Research Commentary on Recommendations with Side Information: A Survey and Research Directions

Zhu Sun[1], Qing Guo[1], Jie Yang[2], Hui Fang[3], Guibing Guo[4], Jie Zhang[1], Robin Burke[5*]
[1]Nanyang Technological University, Singapore; [2]Amazon Research, USA
[3]Shanghai University of Finance and Economics, China
[4]Northeastern University, China; [5]University of Colorado, USA

ABSTRACT

Recommender systems have become an essential tool to help resolve the *information overload* problem in recent decades. Traditional recommender systems, however, suffer from *data sparsity* and *cold start* problems. To address these issues, a great number of recommendation algorithms have been proposed to leverage side information of users or items (e.g., social network and item category), demonstrating a high degree of effectiveness in improving recommendation performance. This Research Commentary aims to provide a comprehensive and systematic survey of the recent research on recommender systems with side information. Specifically, we provide an overview of state-of-the-art recommendation algorithms with side information from two orthogonal perspectives. One involves the different methodologies of recommendation: the memory-based methods, latent factor, representation learning and deep learning models. The others cover different representations of side information, including structural data (flat, network, and hierarchical features, and knowledge graphs); and non-structural data (text, image and video features). Finally, we discuss challenges and provide new potential directions in recommendation, along with the conclusion of this survey.

KEYWORDS

Research commentary; Recommender systems; Side information; Memory-based methods; Latent factor models; Representation learning; Deep learning; Flat features; Social networks; Feature hierarchies; Knowledge graphs

1 INTRODUCTION

With the advent of the era of big data, the volume of information on the web has increased in an exponential fashion. Users are submerged in the flood of countless products, news, movies, etc. Aiming to provide personalized recommendation services for users based on their historical interaction data, recommender systems have become a vital and indispensable tool to help tackle the *information overload* problem (Ricci et al. 2015; Desrosiers and Karypis 2011). Empirical studies have demonstrated the effectiveness in facilitating decision-making process and boosting business across various domains (Zhang et al. 2017c; Song et al. 2012; Adomavicius and Tuzhilin 2005), such as e-commerce (Amazon, Target, Taobao), point-of-interest (Foursquare, Yelp, Groupon), and multi-media (Youtube, Pinterest, Spotify), to name a few. Fig. 1 summarizes popular online services in various domains where recommender systems have been launched to improve the user experience. The

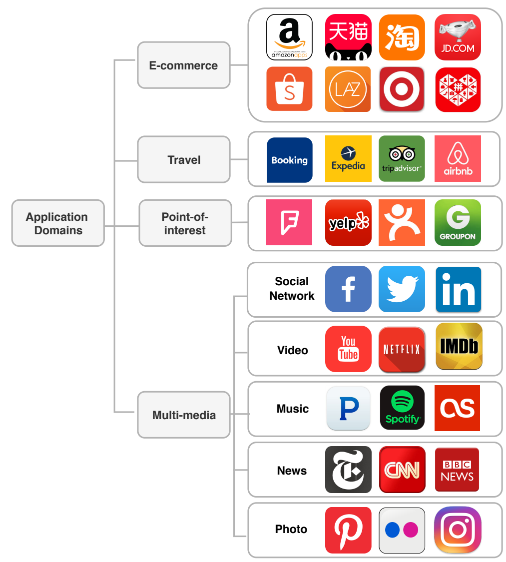


Figure 1: Popular apps that are utilized in various domains with recommender systems.

great blossoming of recommender systems in practical applications has been promoted, to a large extent, by the flourishing research on recommendation. For example, recommender systems have become an important topic in a number of top tier research conferences and journals[1].

The number of publications on recommender systems has increased dramatically in the last few years. RecSys (recsys.acm.org), the leading international conference on recommender systems, has continously attracted a tremendous amount of interest from both academia and industry (Cheng et al. 2016; Covington et al. 2016; Davidson et al. 2010; Gomez-Uribe and Hunt 2016; Okura et al. 2017). Among the different recommender systems, most of them are based on *collaborative filtering* (CF), which is one of the most successful techniques for recommendation (Schafer et al. 2007; Ekstrand et al. 2011; Bobadilla et al. 2013; Shi et al. 2014). Traditional

*Hui Fang is the corresponding author. Email addresses of all authors are as follows: {zhu.sun,qguo006,zhangj}@ntu.edu.sg; jiy@amazon.com; fang.hui@mail.shufe.edu.cn; guogb@swc.neu.edu.cn; robin.burke@colorado.edu

[1]Some of the key conferences and journals include NIPS (Neural Information Processing Systems), ICML (International Conference on Machine Learning), WWW (World Wide Web Conference), WSDM (Conference on Web Search and Data Mining), KDD (Conference on Knowledge Discovery and Data Mining), SIGIR (Conference on Research and Development in Information Retrieval), CIKM (International Conference on Information and Knowledge Management), IJCAI (International Joint Conferences on Artificial Intelligence), AAAI (Conference on Artificial Intelligence), UAI (Conference on Uncertainty in Artificial Intelligence), RecSys (Conference on Recommender Systems), ICLR (International Conference on Learning Representations), and TKDE (IEEE Transactions on Knowledge and Data Engineering), TOIS (ACM Transactions on Information Systems), CSUR (ACM Computing Surveys), etc.

CF-based methods rely on user-item interaction matrices for making recommendations, assuming that a user's preference can be inferred by aggregating the tastes of similar users. They have been widely investigated (Linden et al. 2003; Adomavicius and Tuzhilin 2005; Ekstrand et al. 2011; Koren et al. 2009; Mnih and Salakhutdinov 2008; Rendle et al. 2009), with various variants of CF-based methods developed (Adomavicius and Tuzhilin 2005; Ekstrand et al. 2011; Koren et al. 2009; Mnih and Salakhutdinov 2008; Rendle et al. 2009). Despite that, traditional CF-based methods are confronted with two fundamental issues when only the user-item interaction matrices are taken into consideration:

- *Data sparsity.* Usually, users face an extremely large amount of items to choose from. Even the most active users only rate a small set of items and most items have a very limited amount of feedback from users. This sparsity issue makes it hard for recommender systems to learn users' preferences.
- *Cold start.* It is a critical issue for both new users and items. Without historical data, it is difficult to generate decent recommendations. As a common solution, popular items might be recommended to new users, which will fail to create personalized recommendations.

To address the two issues, different types of side information, such as social networks, user profiles and item descriptions, have been utilized for recommender systems in various domains (Guo et al. 2019) (see Fig. 1). For instance, due to the emergence of social networks, a number of trust-aware recommendation algorithms (Ma et al. 2009a; Jamali and Ester 2010; Ma et al. 2011b; Yang et al. 2012; Guo et al. 2012; Yang et al. 2013a; Fang et al. 2014; Guo et al. 2015b) have been proposed based on the assumption that users share similar preferences with their trusted friends. For example, for restaurants, users often have meals with their trusted friends (Ye et al. 2010; Yang et al. 2013b). Besides social information, the side information for items (e.g., categories, genres, locations and brands) provides an in-depth understanding of both item properties and user preferences. Many recommendation approaches (Kim and Kim 2003; Shi et al. 2011; Koenigstein et al. 2011; Kanagal et al. 2012; Hu et al. 2014; Sun et al. 2017c; Sun et al. 2018) have been proposed by exploiting that kind of item information. Fig. 2 depicts an example of how side information facilitates the generation of more accurate recommendations for users. Regardless of either user or item side information, the evolution of recommendation approaches with side information – especially with the emergence and rapid development of deep learning based approaches, which have superior scalability and flexibility to accommodate arbitrary side information – has proven to be able to achieve great success with resolving the *data sparsity* and *cold start* problems, thus boosting recommendation performance.

Differences between this research commentary and other surveys. Due to the effectiveness of side information for recommender systems, the number of recent research studies have exploded in this field. And, there are also quite a few survey papers published on recommender systems. For instance, earlier works endeavored to conduct literature reviews on collaborative filtering techniques (Sarwar et al. 2001; Breese et al. 1998; Adomavicius and Tuzhilin 2005; Schafer et al. 2007; Su and Khoshgoftaar 2009; Desrosiers and Karypis 2011; Ekstrand et al. 2011; Bobadilla et al.

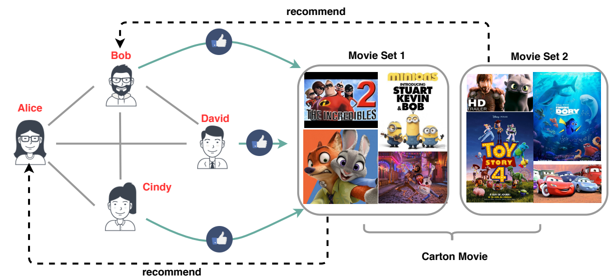


Figure 2: A toy example on leveraging user and item side information (social networks and movie genres) for more accurate recommendations. For instance, Alice has social connections with her friends, Bob, Cindy and David. As all her friends liked movies in Movie Set 1 (e.g., Zootopia and Coco), Alice would also be more likely to favor these movies in Movie Set 1. Besides, as the movies in both Movie Set 1 and Movie Set 2 belong to the genre of Carton movies, Bob would also prefer movies in Movie Set 2 (e.g., Toy Story and Cars), given that he liked movies in Movie Set 1.

2013; Ricci et al. 2015). Lops et al. (2011) provided a review on the state-of-the-arts and trends in content-based recommender systems. Burke (2002) presented a survey on hybrid recommender systems. Bellogín et al. (2013) introduced an empirical comparison of social, collaborative filtering, and hybrid recommenders. Gomez-Uribe and Hunt (2016) and Song et al. (2012) discussed various algorithms in recommending movie (Netflix) and music, respectively. Zhang et al. (2019) proposed a comprehensive review on how deep learning based algorithms are applied for recommender systems. And finally, Shi et al. (2014) provided a systematic review on how side information is employed in collaborative filtering based approaches.

Existing survey papers have mainly focused on a single perspective, instead of conducting a thorough investigation. In other words, they either discussed the general methodologies for recommender systems (e.g., Zhang et al. 2019; Gomez-Uribe and Hunt 2016) or side information per se (e.g., Shi et al. 2014), but ignored to explore the inherent dependency between them that together leads to high-quality recommendations. As a matter of fact, on the one hand, there are plenty of recent research efforts on dealing with the complexity of side information for realizing its full potential for better recommendations. Throughout our investigation, we discovered that existing research studies have been exploring more sophisticated structures to represent various kind of side information, including flat, network, hierarchical features and knowledge graphs. The different structures encode important relationships among the side information. For example, category hierarchies can reflect the affiliation among categories, whereas the flat structure of the category does not have such a property. Such a relationship can be of high value for improving recommendation performance.

On the other hand, many research studies have proposed more advanced recommendation methodologies to accommodate the diverse side information, evolving from memory-based methods to latent factor, representation learning and deep learning models. Based on our literature review, recommendation performance depends on both the structures representing the rich side information

Table 1: **Classifications of recommender systems from different perspectives.**

Perspective	Strategies	Tasks	Outputs
Category	• Content-based filtering • Collaborative filtering • Hybrid methods	• General • Temporal • Sequential	• Rating Prediction • Item Ranking

and the fundamental recommendation methodologies of employing them. The more complex representation of side information often needs to be coupled with more advanced methodologies to fully exploit the value of side information. In other words, it is often impossible to disentangle the useful side information from the fundamental methodologies for better recommendations.

This survey seeks to provide the research community a comprehensive and systematic overview of current progress in the recommendation area by considering both the representation of side information and the fundamental recommendation methodologies. It should not only focus on some cutting-edge techniques (e.g., knowledge graphs and deep learning models), but also other conventional ones (e.g., social networks and latent factor models) which have been the cornerstone in the development of recommender systems with side information. In this way, this Research Commentary provides a complete picture for both researchers and practitioners in this area.

Article collection. To cover recent studies, we collected hundreds of papers published in prestigious international conferences and journals related to recommender systems, including NIPS, ICML, UAI, KDD, WWW, WSDM, IJCAI, AAAI, SIGIR, RecSys, CIKM, ICLR, and TKDE, TOIS, CSUR, etc. Google Scholar was primarily used to searching for papers while other academic search engines were also adopted, such as ACM Digital Library (dl.acm.org), IEEE Xplore (ieeexplore.ieee.org), Web of Science (www.webofknowledge.com), and Springer (www.springer.com). A number of keywords and their combinations were utilized to search for related papers, including recommender systems, recommendations, side information, auxiliary information, social networks, feature hierarchies, knowledge graphs, collaborative filtering, factorization, representation learning, deep learning, neural networks, etc.

