2014) “Multi-criteria collaborative filtering with high accuracy using higher order singular value decomposition and neuro-fuzzy system. *Knowl Based Syst* 60:82–101
165. Nilashi M, Bin-Ibrahim O, Ithnin N (2014) Hybrid recommendation approaches for multi-criteria collaborative filtering. *Expert Syst Appl* 41(8):3879–3900
166. Treerattanapitak K, Jaruskulchai C (2012) Exponential fuzzy C-means for collaborative filtering. *J Comput Sci Technol* 27(3):567–576
167. Xu S, Watada J (2014) A method for hybrid personalized recommender based on clustering of fuzzy user profiles. In: *IEEE International Conference on Fuzzy Systems*, 2014, pp 2171–2177
168. Kant V, Bharadwaj KK (2013) Integrating collaborative and reclusive methods for effective recommendations: a fuzzy Bayesian approach. *Int J Intell Syst* 28(11):1099–1123
169. de Campos LM, Fernández-Luna JM, Huete JF (2008) A collaborative recommender system based on probabilistic inference from fuzzy observations. *Fuzzy Sets Syst* 159(12):1554–1576
170. Serrano-Guerrero J, Herrera-Viedma E, Olivas JA, Cerezo A, Romero FP (2011) A Google wave-based fuzzy recommender system to disseminate information in University Digital Libraries 2.0. *Inf Sci (Ny)* 181(9):1503–1516
171. Bedi P, Vashisth P (2014) Empowering recommender systems using trust and argumentation. *Inf Sci (Ny)* 279(22):569–586
172. Zhang X, Duan F, Zhang L, Cheng F, Jin Y, Tang K (2017) Pattern recommendation in task-oriented applications: a multi-objective perspective. *IEEE Computational Intelligence Magazine*, vol. 12, no. 3, IEEE, pp 43–53, 2017

173. Ribeiro MT, Lacerda A, Veloso A, Ziviani N (2012) Pareto-efficient hybridization for multi-objective recommender systems. In: Proceedings of the 6th ACM Conference on Recommender Systems, 2012, pp 19–26
174. Rodriguez M, Posse C, Zhang E (2012) Multiple objective optimization in recommender systems. In: Proceedings of the 6th ACM Conference on Recommender Systems, 2012, pp 11–18
175. Karabadi NEI, Beldjoudi S, Seridi H, Aridhi S, Dhifli W (2018) Improving memory-based user collaborative filtering with evolutionary multi-objective optimization. *Expert Syst Appl* 98:153–165
176. Mu C, Jiao L, Liu Y, Li Y (2015) Multiobjective nondominated neighbor coevolutionary algorithm with elite population. *Soft Comput* 19(5):1329–1349
177. Rana C, Jain SK (2015) A study of the dynamic features of recommender systems. *Artif Intell Rev* 43(1):141–153
178. Chen Y, Sun X, Gong D, Zhang Y, Choi J, Klasky S (2017) Personalized search inspired fast interactive estimation of distribution algorithm and its application. *IEEE Trans Evol Comput* 21(4):588–600
179. Adomavicius G, Kwon Y (2015) Multi-criteria recommender systems. *Recommender systems handbook*. Springer, Berlin, pp 847–880
180. Konečný J, McMahan HB, Yu FX, Richtárik P, Suresh AT, Bacon D (2016) Federated learning: strategies for improving communication efficiency. *arXiv Prepr. arXiv1610.05492*
181. Huang L, Joseph AD, Nelson B, Rubinstein BIP, Tygar JD (2011) Adversarial machine learning. In: Proceedings of the 4th ACM Workshop on Security and Artificial Intelligence, 2011, pp 43–58
182. Zhu H, Jin Y (2020) Multi-objective evolutionary federated learning. *IEEE Trans Neural Netw Learn Syst* 31(4):1310–1322
183. Zhang W, Ding G, Chen L, Li C, Zhang C (2013) Generating virtual ratings from chinese reviews to augment online recommendations. *ACM Trans Intell Syst Technol* 4(1):1–17
184. Agarwal D, Chen BC (2010) fLDA: matrix factorization through latent Dirichlet allocation. In: Proceedings of the 3rd ACM International Conference on Web Search and Data Mining, 2010, pp 91–100
185. Dong R, Schaal M, O'Mahony MP, McCarthy K, Smyth B (2013) Sentimental product recommendation. In: Proceedings of the 7th ACM Conference on Recommender Systems, 2013, pp 44–58
186. McAuley J, Leskovec J (2013) From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews. In: Proc. 22nd Int. Conf. World Wide Web, pp 897–908
187. McAuley J, Leskovec J (2013) Hidden factors and hidden topics: understanding rating dimensions with review text. In: Proceedings of the 7th ACM Conference on Recommender Systems, 2013, pp 165–172