Contributions. This survey aims to provide a thorough literature review on the approaches of exploiting side information for recommender systems. It is expected to help both academic researchers or industrial practitioners who are interested in recommender systems gain an in-depth understanding of how to improve recommendation performance with the usage of different types of side information. In summary, we make the following key contributions: (1) we conduct a systematic review for recommendation approaches with the incorporation of side information from two orthogonal perspectives. That is, different fundamental methodologies and various representations of side information; (2) we propose a novel taxonomy to classify existing recommendation approaches, which clearly demonstrates the evolution process of recent research studies; (3) we provide a comprehensive literature review of state-of-the-art studies by providing insightful comparison and analysis; and (4) we identify future directions and potential trends in this research area to shed light and promote further investigation on side information for more effective recommendations.

2 EVOLUTION OF RECOMMENDERS WITH SIDE INFORMATION

Prior to diving into state-of-the-art approaches on exploiting side information, we first introduce the relevant concepts and provide an overview of the evolution of research focusing on both recommendation methodologies and side information.

2.1 Overview of recommender systems

Generally, recommender systems predict users' preferences on items to assist users for making easier decisions. This section provides an overview of recommender systems from different perspectives. Specifically, recommender systems can be classified based on the strategies, tasks and outputs, as shown in Table 1.

Classification by strategies. Recommendation strategies can usually be classified into three categories: (1) content-based filtering, (2) collaborative filtering and (3) hybrid methods. The first two are relevant to our review as the content-based filtering methods provide us a vital clue on the various side information as well as ways to use it for recommendation, and the collaborative filtering methods give us a complete picture on the development of fundamental recommendation methodologies that are then studied to incorporate side information for better recommendations. That being said, the hybrid methods are the main focus of our investigation as they inherit and develop both content-based and collaborative filtering strategies. More detailed descriptions of the three types of strategies are presented as follows:

- **Content-based filtering.** It mainly utilizes user profiles and item descriptions to infer users' preferences towards items. The basic process is to build the profile of a user based on her personal attributes or descriptions of historical items that she has purchased or liked. The recommendations are created by matching the content of items with user profiles. In particular, a range of auxiliary data, such as categories, tags, brands, and images, can be utilized to construct descriptive features of an item. As these methods mainly rely on the rich content features of users and items, they are capable of handling the data sparsity and cold-start problems better. Meanwhile, they enable us to gain a deep understanding of how side information is exploited by state-of-the-art algorithms.
- **Collaborative filtering (CF).** This technique aims to predict users' preferences towards items by learning from user-item historical interactions, either in the form of explicit feedback (e.g., ratings and reviews) or implicit feedback (e.g., click and view). Generally, there are two types of CF-based techniques: memory- and model-based methods. The former methods (Hwang et al. 2012; Guo et al. 2012) usually exploit original user-item interaction data (e.g., rating matrices) to predict unobserved ratings by aggregating the preferences of similar users or similar items.

The latter assume that the preference of a user or the characteristic of an item can be represented by a low-dimensional latent vector. More specifically, model-based methods learn the latent feature vectors of users and items from user-item matrices, and predict the recommendations by calculating the dot product of the latent vectors of the user and item (Koren et al. 2009; Mnih and Salakhutdinov 2008). Empirical studies have proven that model-based methods outperform memory-based ones in most cases. However, the *data sparsity* and *cold start* issues inherently hinder the effectiveness of CF-based methods when user-item interaction data are very sparse. As the most successful technique in recommendation, these methods enable us to have a comprehensive understanding on the evolution of fundamental methodologies in this area.

- **Hybrid methods.** They take advantage of both CF- and content-based approaches so as to remedy their shortcomings. There are two types of techniques for blending different recommendation models: early fusion and late fusion. The former refers to combining both explicit contents (e.g., visual, textual, and knowledge-aware features) and historical user-item interaction data, and then feeding them into some CF-based methods to boost recommendation performance (Zhang et al. 2016; Tuan and Phuong 2017). On the other hand, late fusion methods build separate recommender systems that are specialized to each kind of information, and then combine the predictions of these systems (Park et al. 2006; Melville et al. 2002; Pero and Horváth 2013). Hybrid recommendation methods are known to empirically outperform the pure CF- or content-based methods, especially for solving the data sparsity and cold start problems. Our investigation mainly focuses on state-of-the-art hybrid recommendation methods. The vast majority of them were developed in the recent 10 years. In total, around 95% of the papers were published in 2010 – 2019, and more than 60% of the papers were published in the recent five years.

Classification by tasks. In terms of whether to consider time information (e.g., the order of historical interactions), recommender systems can be categorized by general, temporal and sequential recommendation tasks.

- **General recommendation**. It normally leverages global user-item interaction data to recommend the top-N items for users. The algorithms, such as matrix factorization (Koren et al. 2009) and its derived models (e.g., Singh and Gordon 2008; Chen et al. 2012; Rendle 2012), are able to effectively model user preferences, thus providing a static list of recommendations reflecting long-term interests of each user.
- **Temporal recommendation.** It usually captures user preferences given a timestamp or a time period. More specifically, some methods (e.g., TimeSVD++ (Koren 2009)) split time into several segments, and model the user-item interactions in each segment. To build an effective temporal recommender system, the key is to model the dynamics of user preferences that exhibit significant (short- or long-term) temporal drift (e.g., 'what users prefer to have for lunch' or 'which places users want to visit on weekends?') (Koren 2009; Xiong et al. 2010; Wu et al. 2017b; Hosseini et al. 2018).
- **Sequential recommendation (or next-item recommendation).** It is different from the above tasks, as sequential recommendation predicts users' *next preferences* based on their most recent activities (Rendle et al. 2010; Hidasi et al. 2015; Wang et al. 2015a; Yu et al. 2016; Jing and Smola 2017; Tang and Wang 2018; Kang and McAuley 2018; Pasricha and McAuley 2018; Zhang et al. 2018). In other words, sequential recommendation seeks to model sequential patterns among successive items, and generate well-timed recommendations for users. Therefore, it is more difficult than the other two types of recommendation tasks mentioned above.

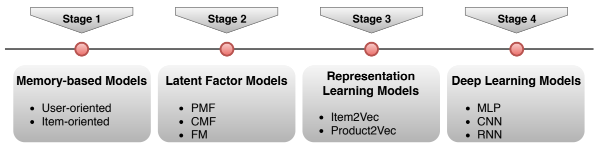
Classification by outputs. Another categorization is based on the form of outputs, and there generally are two types of tasks: rating- and ranking-based item recommendation tasks (Sun 2015). Rating-based recommendation (rating prediction) predicts users' explicit preference scores towards items, which is usually considered as a regression task. In contrast, ranking-based recommendation (item ranking) focuses on the (relative) ranking positions of items and usually generates a top-N item recommendation list to each user.

Discussion. In summary, there can be different ways to categorize recommender systems from various perspectives. Existing classification taxonomies, however, cannot help deliver a complete picture of the research studies in recommendation with side information. In this view, we create a new taxonomy to classify the literature based on two aspects: the representation of side information and the fundamental recommendation methodologies. The proposed taxonomy mainly focuses on the hybrid recommendation methods, sweeping recent state-of-the-art algorithms in various tasks (general, temporal and sequential) with different types of outputs (rating prediction and item ranking). More importantly, it allows the research community to capture a comprehensive understanding of how side information is leveraged for effective recommendations. Detailed discussions of the relevant literature will be presented in Sections 3 and 4.

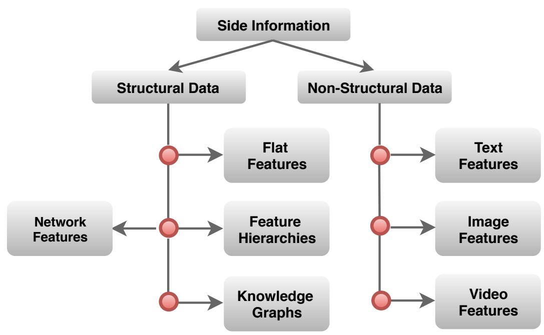
2.2 Evolution of fundamental methodologies for recommendation

In terms of the fundamental recommendation methodologies, we mainly focus on CF-based approaches as most of the advances fall into this category (Koren 2008; Rendle et al. 2009; Koren et al. 2009; Sedhain et al. 2015; He et al. 2017; Wu et al. 2017b). Before diving deep into the specific methods that employ side information, we illustrate the evolution of CF-based recommendation techniques with the progressive timeline shown in Fig. 3a. Generally, two types of CF-based approaches are widely investigated, namely memory-based and model-based (e.g., latent factor models) approaches.

Memory-based approaches. Memory-based approaches are also referred to as *neighborhood-based collaborating filtering algorithms*. They are among the earliest techniques that aggregate the interests of neighbors for recommendation. Specifically, memory-based approaches exploit user-user or item-item similarity derived from the user-item historical interaction matrix to make recommendations. User- and item-oriented methods are two kinds of typical memory-based approaches. User-oriented approaches identify like-minded users who can complement each other's ratings. The ratings of a



(a) Evolution of fundamental methodologies that are applied into recommendation with the progressive timeline marked by the red dots.



(b) Evolution of side information that are exploited for recommendation with the progressive timeline marked by the red dots.

Figure 3: Evolution of fundamental methodologies and side information that are exploited in recommender systems with the progressive timeline marked by the red dots.

target user are predicted based on the ratings of similar users found in a system. In contrast, item-oriented approaches evaluate a user's preference for an item based on the ratings of similar items rated by the same user. Although memory-based approaches have been adopted in real-world applications such as CiteULike, Youtube, and Last.fm, they are ineffective for large-scale datasets as searching for similar users or items can be time-consuming in large user or item space.

Model-based approaches. Model-based approaches aim to build predictive models by adopting data mining or machine learning techniques on user-item rating matrices to uncover complex user behavior patterns. The learned models are then used to predict users' ratings of unknown items. Besides the user-item rating matrix, side information can serve as additionally valuable features that are fed into the predictive models, and thus assist in resolving the data sparsity and cold start issues. Model-based approaches can better adapt and scale up to large-scale datasets with significant performance improvements when compared with memory-based ones. Typically, successful model-based recommendation approaches fall into three categories: latent factor models, representation learning models and deep learning models.

- **Latent factor models (LFMs)**. They decompose the high dimensional user-item rating matrix into low-dimensional user and item latent matrices. Due to high efficiency, state-of-the-art recommendation methods are dominated by LFMs (Shi et al. 2014). The basic idea of LFMs is that both users and items can be characterized by a few latent features, and thus the prediction can be computed as the inner product of user-feature and item-feature vectors. Many effective approaches fall into this category, such as *matrix factorization* (MF) (Koren et al. 2009), *non-negative matrix factorization* (NMF) (Zhang et al. 2006), *tensor factorization* (TensorF) (Bhargava et al. 2015), *factorization machine* (FM) (Rendle 2010, 2012), SVD++ (Koren 2008), *collective matrix factorization* (CMF) (Singh and Gordon 2008) and SVDFeature (Chen et al. 2012).
- **Representation learning models (RLMs)**. They have been proven to be effective in capturing local item relationships by modeling item co-occurrence in an individual user's interaction records. RLMs were originally inspired by word embedding techniques, which can be traced back to the classical neural network language model (Bengio et al. 2003), and the recent breakthroughs of Word2Vec techniques, including CBOW and Skip-gram (Mikolov et al. 2013). Many Item2Vec (Barkan and Koenigstein 2016) based recommendation approaches, which are analogous with the Word2Vec technique, have been proposed to date (Wang et al. 2015a; Grbovic et al. 2015; Liang et al. 2016; Feng et al. 2017).
- **Deep learning models (DLMs)**. They have brought significant breakthroughs in various domains, such as computer vision, speech recognition, and natural language processing (LeCun and Bengio 1995; Socher et al. 2011; Krizhevsky et al. 2012; Luong et al. 2015; Wang et al. 2016), with recommender systems being no exception. In contrast to LFMs and RLMs, DLMs (e.g., AutoRec (Sedhain et al. 2015), NCF (He et al. 2017) and DMF (Xue et al. 2017)) can learn nonlinear latent representations via various types of activation functions (e.g., sigmoid, ReLU (Nair and Hinton 2010)). For instance, recurrent neural network (RNN) based approaches (Hidasi et al. 2015; Jing and Smola 2017; Wu et al. 2017b; Hosseini et al. 2018) have shown powerful capabilities for sequential recommendation due to the ability of preserving historical information over time. Convolutional neural network (CNN) based approaches (Zhang et al. 2016; He et al. 2016b; He et al. 2016a) are capable of extracting local features so as to capture more contextual influences. In summary, DLMs possess essential advantages, and have promoted active and advanced studies in recommendation.

Discussion. In essence, both the LFMs (e.g., matrix factorization) and RLMs (e.g., item2vec) can be considered as a special case of DLMs, that is, the shallow neural networks (He et al. 2017). For instance, matrix factorization can be regarded as a one-layer neural network which transforms one-hot user and item vectors to dense representations with a linear inner product of these vectors for prediction. Although DLMs achieve superior performance against other model-based recommendation methods, the investigation into how to efficiently incorporate diverse side information into DLMs has not reached its full potential. In contrast, such research issues have been well studied for LFMs and RLMs in the recent decades, which could provide inspiration for the development of DLMs with side information. On the other hand, in comparison with DLMs, which involve more computational cost but often only achieve small performance increments, traditional model-based methods (e.g., LFMs and RLMs) have the potential to be further developed to produce better recommendation accuracy. Trading-off between the recommendation accuracy and the computational cost is, therefore, an important direction for future research that

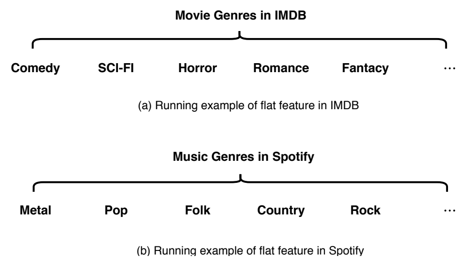


Figure 4: Examples of flat features, where all features are independently orgainzed at the same layer. Both movies in IMDB and music in Spotify are classified by genres.

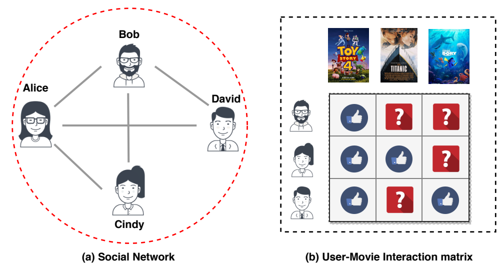


Figure 5: An example of social networks. (a) shows the social network where Bob, Cindy and David are friends of Alice; and (b) presents the user-movie interactions.

requires a comprehensive review of the different types of recommendation methodologies. To this end, we conduct a systematic and comprehensive review on state-of-the-art algorithms along with the evolution of fundamental methodologies, so as to deliver a complete picture in this field.

2.3 Evolution of side information for recommendation

In order to resolve the data sparsity and cold-start issues, recent CF-based recommendation techniques focus more on exploiting different kinds of side information such as social networks and item categories. Such side information can be used to estimate users' preferences even with insufficient user-item historical interaction data. For example, the emergence of social networks help us to indirectly infer a user's preference by aggregating her friends' preferences. Other side information (e.g., item tags or categories) can be directly used for understanding a user's interests, such as the categories of movies or music albums reflect what types of movies or music she enjoys. To achieve a systematic understanding, we propose a new taxonomy to categorize the side information by the presence of their intrinsic structures, including structural data (i.e., flat features, network features, hierarchical features and knowledge graphs) and non-structural data (i.e., text features, image features and video features). The taxonomy is shown in Fig. 3b.

Flat features (FFs). Early studies (Lippert et al. 2008; Sharma et al. 2011; Hwang et al. 2012; Yang et al. 2012; Liu et al. 2013; Ji et al. 2014; Hu et al. 2014; Vasile et al. 2016) mainly focused on integrating flat features (FFs), where the features are organized independently at the same layer. Fig. 4 illustrates an example of flat features in IMDB and Spotify to organize movies or music by genres. Assume that if a user prefers one movie/song under a certain genre, she is more likely to favor other movies/songs under this genre. Such side information has been widely leveraged for better movie or music recommendations (Koenigstein et al. 2011; Pei et al. 2017b; Sun et al. 2017a).

Network features (NFs). The advent of social networks has promoted active research on trust-aware recommender systems (Guo et al. 2012; Fang et al. 2014; Guo et al. 2015a). As a kind of homogeneous graph with single type of entity (user) and entity relation (friendship), social networks provide an alternative view of user preferences other than item ratings. The intuition is that social friends may share similar preferences and influence each other by recommending items. It has been proven that the fusion of social networks can yield significant performance enhancements (Jamali and Ester 2010; Ma et al. 2011b; Forsati et al. 2014; Guo et al. 2015b; Ding et al. 2017). Fig. 5 illustrates an example of social networks to help resolve the cold start issue of recommender systems. Alice, as a newly enrolled user, can also get movie recommendations (Toy Story 4), as all of her friends (Bob, Cindy and David) favor this movie.

Feature hierarchies (FHs). More recently, researchers have attempted to investigate user / item features with a more complicated structure, a *feature hierarchy* (FH), to further enhance recommendation performance. A FH is a natural yet powerful structure for human knowledge, and it provides a machine- and human-readable description of a set of features and their *affiliatedTo* relations. The benefits brought by explicitly modeling feature relations through FHs have been studied in a broad spectrum of disciplines, from machine learning (Jenatton et al. 2010; Kim and Xing 2010) to natural language processing (Hu et al. 2015). In the context of recommender systems, FHs have been proven to be more effective in generating high-quality recommendations than FFs (Ziegler et al. 2004; Weng et al. 2008; Menon et al. 2011; Koenigstein et al. 2011; Mnih 2011; Kanagal et al. 2012; He et al. 2016b; He et al. 2016b; Yang et al. 2016a; Sun et al. 2017b). Typical examples of FHs include online products hierarchies (e.g., the Amazon web store (McAuley et al. 2015)) and food hierarchies (e.g., Gowalla (Liu et al. 2013)). Fig. 6 offers an example of a 3-layer FH for Women's Clothing in Amazon. If a customer prefers skirts, she may possibly like heels to match her skirt instead of athletic shoes. This is due to both Skirts and Heels belonging to a higher layer category – Fashion Clothing, and they inherit similar characteristics of fashion style. By considering the *affiliatedTo* relations among features in FHs, recommendations can be generated in a more accurate and diverse manner.

Knowledge graphs (KGs). Recently, with the development of semantic web, knowledge graphs (KGs) (Yan et al. 2007; Lin et al. 2015; Wang et al. 2017b; Cai et al. 2018) as an auxiliary data source have attracted extensive interest in the community of recommender systems. In contrast with FHs, which are generally limited to describing features with the child-parent (i.e., *affiliatedTo*) relationship, KGs connect various types of features related to users (e.g., demographics and social networks) or items (e.g., the genre, director and actor of a movie), in a unified global representation space (See Fig. 7). Leveraging the heterogeneous connected information from KGs

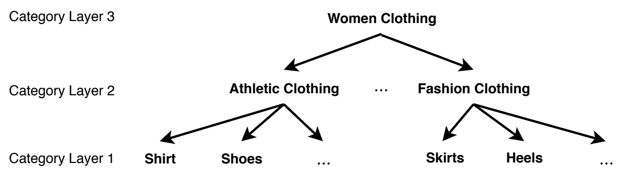


Figure 6: An example of the FH in Amazon Women's Clothing, where Women Clothing is first classified into several general categories (e.g., Athletic Clothing), and then is divided into more specific sub-categories (e.g., Shirts).

Table 2: Comparison of different data structure w.r.t. entity types and entity relations.

Data	Flat features	Network features	Feature hierarchies	Knowledge graphs
Types	1	1	>= 1	> 1
Relations	0	1	1	> 1

helps with the inference of subtler user or item relationships from different angles, which are difficult to be uncovered merely with homogeneous information (e.g., genre). The recommendation accuracy can, therefore, be further boosted with the incorporation of KGs (Yu et al. 2013a; Yu et al. 2013b; Luo et al. 2014; Shi et al. 2015; Grad-Gyenge et al. 2015; Catherine and Cohen 2016; Shi et al. 2016; Zhang et al. 2016; Zheng et al. 2017a; Wang et al. 2017d; Zhang et al. 2017a; Sun et al. 2018; Wang et al. 2018a).

Non-structural data. All the aforementioned side information, including FFs, FHs, NFs and KGs, is structural knowledge. Apart from that, some non-structural data (e.g., text, image and video content) has also been widely utilized for generating high-quality recommendations. For instance, reviews posted by users have been adopted for evaluating their experience (e.g., online shopping, POI check-in). Compared with ratings, reviews can better reflect different aspects of users' preferences (Yin et al. 2013; He et al. 2015; Gao et al. 2015; Wang et al. 2017a).

Suppose a user, Sarah, posted a review for a restaurant – "*The staff was super friendly and food was nicely cooked! will visit again*". From this we may infer that Sarah is quite satisfied with the "food" and "service" of the restaurant. Hence, reviews can serve as complementary information to explain the ratings and model users' preferences in a finer granularity (Wu et al. 2017a; Catherine and Cohen 2017; Zheng et al. 2017b; Seo et al. 2017; Tay et al. 2018; Lu et al. 2018). Moreover, image has also been taken into account for better visual recommendations (Lei et al. 2016; Liu et al. 2017; Niu et al. 2018) and general recommendations (McAuley et al. 2015; Zhou et al. 2016; Wang et al. 2017c; Alashkar et al. 2017; Yu et al. 2018), as the visual features related to items (e.g., movie poster, book covers, hotel/food/clothing photos) play an important role to attract users and further affect their decision-making process (Zhang et al. 2016; He and McAuley 2016b; He and McAuley 2016a; Chen et al. 2017; Chu and Tsai 2017).

Discussion. For the structural information, from flat features to network features and feature hierarchies, and to knowledge graphs, the structure becomes more and more complex, evolving from a homogeneous structure to a heterogeneous one, with increasing

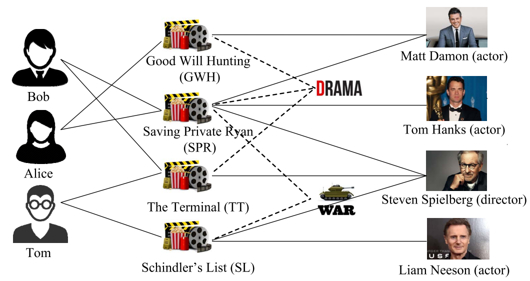


Figure 7: An example of a KG in movie domain, which contains users, movies, actors, directors and genres as entities; rating, categorizing, acting, and directing as entity relations.

entity types and entity relations, as summarized in Table 2. For instance, in the flat features, there is only one type of entity (genres of movie) and no entity relation; while in the social network, besides one entity type (users), there is only one type of entity relation (friendship). In terms of the knowledge graph, it contains multiple types of entities and entity relations in a unified space. The more sophisticated the side information is, the more knowledge and information are encoded. Therefore, it is a necessity to develop more advanced fundamental methodologies to efficiently accommodate such information. When it comes to the non-structural side information (e.g., text, images, videos), we need to utilize the deep learning advances to help extract the hidden features. In sum, it is often impossible to disentangle various useful side information from the fundamental methodologies for better recommendations: they are mutually enhanced by each other in a cooperative fashion.

To sum up, Fig. 3a and b depict the overall scheme of the proposed new taxonomies to categorize the fundamental methodologies and diverse side information for recommendation. Specifically, we propose a novel taxonomy to categorize: (1) the fundamental recommendation methodologies from memory-based methods, latent factor models and representation learning models towards deep learning models; and (2) the side information by their intrinsic data types, including structural data (flat features, network features, feature hierarchies and knowledge graphs), and non-structural data (text, images and videos). Based on this, we conducted a systematic, comprehensive, and insightful analysis on state-of-the-art hybrid recommendation approaches with side information. Table 3 summarizes the statistics of all representative algorithms that we selectd for coverage (164-28=136 in total) from the above two perspectives. Around 95% of the papers were published in recent 10 years. For ease of exposition, we will present and analyze all the conventional models (i.e., memory based methods, latent factor models and representation learning models) with various types of side information in Section 3. Following this, Section 4 introduces deep learning models with diverse side information.

3 CONVENTIONAL MODELS WITH SIDE INFORMATION

In this section, we present and analyze the exploitation of various side information for conventional recommendation models, including memory-based methods, and latent factor models, as well as representation learning models.

Table 3: Summary of representative state-of-the-art recommendation algorithms with side information, where 'FFs, NFs, FHs, KGs' denote the structural side information, namely flat features, network features, feature hierarchies, and knowledge graphs, respectively; 'MMs, LFMs, RLMs, DLMs' represent memory-based, latent factor, representation learning and deep learning models, respectively. Note that they have the same meanings for all the following tables. Besides, in this table we also include the 'Basic' methods without incorporating side information for each type of methodology.

No.	Basic	Structural Data				Non-Structural Data			Total
		FFs	NFs	FHs	KGs	Text	Images	Videos	
MMs	2	2	5	3	–	2	–	–	14
LFMs	8	17	15	10	6	10	9	–	75
RLMs	6	4	–	–	–	–	–	–	10
DLMs	12	7	5	2	14	18	6	1	65
Total	28	30	25	15	20	30	15	1	164

3.1 Memory-based methods with side information

Early recommendation approaches with side information were mainly built upon memory-based methods (MMs) (Schafer et al. 2007; Adomavicius and Tuzhilin 2005; Desrosiers and Karypis 2011). Typical research includes approaches either with item side information (e.g., item categories (Sharma et al. 2011; Hwang et al. 2012)) or user side information (e.g., social networks (Guo et al. 2012)).