188. Ling G, Lyu MR, King I (2014) Ratings meet reviews, a combined approach to recommend. In: Proceedings of the 8th ACM Conference on Recommender Systems, 2014, pp 105–112
189. Xin X, Liu Z, Lin CY, Huang H, Wei X, Guo P (2015) Cross-domain collaborative filtering with review text. In: International Joint Conference on Artificial Intelligence, 2015, pp 1827–1834
190. Barkan O, Noam K (2016) Item2vec: neural item embedding for CF. In: IEEE 26th International Workshop on Machine Learning for Signal Processing, 2016, pp 1–6
191. Sun Z, Yang J, Zhang J, Bozzon A, Chen Y, Xu C (2017) MRLR: multi-level representation learning for personalized ranking in recommendation. In: International Joint Conference on Artificial Intelligence, 2017, pp 2807–2813
192. Iovine A, Narducci F, Semeraro G (2020) Conversational recommender systems and natural language: a study through the ConveRSE framework. *Decis Support Syst* 131:113250
193. Lei C, Liu D, Li W, Zha ZJ, Li H (2016) Comparative deep learning of hybrid representations for image recommendations. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp 2545–2553
194. He R, McAuley J (2015) VBPR: visual bayesian personalized ranking from implicit feedback. In: AAAI, 2015, pp 144–150
195. Gaspar P (2017) User preferences analysis using visual stimuli. In: Proceedings of the 11th ACM Conference on Recommender Systems, 2017, pp 436–440
196. Zhao L, Lu Z, Pan SJ, Yang Q (2016) Matrix factorization+ for movie recommendation. In: International Joint Conference on Artificial Intelligence, 2016, pp 3945–3951
197. Wang S, Wang Y, Tang J, Shu K, Ranganath S, Liu H (2017) What your images reveal: exploiting visual contents for point-of-interest recommendation. In: Proceedings of the 26th International Conference on World Wide Web, 2017, pp 391–400
198. He R, McAuley J (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, 2016, pp 507–517
199. Jaradat S (2017) Deep cross-domain fashion recommendation. In: Proceedings of the 11th ACM Conference on Recommender Systems, 2017, pp 407–410
200. Lu J, Liu A, Song Y, Zhang G (2020) Data-driven decision support under concept drift in streamed big data. *Complex Intell Syst* 6(1):157–163
201. Harries M, Horn K (1995) Detecting concept drift in financial time series prediction using symbolic machine learning. In: Proceedings of the 8th Australian Joint Conference on Artificial Intelligence, 1995, pp 91–98
202. Campos PG, Díez F, Cantador I (2014) Time-aware recommender systems: a comprehensive survey and analysis of existing evaluation protocols. *User Model User-Adapt Interact* 24(1–2):67–119
203. Yin H, Cui B, Chen L, Hu Z, Zhou X (2015) Dynamic user modeling in social media systems. *ACM Trans Inf Syst* 33(3):10
204. Chua FCT, Oentaryo RJ, Lim EP (2013) Modeling temporal adoptions using dynamic matrix factorization. In: Proceedings of IEEE International Conference on Data Mining, 2013, pp 91–100
205. Yin H, Cui B, Li J, Yao J, Chen C (2012) Challenging the long tail recommendation. In: Proceedings of the VLDB Endowment, 2012, vol 5, no 9, pp 896–907
206. Canny J (2002) Collaborative filtering with privacy. In: Proc. IEEE Symp. Secur. Priv., vol. 2002-Jan, pp 45–57, 2002
207. Kikuchi H, Mochizuki A (2013) Privacy-preserving collaborative filtering using randomized response. *J Inf Process* 21(4):617–623
208. Chow R, Pathak MA, Wang C (2012) A practical system for privacy-preserving collaborative filtering. In: Proc. 12th IEEE Int. Conf. Data Min. Work. ICDMW 2012, pp 547–554, 2012
209. Bostandjiev S, O'Donovan J, Höllerer T (2012) TasteWeights: a visual interactive hybrid recommender system. In: Proceedings of the sixth ACM conference on Recommender systems, 2012, pp 35–42
210. Wang W, Zhang G, Lu J (2017) Hierarchy visualization for group recommender systems. In: *IEEE Trans Syst Man Cybern Syst*, pp 1–12, 2017
211. Hernando A, Moya R, Ortega F, Bobadilla J (2014) Hierarchical graph maps for visualization of collaborative recommender systems. *J Inf Sci* 40(1):97–106