MMs+FFs. Many MMs consider *flat features* (FFs) for recommendation with pre- or post-filtering manner, based on the assumption that users may have similar interests with other users who are affiliated to the same features. For instance, Hwang et al. (2012) introduced the notion of category experts, and predicted unknown ratings for the target user by aggregating the ratings of category experts instead of traditional similar users. It is equivalent to leveraging the flat categories to cluster (i.e., pre-filter) users into different groups. Davidson et al. (2010) proposed a Youtube video recommender, where flat categories are used to post-filter videos, to further ensure the diversity of the final recommended videos.

MMs+NFs. Later, the advent of social networks has promoted active research in the area of trust-aware recommender systems. A number of works leverage social networks, that is, the *network features* (NFs), for effective recommendations (Guo et al. 2012; Guo 2012; Guo 2013; Guo et al. 2014; Guo et al. 2015a). These methods posit that social friends may share similar interests. Specifically, they estimate the unknown ratings for the target user by merging the ratings of her trusted friends.

MMs+FHs. Several researchers also attempted to fuse *feature hierarchies* (FHs) into MMs by exploiting the user- and product-taxonomy distributions. For example, Ziegler et al. (2004) devised a user-based taxonomy-driven product recommendation method. In particular, they first represented each product by a taxonomy distribution vector, where elements denote the scores of the product's affiliation to the respective topics in the taxonomy. Then, the user taxonomy vector is obtained by summarizing the vectors of products that the user has interacted with. It discovers the user neighbors by calculating the similarity of the corresponding user-taxonomy vectors. Following this, Weng et al. (2008) proposed an item-based approach named HTR with the incorporation of both the user-item preference and user-taxonomic preference. Besides, category hierarchies give a precise description about functions and properties of products. They are utilized to estimate user preferences at different category levels for recommending POIs to users who visit a new city (Bao et al. 2012).

MMs+TFs. Some researchers adopted text features (e.g. reviews, comments) via either word-level text similarity or extracted sentiment. For instance, Terzi et al. (2014) proposed TextKNN to measure the similarity between users based on the similarity of text reviews instead of ratings. Pappas and Popescu-Belis (2013) developed a *sentiment-aware nearest neighbor model* (SANN) for recommendations over TED talks. It adapts the estimated ratings by making use of the sentiment scores extracted from user comments.

Discussion. Memory-based methods (Sarwar et al. 2001;Koren 2008), however, are widely recognized as being less effective than model-based ones in large-scale datasets due to the time-consuming search in the user or item space. In a nutshell, the weak scalability of MMs limits their exploitation of the knowledge encoded in various side information, and even hinders them to encode side information with more complicated structural data (e.g., knowledge graphs) and non-structural data (e.g., images and videos). On the other hand, the underlying principles of fusing side information still provide valuable guidance for model-based methods.

3.2 Latent factor models with side information

Due to the high efficiency, state-of-the-art recommendation methods are mainly dominated by latent factor models (LFMs) (Shi et al. 2014), including *matrix factorization* (MF) (Mnih and Salakhutdinov 2008; Koren et al. 2009; Wang et al. 2015b), *weighted non-negative matrix factorization* (WNMF) (Zhang et al. 2006), *Bayesian personalized ranking* (BPR) (Rendle et al. 2009), *tensor factorization* (TensorF) (Karatzoglou et al. 2010), *factorization machine* (FM) (Rendle 2010, 2012), SVD++ (Koren 2008), timeSVD++ (Koren 2009) and RFSS (Zhao et al. 2017). As discussed, they typically learn and model users' behavior (e.g., ratings, purchases) patterns by employing the global statistical information of historical user-item interaction data. Specifically, they usually decompose the high-dimensional user-item rating matrices into low-rank user and item latent matrices. The basic idea is that both users and items can be characterized by a number of latent features, and thus the prediction can be computed as the inner product of corresponding user and item latent vectors. Many effective recommendation methods with side information fall into this category (Shi et al. 2011; Yang et al. 2012; Chen et al. 2012; Hu et al. 2014; Sun et al. 2017a).

LFMs+FFs. Early LFMs (See Table 4) incorporate *flat features* (FFs) to help learn better user and item latent representations[2]. As further summarized in Table 5, several generic feature-based methods have been proposed. For instance, Singh and Gordon (2008) proposed *collective matrix factorization* (*CMF*) by simultaneously decomposing the user-item and user-feature/item-feature matrices. Then,

[2]In this survey, the following words 'embedding', 'representation', 'latent vector' and 'latent feature' are interexchangablely used.

Table 4: Summary of state-of-the-art latent factor model based recommendation algorithms with side information, where 'FFs, NFs, FHs, KGs' represent structural side information, namely flat features, network features, feature hierarchies and knowledge graphs; 'TFs, IFs' denote the non-structural side information, namely text features and image features.

Algorithm	Venue	Year	Structural				Non-Str.		Reference
			FFs	NFs	FHs	KGs	TFs	IFs	
CMF	KDD	2008	✓	–	–	–	–	–	Singh and Gordon
TensorF	RecSys	2010	✓	–	–	–	–	–	Karatzoglou et al.
HOSVD	TKDE	2010	✓	–	–	–	–	–	Symeonidis et al.
FPMC	WWW	2010	✓	–	–	–	–	–	Rendle et al.
TagCDCF	UMAP	2011	✓	–	–	–	–	–	Shi et al.
CircleCon	KDD	2012	✓	✓	–	–	–	–	Yang et al.
FM	TIST	2012	✓	–	–	–	–	–	Rendle, Rendle
SVDFeature	JMLR	2012	✓	–	–	–	–	–	Chen et al.
NCRP-MF	SIGIR	2014	✓	–	–	–	✓	–	Hu et al.
GeoMF	KDD	2014	✓	–	–	–	–	–	Lian et al.
CAPRF	AAAI	2015	✓	–	–	–	✓		Gao et al.
ARMF	KDD	2016	✓	✓	–	–	–	–	Li et al.
ICLF	UMAP	2017	✓	–	–	–	–	–	Sun et al.
TransFM	RecSys	2018	✓	–	–	–	–	–	Pasricha and McAuley
TRec	ECRA	2019	✓	–	–	–	✓	–	Bruno et al.
SoRec	CIKM	2008	–	✓	–	–	–	–	Ma et al.
RSTE	SIGIR	2009	–	✓	–	–	–	–	Ma et al.
RWT	RecSys	2009	–	✓	–	–	–	–	Ma et al.
SocialMF	RecSys	2010	–	✓	–	–	–	–	Jamali and Ester
SoReg	WSDM	2011	–	✓	–	–	–	–	Ma et al.
RSTE	TIST	2011	–	✓	–	–	–	–	Ma et al.
TrustMF	IJCAI	2013	–	✓	–	–	–	–	Yang et al.
SR	SIGIR	2013	–	✓	–	–	–	–	Ma
DTrust	AAAI	2014	–	✓	–	–	–	–	Fang et al.
MFTD	TOIS	2014	–	✓	–	–	–	–	Forsati et al.
TrustSVD	AAAI	2015	–	✓	–	–	–	–	Guo et al.
MF-Tax	RecSys	2011	–	–	✓	–	–	–	Koenigstein et al.
TaxLF	JMLR	2011	–	–	✓	–	–	–	Mnih
H+LR++	KDD	2011	–	–	✓	–	–	–	Menon et al.
BMF	NIPS	2012	–	–	✓	–	–	–	Mnih and Teh
Tran-Cate	CIKM	2013	–	–	✓	–	–	–	Liu et al.
TaxF	VLDB	2013	–	–	✓	–	–	–	Kanagal et al.
ReMF	RecSys	2016	–	–	✓	–	–	–	Yang et al.
CHMF	UMAP	2016	–	–	✓	–	–	–	Sun et al.
Sherlock	IJCAI	2016	–	–	✓	–	–	✓	He et al.
HieVH	AAAI	2017	–	–	✓	–	–	–	Sun et al.
HeteMF	IJCAI	2013	–	–	–	✓	–	–	Yu et al.
HeteRec	RecSys	2013	–	–	–	✓	–	–	Yu et al.
HeteRec_p	WSDM	2014	–	–	–	✓	–	–	Yu et al.
HeteCF	ICDM	2014	–	✓	–	✓	–	–	Luo et al.
SemRec	CIKM	2015	–	✓	–	✓	–	–	Shi et al.
GraphLF	RecSys	2016	–	–	–	✓	–	–	Catherine and Cohen
HFT	RecSys	2013	–	–	–	–	✓	–	McAuley and Leskovec
O_Rec	UMAP	2013	–	–	–	–	✓	–	Pero and Horváth
EFM	SIGIR	2014	–	–	–	–	✓	–	Zhang et al.
TopicMF	AAAI	2014	–	–	–	–	✓	–	Bao et al.
EnFM	WWW	2017	–	–	–	–	✓	✓	Chu and Tsai
EBR	ECRA	2017	–	–	–	–	✓	–	Pourgholamali et al.
AFV	ECRA	2018	–	–	–	–	✓	–	Xu et al.
IRec	SIGIR	2015	–	–	–	–	–	✓	McAuley et al.
Vista	RecSys	2016	–	–	–	–	–	✓	He et al.
VBPR	AAAI	2016	–	–	–	–	–	✓	He and McAuley
TVBPR	WWW	2016	–	–	–	–	–	✓	He and McAuley
VPOI	WWW	2017	–	–	–	–	–	✓	Wang et al.
DeepStyle	SIGIR	2017	✓	–	–	–	–	✓	Liu et al.
DCFA	WWW	2018	–	–	–	–	–	✓	Yu et al.

Chen et al. (2012) designed *SVDFeature*, which assumes that the representations of users or items can be influenced by those of their affiliated features. Karatzoglou et al. (2010) proposed *tensor factorization* (*TensorF*), which is a generalization of MF that allows for a flexible and generic integration of features by modeling the data as a *user-item-feature* N-dimensional tensor instead of the traditional 2D user-item matrix. Rendle (2010, 2012) devised *factorization machine* (*FM*) algorithm to model the pairwise interactions between all variables using factorized parameters.

Most of the state-of-the-art LFMs+FFs methods are built upon the four types of generic feature-based methods mentioned above: (1) based on CMF, Shi et al. (2014) introduced TagCDCF by factorizing the user-item and cross-domain tag-based user and item similarity matrices. Lippert et al. (2008) proposed a prediction model – MRMF by jointly factorizing the user-item and user-feature (e.g., gender) as well as item-feature (e.g., genre) matrices. Gao et al. (2015) proposed a location recommender – CAPRF, which jointly decomposes the user-location interaction and location-tag affinity matrices; (2) based on SVDFeature, Hu et al. (2014) proposed a rating prediction approach called NCRP-MF, which learns the embeddings of items by adding their affiliated categories. Bruno et al. (2019) proposed a hotel recommender – TRec – with the incorporation of hotel themes. They argue that the embedding of a hotel should be reflected by those themes that the hotel belongs to; (3) based on TensorF, Symeonidis et al. (2010) proposed a unified recommendation model (HOSVD) via tensor factorization for user-tag-item triplet data; and (4) based on FM, Pasricha and McAuley (2018) proposed a sequential recommendation model – TransFM, which adopts FM to fuse user and item flat features, such as user gender and item category.

In addition to the aforementioned ones, there are still other related works. For instance, Yang et al. (2012) leveraged FFs to do pre-filtering. They designed CircleCon to infer the category-specific social trust circle for recommendation by assuming that a user may trust different subsets of friends regarding different categories. Given the assumption that users (items) have different preferences (characteristics) on different categories, Sun et al. (2017a) proposed a category-aware model – ICLF, which estimates a user's preference to an item by multiplying the inner product of the user and category latent vectors, and that of item and category latent vectors, where the category is the one that the item belongs to. Similarly, Li et al. (2016) proposed ARMF to predict a user's taste over an item by multiplying the inner product of user and item latent vectors, and the user's preference to the affiliated categories of the item.

Table 5: Classifications of state-of-the-arts w.r.t. LFMs+FFs.

Type	Representative Method
CMF	(1) CMF (Singh and Gordon 2008); (2) MRMF (Lippert et al. 2008); (3) TagCDCF (Shi et al. 2011); (4) CAPRF (Gao et al. 2015)
SVDFeature	(1) SVDFeature (Chen et al. 2012); (2) NCRP-MF (Hu et al. 2014); (3) TRec (Bruno et al. 2019)
TensorF	(1) TensorF (Karatzoglou et al. 2010); (2) HOSVD (Krizhevsky et al. 2012)
FM	(1) FM (Rendle 2010, 2012); (2) TransFM (Pasricha and McAuley 2018)
Others	(1) CircleCon (Yang et al. 2012); (2) ICLF (Sun et al. 2017a); (3) ARMF (Li et al. 2016)

Summary of LFMs+FFs. Table 5 summarizes all the methods that belong to LFMs+FFs category. First, significant improvements have been achieved with these methods in comparison with the plain LFMs without considering FFs, which strongly verifies the usefulness of FFs for more effective recommendations. Second, comparable performance can be obtained by CMF, SVDFeature and TensorF based methods, while the time complexity of TensorF based methods far exceeds the other two types of methods. Third, extensive empirical studies have demonstrated the superiority of the FM based approaches among all the counterparts, as they explicitly consider the pair-wise interactions between users and items as well as their flat features.

Table 6: Classifications of state-of-the-arts w.r.t. LFMs+NFs.

Type	Representative Method
CMF	(1) SoRec (Ma et al. 2008); (2) DTrust (Fang et al. 2014); (3) TrustMF (Yang et al. 2013a); (4) TrustSVD (Guo et al. 2015b)
Regularization	(1) RSTE (Ma et al.2009a, 2011a)
SVDFeature	(1) SocialMF (Jamali and Ester 2010); (2) SoReg (Ma et al. 2011b); (3) CircleCon (Yang et al. 2012); (4) SR (Ma 2013) (5) MFTD (Forsati et al. 2014); (6) RWT/RWD (Ma et al. 2009b)

LFMs+NFs. Many studies integrated social networks into LFMs for achieving better recommendation performance. The underlying rationale is that users could share similar interests with their trusted friends. Three types of representative methods, including CMF-based, SVDFeature-based, and regularization-based ones, are discussed in detail as follows.

(1) CMF based methods. One line of research is mainly based on *collective matrix factorization* (CMF) (Singh and Gordon 2008), which jointly decomposes both the user-item interaction matrix and the user-user trust matrix. For example, Ma et al. (2008) proposed SoRec to better learn the user embeddings by simultaneously factorizing the user-item and user-trust matrices. Fang et al. (2014) proposed DTrust, which decomposes trust into several aspects (e.g., benevolence, integrity) and further employs the support vector regression technique to incorporate them into the matrix factorization model for rating prediction. Yang et al. (2013a) presented TrustMF, which leverages truster and trustee models to properly catch on a twofold influence of trust propagation on the user-item interactions. Guo et al. (2015b) devised TrustSVD, which inherently involves the explicit and implicit influence of rated items, and thus further incorporates both the explicit and implicit influence of trusted users.

(2) SVDFeature based methods. Another line of research mainly follows the idea of SVDFeature, which supposes that the representation of a user will be affected by that of her trusted friends. For example, Ma et al. (2009a, 2011a) proposed RSTE, which represents the embedding of a user by adding those of her trusted friends.

(3) Regularization based methods. The third line of research adopted the regularization technique (Smola and Kondor 2003) to constrain the distance of embeddings between a user and her trusted friends. For example, SocialMF (Jamali and Ester 2010) and CircleCon (Yang et al. 2012) are designed on the assumption that a user and her trusted friends should be close to each other in their embedding space. Ma et al. (2013, 2011b) proposed SoReg and SR to minimize the embedding difference of a user and their trusted friends. Later, Forsati et al. (2014) and Ma et al. (2009b) respectively introduced MFTD and RWT/RWD to further employ distrust information to maximize the distance of embeddings between a user and her distrusted users.

Summary of LFMs+NFs. Table 6 summarizes the three types of LFMs+NFs. To conclude, first, the effectiveness of NFs for more accurate recommendation has been empirically validated, when comparing with the plain LFMs. Second, *regularization* is generally a quite straightforward and time-efficient way to incorporate social influence, which naturally allows trust propagation among indirect social friends. For instance, suppose that users u_j, u_k are friends of user u_i. By regularizing the distances of (u_i, u_j) and (u_i, u_k) respectively, the distance of (u_j, u_k) is indirectly constrained. Third, CMF based methods usually achieve the best performance. For instance, DTrust and TrustSVD outperform most of the other trust-aware approaches. Finally, the methods fusing both trust and distrust information perform better than those merely considering single aspect, suggesting the usefulness of distrust for recommendation; and this is further confirmed by the fact that the distrust-based methods perform almost as well as the trust-based methods (Ma et al. 2009b), which proves that the distrust information among users is as important as the trust information (Fang et al. 2015).

Table 7: Classifications of state-of-the-arts w.r.t. LFMs+FHs.

Type	Representative Method
SVDFeature	(1) MF-Tax (Koenigstein et al. 2011); (2) TaxF (Kanagal et al. 2012); (3) Sherlock (He et al. 2016b); (4) TaxLF (Mnih 2011); (5) CHLF (Sun et al. 2017a); (6) HieVH (Sun et al. 2017b)
Regularization	(1) H+LR++ (Menon et al. 2011); (2) ReMF (Yang et al. 2016a)

LFMs+FHs. As summarized in Table 7, the first type of algorithms are based on the basic idea of SVDFeature. For instance, both MF-Tax (Koenigstein et al. 2011) and TaxF (Kanagal et al. 2012) model the embedding of an item by equally adding those of its ancestor features in the hierarchy. Later, He et al. (2016b) proposed Sherlock, which manually defines the various influence of categories at different layers of the hierarchy. In contrast, TaxLF (Mnih 2011), CHLF (Sun et al. 2017a) and HieVH (Sun et al. 2017b) strive to automatically learn the different influences. The second type utilizes the regularization technique. For instance, Menon et al. (2011) proposed an ad-click prediction method that regularizes the embeddings of features in the hierarchy via the child-parent relation. However, it assumes that an ad is conditionally independent from all higher layer features. Yang et al. (2016a) proposed ReMF to automatically learn the impacts of category hierarchies by parameterizing regularization traversing from the root to leaf categories.

Summary of LFMs+FHs. All representative LFMs+FHs methods are summarized in Table 7. Compared with FFs where features are independently organized at the same layer, the FHs provide human- and machine-readable descriptions of a set of features, and their parent-child relations. The richer knowledge encoded in FHs enables a more accurate and diverse recommendation. Regardless of SVDFeature or regularization based methods, they all indicate that the categories at different layers of the hierarchy play different roles in characterizing the user-item interactions. The type of methods being able to automatically identify the different saliency of the hierarchical categories can achieve a better exploitation of FHs, so as to generate much more high-quality recommendations.

LFMs+KGs. Most of the LFMs+KGs methods generally first extract meta paths (Sun et al. 2011) from KGs, and these paths are then fed into LFMs for high-quality recommendations. Some of these methods adopt the regularization technique to incorporate the influence of the extracted meta paths. For instance, Yu et al. (2013a) extracted paths connecting item pairs, and leveraged the path-based item similarity as the regularization coefficient of the pairwise item

Table 8: Classifications of state-of-the-arts w.r.t. LFMs+KGs.

Meta-path-based method		Graph Method
Regularization	**Diffusion**	
• HeteMF [224] • HeteCF [112]	• HeteRec [226] • HeteRec_p [225] • HeteCF [112] • SemRec [158]	• GraphLF [20]

embeddings. Another type of methods employs the path-based similarity to learn the user preference diffusion. For example, Yu et al. (2013b) developed HeteRec to learn the user preference diffusion to the unrated items that are connected with her rated items via meta paths. It was further extended to HeteRec_p for incorporating personalization via clustering users based on their interests. Similarly, Luo et al. (2014) proposed HeteCF, which leverages the path-based similarity to model user preference diffusion to unrated items. In addition, it also adds pairwise user (item) regularization to constrain the distance of embeddings of users (items) that are connected by meta paths. Shi et al. (2015) devised SemRec that predicts the rating of a user to an item via a weighted combination of those of her similar users under different meta paths.

Besides the meta-path-based approaches, there is another line of research focusing on designing graph-based methods mainly attributed to the underlying technique of random walk. For instance, by combining the strengths of LFMs with graphs, Catherine and Cohen (2016) proposed GraphLF which adopts a general-purpose probabilistic logic system (ProPPR) for recommendation.

Summary of LFMs+KGs. Table 8 summarizes all the representative recommendation methods under LFMs+KGs. For a more in-depth discussion, first, by simplifying entity types and relation types, the complex KGs can be downgraded to other simple structural side information, such as FFs and NFs. For instance, we can only keep the item-category affinity relations, or user-user friendship relations in KGs to mimic FFs and NFs, respectively. From this point of view, LFMs+KGs can be regarded as the generalized version of feature-based approaches. Second, the majority of these methods make use of meta paths (Sun et al. 2011) to extract knowledge from the KG. By incorporating meta paths, the ideas of other recommendation models such as user-/item-oriented CF can be easily modeled in a generic way. Consider an example where we start with user u_i and follow a meta path:

$$\text{User} \xrightarrow{\text{isFriendOf}} \text{User} \xrightarrow{\text{watched}} \text{Movie}.$$

We thus can reach the movies that are watched by the friends of u_i. Hence, this meta path underpins the idea of user-oriented CF. To sum up, the usage of meta paths helps deliver an ensemble recommender. Third, the success of these methods, nevertheless, heavily relies on the quality and quantity of the handcrafted meta paths, which additionally requires domain knowledge. Besides, the manually designed features are often incomplete to cover all possible entity relations. These issues largely limit the capability of these methods to generate high-quality recommendations.

LFMs+TFs. Aside from the user-item rating matrices, the relevant reviews often provide the rationale for users' ratings and identify what aspects of an item they cared most about, and what sentiment they held for the item. We summarize four types of methods under LFMs+TFs: word-level, sentiment-level, aspect-level, and topic-level methods. They mainly focus on extracting useful information encoded in the text features, such as reviews, tips, comments, content and descriptions, to further boost recommendation accuracy.

(1) Word-level methods. Word-level methods usually directly extract the words from textual information. For instance, Hu et al. (2014) proposed a SVDFeature based method (NCRP-MF) that models the embedding of a business by adding those of words extracted from its relevant reviews. Pourgholamali et al. (2017) proposed a feature-based matrix factorization method (EBR) which first extracts words from product descriptions and user review texts, and then employs the word embedding technique (Mikolov et al. 2013) to learn semantic product and user representations. These representations are ultimately incorporated into the matrix factorization model for better recommendations. Chu and Tsai (2017) proposed EnFM, which extracts important words from textual reviews via *term frequency and inverse document frequency* (TF-IDF) technique (Ramos et al. 2003). They enhanced the *factorization machine* (FM) by fusing the extracted words as features of users and items.

(2) Sentiment-level methods. The second type is the sentiment-level, that is, analyzing the sentiment expressed by the textual information. Some studies leverage the extracted sentiment to do pre- or post-filtering. For instance, Pero and Horváth (2013) proposed O_pre to pre-process the user-item interaction matrix to generate the user-item opinion matrix, where the opinion matrix is obtained based on textual reviews. They also devise O_post to post-process the predicted ratings by adding the estimated opinion score. Bruno et al. (2019) proposed TRec to utilize binary sentiment score extracted from reviews to post-filter low-quality items from the final item ranking list. Other studies leverage the extracted sentiment from reviews as the corresponding confidence for the factorization, which indicates the importance of each user-item interaction pair. O_model (Pero and Horváth 2013) and CAPRF (Gao et al. 2015) both adopted the user-item sentiment matrix as a confidence matrix to constrain the factorization process. Recently, Xu et al. (2018) proposed AFV based on SVD (Paterek 2007), which employs the adjective features extracted from user reviews to reflect users' perceptions on items. It automatically learns user and item representations under these features for more accurate and explainable item recommendations.

(3) Aspect-level methods. To minimize the reliance on sentiment analysis accuracy, the third type is based on the aspect-level, which extracts aspects (the specific properties of items) from textual information. For instance, He et al. (2015) proposed Trirank that accommodates users, items and aspects into a heterogeneous graph. They adopted graph regularization technique (Smola and Kondor 2003) to constrain the distance of user-item, user-aspect and item-aspect pairs. Guo et al. (2017) developed a knowledge graph named as *aspect-aware geo-social influence graph*, which incorporates the geographical, social and aspect information into a unified graph. Zhang et al. (2014a) devised EFM by extracting both aspect and sentiment from the user reviews. It builds user-aspect attention and item-aspect quality matrices based on the phrase-level sentiment analysis, and then simultaneously decomposes these two matrices together with the user-item interaction matrix.

Table 9: Classifications of state-of-the-arts w.r.t. LFMs+TFs.

Type	Representative Method
Word	(1) NCRP-MF (Hu et al. 2014); (2) EBR (Pourgholamali et al. 2017) (3) EnFM (Chu and Tsai 2017)
Sentiment	(1) O_pre, O_post, O_model (Pero and Horváth 2013); (2) CAPRF (Gao et al. 2015); (3) AFV (Xu et al. 2018); (4) TRec (Bruno et al. 2019); (5) ORec (Zhang and Chow 2015)
Aspect	(1) Trirank (He et al. 2015); (2) EFM (Zhang et al. 2014a)
Topic	(1) HFT (McAuley and Leskovec 2013); (2) TopicMF (Bao et al. 2014) (3) AFV (Xu et al. 2018)

(4) Topic-level methods. This type of methods exploits the topic modeling methods, such as *latent Dirichlet allocation* (LDA) (Blei et al. 2003), to extract the latent topics in the review texts. For instance, McAuley and Leskovec (2013) proposed the *hidden factors as topics* (HFT) approach, which learns the item latent-topic distribution from all related reviews. The learned distribution is then linked with the corresponding item latent factor via a transformation function. Later, Bao et al. (2014) further extended HFT by proposing TopicMF. It correlates the latent topics of each review with the user and item latent factors simultaneously. Also, AFV (Xu et al. 2018) adopts LDA to learn the item-topic distribution. Then the Kullback-Leibler (KL) divergence (Hershey and Olsen 2007) is utilized to calculate the review-topic-based neighbors of items.

Summary of LFMs+TFs. We summarize these state-of-the-art methods in Table 9, where word-level approaches are the most straightforward ones without any understanding about the content of text features, while the aspect- and sentiment-level approaches leverage *natural language processing* (NLP) toolkits to extract useful information from the text features. The NLP technique helps the algorithms explicitly understand what aspects of an item that users care most about, and what opinions they possess for the item. The accuracy of aspect extraction and sentiment analysis, nonetheless, is the bottleneck for further advancements. Moreover, the topic-level approaches go deeper to extract the latent topics hidden in the text features. The learned latent topic distribution enables the algorithms to achieve a subtle understanding of the user-item interactions. In a nutshell, from word-level to the sentiment- and aspect-level, and ultimately to the topic-level, an increasingly deeper and more subtle understanding on the text features is gradually achieved, which enables the algorithms to model text features from a raw- to fine-grained manner. All methods mentioned above, however, ignore the fact that not all reviews written by a user (or written for an item) are equally important for modeling user preferences (or item characteristics), and even not all the words contained in one review contribute identically to represent this review. These issues have been well addressed by the deep learning models with attention mechanisms, which we will introduce later.

LFMs+IFs. There are also several latent factor models that consider image features, since images play a vital role in domains like fashion, where the visual appearances of products have great influence on user decisions. As a type of extremely complicated non-structural data, the fusion of image features for LFMs, however, generally follows two phases: (1) extract visual features from images based on the pre-trained models, such as deep neural networks; and (2) fuse the extracted visual features into LFMs for better recommendations.

He et al. (2016a, 2016b, 2016a, 2016b) proposed a series of recommendation methods for the fashion domain to exploit visual features that are extracted from product images by pre-trained deep neural networks (Jia et al. 2014). They include: (1) VBPR (He and McAuley 2016b) is an extension of BPR-MF (Rendle et al. 2009) that learns an embedding kernel to linearly transform the high-dimensional product raw visual features into a much lower-dimensional 'visual rating' space. The low-dimensional product visual features are then fused into BPR-MF for more accurate recommendations; (2) Sherlock (He et al. 2016b) upgrades VBPR, and learns additional product visual vectors by mapping the raw product visual features across hierarchical categories of products. It, thus, accounts for both high-level and subtle product visual characteristics simultaneously; (3) TVBPR (He and McAuley 2016a) advances VBPR by studying the evolving visual factors that customers consider when evaluating products, so as to make better recommendations; and (4) Vista (He et al. 2016a) further takes into account visual, temporal and social influences simultaneously for a sequential recommendation in fashion domain.

Other researchers also endeavored to make use of visual features for more accurate recommendation. For instance, McAuley et al. (2015) proposed an image-based recommender, IRec, which employs the visual features to distinguish alternative and complementary products. Chu and Tsai (2017) devised a restaurant recommder, EnFM, by leveraging two types of visual features, namely *convolutional neural network* (CNN) (LeCun and Bengio 1995) features and *color name* (CN). It first represents the attribute of a restaurant by the visual features of its related images, and then feeds the attribute into the factorization model to help achieve high-quality recommendations. Wang et al. (2017c) designed VPOI to incorporate the visual content of restaurants for an enhanced *point-of-interest* (POI) recommendation. For a user-location pair (u, l), it minimizes the difference between user u (location l) and her posted images (its relevant images) with regard to their embeddings. Liu et al. (2017) proposed DeepStyle for learning style features of items and sensing preferences of users. It learns item style representations by subtracting the corresponding item category representations from the visual features generated via CNN. The learned item style representations are then fused into BPR for personalized recommendations. Similarly, Yu et al. (2018) proposed a tensor factorization model – DCFA to leverage the aesthetic features rather than the conventional features to represent an image. They believed that a user' s decision largely depends on whether the product is in line with her aesthetics.

Summary of LFMs+IFs. Undoubtedly, the image features have inspired new models for recommender systems, that is, exploiting visual features (e.g., style of an item) to model the user-item interactions, and have greatly boosted recommendation accuracy and attractiveness. As we mentioned earlier, due to the intricacy of image features, they can not be directly used by LFMs. To this end, two modules are required to be separately trained: (1) feature extraction module usually adopts deep learning techniques, such as CNN (LeCun and Bengio 1995), to learn visual representations of items from images; and (2) the preference learning module, such as matrix factorization, takes the learned visual representations to help adjust the final item representations learning process. Such separately learning hinders these methods from achieving optimal performance improvements. In this view, it calls for unified and elegant recommendation models to exploit such kinds of features.

Discussion of LFMs with side information. First, compared with memory-based methods, LFMs have relatively higher scalability and flexibility, which enables them to incorporate various types of side information, regardless of the structural or non-structural ones, for more accurate recommendations. However, the investigation of fusing structural data is more prevalent than that of non-structural data, as illustrated by Table 4; Second, most of the LFMs with side information are extended from the generic feature-based methods, such as matrix factorization with side information regularization (Jamali and Ester 2010), *collective matrix factorization* (CMF) (Singh and Gordon 2008), SVDFeature (Chen et al. 2012), *tensor factorization* (TensorF) (Karatzoglou et al. 2010), and *factorization machine* (FM) (Rendle 2010, 2012); Third, generally, the more complex side information has been incorporated, the more high-quality recommendations can be achieved. For instance, LFMs+FHs outperforms LFMs+FFs, and the performance of LFMs+KGs is better than that of other LFMs with structural side information; Fourth, although LFMs are capable of accommodating more complex side information (i.e., knowledge graphs, text and image features), the information encoded in such complex data cannot be directly utilized by LFMs. In this sense, most of these methods are composed of two independent phases, namely *feature extraction* and *preference learning*. For instance, the fusion of knowledge graphs is based on meta-path (Sun et al. 2011) extraction. The integration of text features relies on aspect extraction and sentiment analysis advances (Zhang et al. 2014b), and the utilization of image features highly depends on deep neural networks (Jia et al. 2014; LeCun and Bengio 1995) to help extract visual features of images. The independence of these two phases limits further performance increments of LFMs with side information to some degree. This issue has been partially alleviated by deep learning models, as elaborated in the subsequent subsections.

3.3 Representation learning models with side information

In contrast to LFMs, *representation learning models* (RLMs) have proven to be effective for recommendation tasks in terms of capturing local item relations by utilizing item embedding techniques.

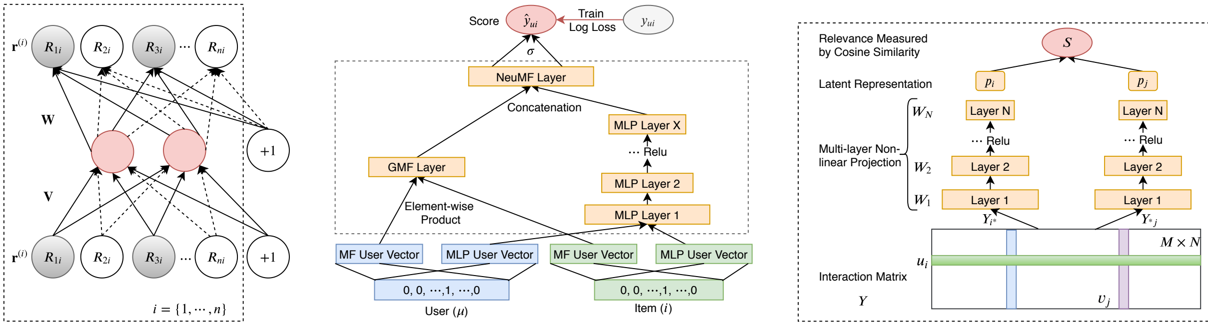
There are many studies related to RLMs for recommendation. For instance, Barkan and Koenigstein (2016) first devised a neural item embedding model (Item2Vec) for collaborative filtering, which is capable of inferring item-to-item relationships. Note that the original Item2Vec cannot model user preferences by only learning item representations. Some studies, therefore, extended Item2Vec by taking personalization into account. For instance, Grbovic et al. (2015) developed three recommendation models: Prod2Vec, BagProd2Vec and User2Vec. Specifically, Prod2Vec learns product representations at product-level over the entire set of receipt logs, whereas BagProd2Vec learns at the receipt-level. User2Vec simultaneously learns representations of products and users by considering the user as a "global context", motivated by Paragraph2Vec algorithm (Le and Mikolov 2014). Next, Wang et al. (2015a) proposed a novel *hierarchical representation model* (HRM) to predict what users will buy in the next basket (sequential recommendation). HRM can capture both sequential behavior and users' general tastes. Liang et al. (2016) proposed CoFactor based upon CMF (Singh and Gordon 2008), which synchronously decomposes the user-item interaction matrix and the item-item co-occurrence matrix.

Table 10: Classifications of state-of-the-arts w.r.t. representation learning models (RLMs), where 'Basic' denotes the fundamental RLMs without side information; and 'Side' represents the RLMs with side information.

Type	Non-Personalized	Personalized
Basic	Item2Vec (Barkan and Koenigstein 2016) Prod2Vec (Grbovic et al. 2015) BagProd2Vec (Grbovic et al. 2015)	User2Vec (Grbovic et al. 2015) HRM (Wang et al. 2015a) CoFactor (Liang et al. 2016)
Side	MetaProd2Vec (Vasile et al. 2016)	CWAPR-T (Liu et al. 2016a) POI2Vec (Feng et al. 2017) MRLR (Sun et al. 2017c)

By taking advantages of RLMs, some researchers attempted to integrate side information (e.g., categories, tags) into RLMs to help learn better user and item embeddings, thus to gain further performance enhancements for recommendation (Grbovic et al. 2015; Vasile et al. 2016; Sun et al. 2017c). For instance, Vasile et al. (2016) extended Item2Vec to a more generic non-personalized model – MetaProd2Vec, which utilizes item categories to assist in regularizing the learning of item embeddings. Liu et al. (2016a) proposed temporal-aware model (CWARP-T) by leveraging the Skip-gram model. It jointly learns the latent representations for a user and a location, so as to respectively capture the user's preference as well as the influence of the context of the location. Feng et al. (2017) designed POI2Vec which incorporates the geographical influence to jointly model user preferences and POI sequential transitions. Recently, Sun et al. (2017c) proposed a personalized recommender, MRLR, to jointly learn the user and item representations, where the item representation is regularized by item categories.

Discussion of RLMs with side information. Table 10 summarizes all the RLMs based recommendation approaches. First, although there is much less work on RLMs with side information than that on LFMs with side information, RLMs provide a different viewpoint to learn item representations: by capturing local item relations in terms of each individual user's interaction data while LFMs aim to learn user and item representations at the global level. Second, the objective function of fundamental RLMs (Item2Vec) is actually a softmax layer which has been widely adopted in the attention mechanisms (Chen et al. 2017; Seo et al. 2017), or as the output layer of many deep learning models (Zhang et al. 2017b; Yang et al. 2017). From this viewpoint, RLMs can be considered as transitions from shallow to deep neural networks. Third, the fundamental RLMs (Item2Vec) do not consider personalization very well, and should be further extended to accommodate the user's preference. This can be expressed in different manners, such as the averaged representations of items that the user has interacted with, or be treated as the "global context" via Paragraph2Vec (Le and Mikolov 2014). Fourth, most of the RLMs with side information focus on incorporating simple flat features, such as item categories (Sun et al. 2017c; Vasile et al. 2016). This is equivalent to adding regularizations on the item embedding learning process. Similarly, other



(a) Auto-Encoder Model (b) NeuMF Model (c) DMF Model

Figure 8: (a) Item-based AutoRec model, where we use the plate notation to indicate that there are n copies of the neural network (one for each item), where W, V are tied across all copies; $\mathbf{r}^{(i)}$ denotes the observed rating vector for item i (Sedhain et al. 2015); (b) Neural matrix factorization model, which fuses generalized matrix factorization (GMF) on the left side and multi-layer perceptron (MLP) on the right side (He et al. 2017); (c) Deep matrix factorization model, which leverages multi-layer non-linear projections to learn user and item representations (Xue et al. 2017).

types of data (e.g., FHs) could be considered and adapted to further augment the performance of RLMs based recommendation methods. Lastly, the exploitation of Item2Vec stemming from Word2Vec inspires more technique transformations from the NLP domain to recommendation tasks, such as Paragraph2Vec and Document2Vec (Le and Mikolov 2014).

4 DEEP LEARNING MODELS WITH SIDE INFORMATION

Deep learning models (DLMs) have gained significant success in various domains, such as computer vision (CV) (Krizhevsky et al. 2012), speech recognition (Schmidhuber 2015), and natural language processing (NLP) (Cho et al. 2014b). They have also recently attracted tremendous research interest from the recommendation community. In contrast to LFMs and RLMs, DLMs based recommendation approaches (e.g., AutoRec (Sedhain et al. 2015), NCF (He et al. 2017)) can learn nonlinear latent representations through various types of activation functions, such as sigmoid, ReLU (Nair and Hinton 2010). Thanks to the excellent flexibility of DLMs, side information can be efficiently integrated. Plenty of DLMs employ different kinds of structural side information to help achieve better recommendations, such as item categories (Pei et al. 2017b), social networks (Ding et al. 2017) and knowledge graphs (Sun et al. 2018; Wang et al. 2018a). Moreover, as DLMs have achieved superior performance in CV and NLP, another important research line focuses on leveraging non-structural side information for more effective recommendations, including visual content (Niu et al. 2018; Liu et al. 2017; Chen et al. 2017) and textual content (Catherine and Cohen 2017; Zheng et al. 2017b; Seo et al. 2017; Tay et al. 2018). Therefore, this section aims to provide in-depth analysis on DLMs with various types of side information.[3]

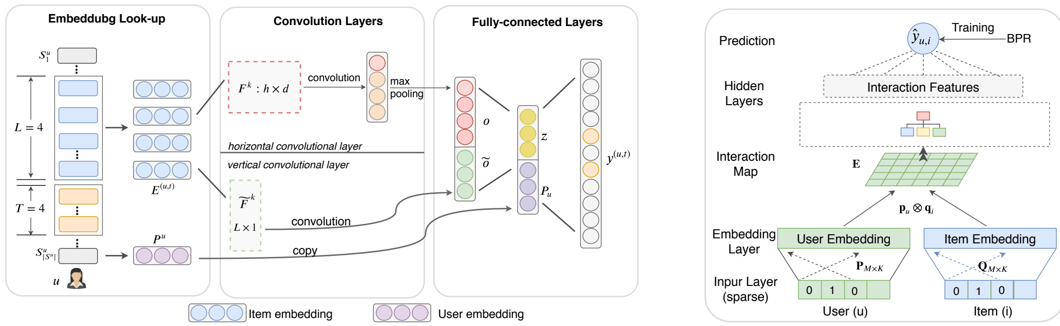
[3]Here, for facilitating the presentation, we consider all artificial neural networks as deep learning models, including the ones with one hidden layer (e.g., shallow auto-encoder).

4.1 Basic deep learning models

We first provide an overview of the basic DLMs without integration of side information. These methods, though, merely take into account the user-item historical interaction data, have achieved significant improvements on the recommendation performance due to the superiority of DLMs. They can be broadly classified into five categories as introduced below. We provide a relatively detailed elaboration of these models as they are the bases for more sophisticated deep learning models with side information.

Auto-Encoder based methods. Auto-Encoder is the simplest neural network with three layers which projects (encodes) the high-dimensional input layer into a low-dimensional hidden layer, and finally re-projects (decodes) the hidden layer to the output layer. The goal is to minimize the reconstruction error, that is, to find the most efficient compact representations for the input data. One early work was AutoRec proposed by (Sedhain et al. 2015), as illustrated by Fig. 9(a). It adopts fully-connected layers to project the partially observed user or item vectors into a low-dimensional hidden space, which is then reconstructed into the output space to predict the missing ratings.

MLP based methods. MLP is short for *multi-layer perceptron* (Rumelhart et al. 1985), which contains one or more hidden layers with arbitrary activation functions providing levels of abstraction. Thus, it is a universal network to extract the high-level features for approximating the user-item interactions. Based on MLP, He et al. (2017) proposed the *neural collaborative filtering* (NCF) framework which tries to integrate generalized matrix factorization (GMF) with MLP: (1) GMF applies a linear kernel to model the user-item interactions in latent space; and (2) MLP uses a non-linear kernel to learn the user-item interaction function from the data. Fig. 8b shows a way of fusing GMF and MLP (called the NeuMF model), where their outputs are concatenated and fed into the NeuMF layer. Xue et al. (2017) designed a *deep matrix factorization* model (DMF), which exploits multi-layer non-linear projections to learn the user and item representations by making use of both explicit and implicit feedbacks (See Fig. 8c).



(a) Overview of Caser

(b) Overview of ConvNCF

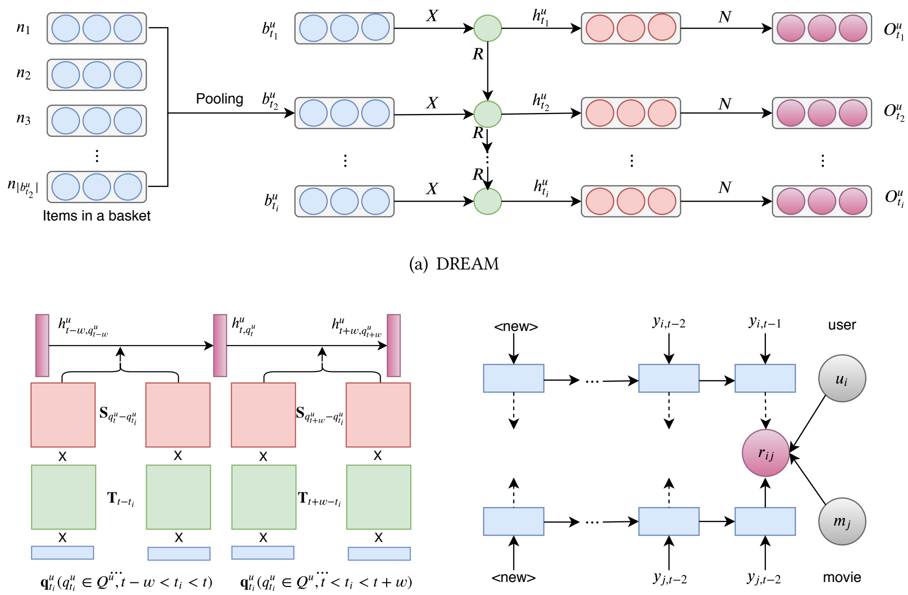
Figure 9: (a) the architecture of Caser, where the dash rectangular boxes are convolutional filters with different sizes, It uses previous 4 actions (L = 4) to predict which items user *u* will interact with in next 2 steps (T = 2) (Tang and Wang 2018); (b) the overall framework of ConvNCF, where the correlation of user and item embeddings is expressed by the outer product, and then CNN is adopted to learn the high-level abstract correlation (He et al. 2018a).

CNN based methods. In essence, *convolutional neural network* (CNN) (LeCun and Bengio 1995) can be treated as a variant of MLP. It takes input and output with fixed sizes, and its hidden layers typically consist of convolutional layers, pooling layers, and fully connected layers. By regarding the input data as an image, CNN can be utilized to help capture the local features. For instance, Tang and Wang (2018) proposed Caser for next item recommendation, as depicted in Fig. 9a. It embeds a sequence of recent interacted items into a latent space which is considered as an image. Convolutional filters for two directions are then adopted: (1) horizontal filters facilitate to capture union-level patterns with multiple union sizes; and (2) vertical filters help capture point-level sequential patterns through weighted sums over latent representations of previous items. He et al. (2018a) designed ConvNCF, as shown by Fig. 9b. It first utilizes an outer product (interaction map) to explicitly model the pairwise correlations between user and item embeddings, and then employs CNN to learn high-order correlations among embedding dimensions from locally to globally in a hierarchical way.

RNN based methods. *Recurrent neural network* (RNN) (Collobert et al. 2011) has been introduced in recommendation tasks mainly for temporal recommendation and sequential recommendation (or next item recommendation), as it is capable of memorizing historical information and finding patterns across time. For instance, Hidasi et al. (2015) developed SeRNN for session-based next-item recommendation. It is built upon the *gated recurrent unit* (GRU) (Cho et al. 2014a), a more elaborate model of RNN for dealing with the vanishing gradient problem. Yu et al. (2016) designed a *dynamic recurrent basket model* (DREAM) based on RNN, which not only learns a dynamic representation of a user but also captures the global sequential features among the baskets, as illustrated by Fig. 10a. Jing and Smola (2017) devised NSR based on *long-short term memory* (LSTM) (Hochreiter and Schmidhuber 1997) to estimate when a user will return to a site and predict her future listening behavior. Liu et al. (2016b) proposed a novel *spatial temporal RNN* (ST-RNN) for next-location recommendation as depicted in Fig. 10b, which models both local temporal and spatial contexts in each layer with time-specific as well as distance-specific transition matrices, respectively. Wu et al. (2017b) proposed RRN to capture both the user and item temporal dynamics by endowing both users and items with a LSTM auto-regressive model, as depicted by Fig. 10c.

Attention based methods. Motivated by the human visual attention nature and attention mechanisms in natural language processing (Yang et al. 2016b) and computer vision (Pei et al. 2017a; Xu et al. 2015), attention has gained tremendous popularity in the community of recommender systems. It mainly aims to cope with the data noisy problem by identifying relevant parts of the input data for modeling the user-item interactions (Pei et al. 2017b). The standard vanilla attention mechanism learns the attention scores for the input data by transforming the representations of input data via fully-connected layers, and then adopting an extra softmax layer to normalize the scores (Pei et al. 2017b; Chen et al. 2017; Wang et al. 2018c). Normally, the attention mechanism often cooperates with either RNN to better memorize very long-range dependencies, or CNN to help concentrate on the important parts of the input. For instance, Feng et al. (2018) designed DeepMove for user mobility prediction depicted by Fig. 11a. It designs RNN to capture the sequential transitions contained in the current trajectory, and meanwhile proposes a historical attention model to capture the mobility regularity from the lengthy historical records.

Recently, self-attention (Vaswani et al. 2017) has started to gain exposure, as it can replace RNN and CNN in the sequence learning, achieving better accuracy with lower computation complexity. It focuses on co-learning and self-matching of two sequences whereby the attention weights of one sequence are conditioned on the other sequence, and vice verse (Zhang et al. 2018). Inspired by self-attention, Zhang et al. (2018) proposed a novel sequence-aware recommendation model, AttRec, by considering both short- and long-term user interests shown as Fig. 11b. It utilizes self-attention mechanism (Vaswani et al. 2017) to estimate the relative weights of each item in the user's interaction trajectories to learn better representations for the userâĂŹs transient interests. Similarly, Kang and McAuley (2018) also developed a self-attention based sequential



(a) DREAM

(b) ST-RNN

(c) RRN

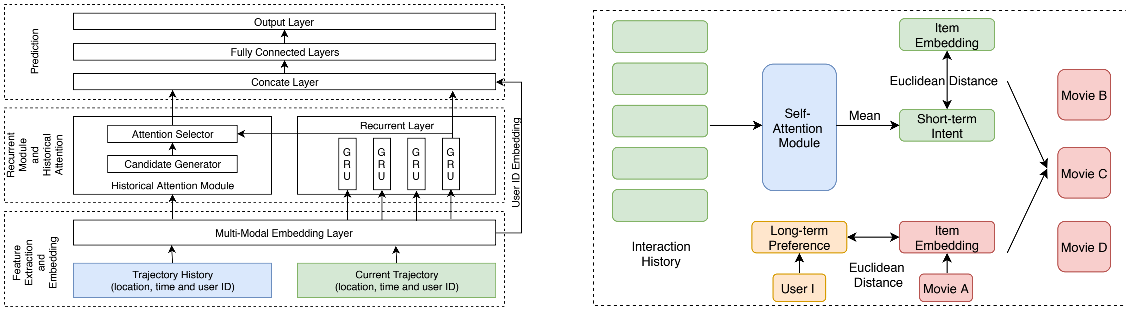
Figure 10: (a) the overall framework of DREAM, where the pooling operation on the items in a basket aims to get the representation of the basket. The input layer comprises a series of basket representations of a user. The hidden layer handles the dynamic representation of the user, and the output layer shows scores of this user for all items (Yu et al. 2016); (b) ST-RNN injects the time- and distance-specific transition matrices to the input embedding (i.e., embedding of location visited by user u at time $t_i - q^u_{t_i}$) of RNN at each time step (Liu et al. 2016b); (c) RRN utilizes individual recurrent networks to address the temporal evolution of user and movie state respectively. The state evolution for a user depends on which movies (and how) a user rated previously. Likewise, a movie's parameters are dependent on the users that rated it in the previous time interval and its popularity among them. To capture stationary attributes, it adopts an additional (conventional) set of auxiliary parameters u_i and m_j for users and movies respectively (Wu et al. 2017b).

Table 11: Classifications of basic deep learning models for recommendation, where 'AE' denotes auto-encoder; 'Attn' refers to attention.

Type	Representative Method
AE	(1) AutoRec (Sedhain et al. 2015);
MLP	(1) NCF (He et al. 2017); (2) DMF (Xue et al. 2017)
CNN	(1) Caser (Tang and Wang 2018); (2) ConvNCF (He et al. 2018a)
RNN	(1) SeRNN (Hidasi et al. 2015); (2) DREAM (Yu et al. 2016) (3) NSR (Jing and Smola 2017); (4) RRN (Wu et al. 2017b)
Attn	(1) DeepMove (Feng et al. 2018); (2) AttRec (Zhang et al. 2018) (3) SASRec (Kang and McAuley 2018)

model (SASRec) that can capture the long-term user interests, but makes the predictions based on relatively few actions. It identifies which items are 'relevant' from a user's action history with an attention mechanism.

Summary of basic DLMs. Table 11 summarizes the representative basic DLMs that are the essential bases, and can be readily adapted for DLMs with side information. First, Auto-Encoder, as the simplest neural network, can be extended to fuse both structural and non-structural side information by learning the contextual representations of items from flat features (e.g., item categories) (Dong et al. 2017), text (e.g., articles, reviews) (Okura et al. 2017) or image (e.g., movie posters) (Zhang et al. 2016) features, as we will discuss later. Second, as a universal network, MLP helps efficiently extract the high-level user and item representations for better recommendations. Besides, it can be easily extended to fuse structural side information by concatenating flat features with user or item embeddings as the input data (Cheng et al. 2016; Covington et al. 2016; Niu et al. 2018). Third, CNN is extensively exploited to capture the spatial patterns, that is, the local relations among the features in the "image" format data with fixed input and output lengths. Thus, it is more capable of coping with non-structural side information, such as texts and images. Fourth, RNN is generally employed to capture sequential patterns or temporal dynamics with arbitrary input and output lengths. Hence, it is more suitable for sequential recommendation to predict what next item that users will be interested in (Yao et al. 2017; Zhang et al. 2017b), or explainable recommendation to generate texts (e.g., review, tips) (Li et al. 2017; Lu et al. 2018). Last, the emergence of vanilla attention mechanisms further advance existing neural networks (e.g., CNN, RNN) by explicitly distinguishing the different importance of the input data. Self-attention mechanisms can revolutionize the deep learning structures. To sum up, all of the above methods are foundations of DLMs with side information taken into account (as summarized in Table 12), and will be elaborated in the following subsections.



(a) DeepMove

(b) AttRec

Figure 11: (a) the overall framework of DeepMove, where the Recurrent Layer captures sequential transitions contained in the current trajectory, and the Historical Attention Module captures the mobility regularity from the historical trajectories. The Concate Layer combines all the features from Attention, Recurrent and Embedding modules into a new vector (Feng et al. 2018); (b) the overview of AttRec, where both short- and long-term user's interests have been considered, and the short-term interest is learned via a self-attention mechanism (Zhang et al. 2018).

4.2 Deep learning models with flat features (DLMs+FFs)

Plenty of DLMs have incorporated flat features (e.g., user gender, item categories) for better recommendations.

Auto-Encoder based methods. Dong et al. (2017) developed a hybrid recommender, HDS, which makes use of both the user-item rating matrix and flat features. It learns deep user and item representations based on two *additional stacked denoising auto-encoders* (aSDAEs) (Vincent et al. 2010) with the side information as the input (e.g., user gender, item categories). The aSDAEs are jointly trained with the matrix factorization to minimize the rating estimation error as well as aSDAE reconstruction error. Okura et al. (2017) presented a news recommender, ENR, which first learns distributed representations of articles based on a variant of a denoising autoencoder (Vincent et al. 2008). The model is trained in a pair-wise manner with a triplet (a_0, a_1, a_2) as input to preserve their categorical similarity, where the articles a_0, a_1 are in the same category and a_0, a_2 belong to different categories. Then, it generates user representations by using a *recurrent neural network* (RNN) with browsing histories as input sequences, and finally matches and lists articles for users based on inner-product operations.

MLP based methods. Cheng et al. (2016) designed Wide&Deep to jointly train wide linear regression models and deep neural networks, where the categorical features are converted into low-dimensional embeddings, and then fed into the hidden layers of the deep neural network. Covington et al. (2016) introduced DNN for video recommendation on Youtube (www.youtube.com). It includes: (1) a deep candidate generation model where a user's watch history and side information such as the demographic features of users, are concatenated into a wide layer followed by several fully connected layers with ReLU (Nair and Hinton 2010) to generate video candidates that users are most likely to watch; and (2) a deep ranking model which has similar architecture as the candidate generation model to assign a ranking score to each candidate video. Niu et al. (2018) proposed a pair-wise image recommender, NPR, with the fusion of multiple contextual information including tags,

Table 12: Summary of state-of-the-art deep learning based recommendation algorithms with side information, where 'FFs, NFs, FHs, KGs' represent structural side information, namely *flat features, network features, feature hierarchies* and *knowledge graphs*; 'TFs, IFs, VFs' denote the non-structural side information, namely *text features, image features* and *video features*.

Algorithm	Venue	Year	Structural				Non-Struct.			Reference
			FFs	NFs	FHs	KGs	TFs	IFs	VFs	
Wide&Deep	RecSys	2016	✓	–	–	–	–	–	–	Cheng et al.
DNN	RecSys	2016	✓	–	–	–	✓	–	–	Covington et al.
CDL-Image	CVPR	2016	✓	✓	–	–	–	✓	–	Lei et al.
HDS	AAAI	2017	✓	–	–	–	–	–	–	Dong et al.
IARN	CIKM	2017	✓	–	✓	–	–	–	–	Pei et al.
ENR	KDD	2017	✓	–	–	–	✓	–	–	Okura et al.
SH-CDL	TKDE	2017	✓	–	–	–	✓	–	–	Yin et al.
NPR	WSDM	2018	✓	–	–	–	–	✓	–	Niu et al.
3D-CNN	RecSys	2017	–	–	✓	–	✓	–	–	Tuan and Phuong
NEXT	arXiv	2017	–	✓	–	–	✓	–	–	Zhang et al.
BayDNN	CIKM	2017	–	✓	–	–	–	–	–	Ding et al.
DeepSoR	AAAI	2018	–	✓	–	–	–	–	–	Fan et al.
GraphRec	arXiv	2019	–	✓	–	–	–	–	–	Fan et al.
CKE	KDD	2016	–	–	–	✓	✓	✓	–	Zhang et al.
TransNets	RecSys	2017	–	–	–	✓	–	–	–	Catherine and Cohen
PACE	KDD	2017	–	–	–	✓	✓	–	–	Yang et al.
RKGE	RecSys	2018	–	–	–	✓	–	–	–	Sun et al.
KPRN	AAAI	2018	–	–	–	✓	–	–	–	Wang et al.
DKN	WWW	2018	–	–	–	✓	✓	–	–	Wang et al.
RippleNet	CIKM	2018	–	–	–	✓	–	–	–	Wang et al.
RippleNet-agg	TOIS	2019	–	–	–	✓	–	–	–	Wang et al.
RCF	arXiv	2019	–	–	–	✓	–	–	–	Xin et al.
KGCN	arXiv	2019	–	–	–	✓	–	–	–	Wang et al.
KGCN-LS	arXiv	2019	–	–	–	✓	–	–	–	Wang et al.
MKR	arXiv	2019	–	–	–	✓	–	–	–	Wang et al.
KTUP	arXiv	2019	–	–	–	✓	–	–	–	Cao et al.
KGAT	arXiv	2019	–	–	–	✓	–	–	–	Wang et al.
CDL	KDD	2015	–	–	–	–	✓	–	–	Wang et al.
SERM	CIKM	2017	–	–	–	–	✓	–	–	Yao et al.
JRL	CIKM	2017	–	–	–	–	✓	✓	–	Zhang et al.
DeepCoNN	WSDM	2017	–	–	–	–	✓	–	–	Zheng et al.
NRT	SIGIR	2017	–	–	–	–	✓	–	–	Li et al.
D-Attn	RecSys	2017	–	–	–	–	✓	–	–	Seo et al.
RRN-Text	arXiv	2017	–	–	–	–	✓	–	–	Wu et al.
MT	RecSys	2018	–	–	–	–	✓	–	–	Lu et al.
MPCN	KDD	2018	–	–	–	–	✓	–	–	Tay et al.
Exp-Rul	AAAI	2017	–	–	–	–	–	✓	–	Alashkar et al.
ACF	SIGIR	2017	–	–	–	–	–	✓	✓	Chen et al.

Table 13: Classifications of DLMs+FFs, where 'AE' denotes auto-encoder; 'Pre-f, Conc, Proj' represent *Pre-filter, Concatenate, Projection*, repsectively.

Method	Model Type				FF Usage Type			Reference
	AE	MLP	CNN	RNN	Pre-f.	Conc.	Proj.	
ENR	✓				✓			Okura et al. 2017
HDS	✓						✓	Dong et al. 2017
Wide&Deep		✓				✓		Cheng et al. 2016
DNN		✓				✓		Covington et al. 2016
NPR		✓					✓	Niu et al. 2018
CDL-Image			✓			✓		Lei et al. 2016
IARN				✓			✓	Pei et al. 2017b

geographic and visual features. It adopts one fully-connected layer and element-wise product to learn the user's contextual (i.e., topic, geographical and visual) preference representations, which is concatenated with her general preference representation in the merged layer. Finally, the merged preference representation is connected with the feed-forward network to generate recommendations.

CNN based methods. Lei et al. (2016) developed a pair-wise learning method (CDL-Image) for image recommendation. Three sub-networks are involved, wherein two identical sub-networks are used to learn representations of positive and negative images for each user via CNN, and the remaining one is used to learn the representation of user preference via four fully-connected layers. The input user vectors for this network are vectors of relevant tags generated by Word2Vec.

RNN based methods. Pei et al. (2017b) devised a 'RNN+Attention' based approach (IARN), which is similar as RRN (Wu et al. 2017a). It uses two RNNs to capture both user and item dynamics. The estimated rating of user u on item i is the inner product of the corresponding hidden representations, which are transformations of the final hidden states in the two RNNs. Furthermore, it learns the attention scores of user and item history in an interactive way to capture the dependencies between the user and item dynamics. Feature encoder is used to fuse a set of categories K that item i belongs to, where a category $k \in K$ is modeled as a transformation function $\mathbf{M}_i^k$ that projects the item embedding $\mathbf{e}_i$ into a new space. Therefore, the influence of K flat categories is denoted by the sum of their respective impacts, that is, $\sum_{k=1}^{K} \mathbf{M}_i^k \cdot \mathbf{e}_i$.

Summary of DLMs+FFs. The flat features are generally incorporated into various DLMs in three different ways, as summarized in Table 13: (1) *pre-filtering* is the simplest way. For instance, ENR (Okura et al. 2017) adopts item categories to pre-select the positive (within the same or similar categories) and negative (across different categories) articles; (2) *concatenation* is the most straightforward way. For instance, Wide&Deep (Cheng et al. 2016), DNN (Covington et al. 2016), and CDL-Image (Lei et al. 2016) directly concatenate all feature vectors together and feed them into the proposed network architecture. This is quite similar to what SVDFeature does in LFMs which directly adds feature representations to corresponding user and item representations. Despite their success on lifting accuracy, the flat features are mostly exploited in a coarse fashion by mere concatenation; and (3) *projection* is the most fine-grained way in comparison with the above two ways. For example, HDS (Dong et al. 2017) and NPR (Niu et al. 2018) employ neural networks to learn deep user or item representations under different features, that is, the user and item contextual representations. By doing so, the special information from different flat features is taken into consideration. In other words, more elaborate design on feature fusion, together with the natural superiority of DLMs, may bring extra performance increments on recommendation.

4.3 Deep learning models with network features (DLMs+NFs)

In the proposed CNN based pair-wise image recommendation model (CDL-Image) (Lei et al. 2016), social networks are utilized to help exclude negative images for each user. In particular, they assign the images to be negative, which do not have tags indicating the interests of the user and her friends. The social network is merely used to do pre-filter. Zhang et al. (2017b) proposed a MLP-based method, named NEXT for next POI recommendation. Similar to SVDFeature (Chen et al. 2012), the embeddings of user and item are influenced by those of corresponding meta-data (i.e., social friends, item descriptions). Based on this, the user and item embeddings are further modeled by one layer of feed-forward neural network supercharged by ReLU (Nair and Hinton 2010) for generating recommendations. The exploitation of social networks in this approach is exactly the same as that in LFMs.

Later, Ding et al. (2017) designed a CNN-based method (BayDNN) for friend recommendation in social networks. It first exploits CNN to extract latent deep structural feature representations by regarding the input network data as an image and then adopts the Bayesian ranking to make friend recommendations. This is fairly the first work that proposes an elegant and unified deep learning approach with social networks. After that, Fan et al. (2018) proposed a rating prediction model (DeepSoR), which first uses Node2Vec (Grover and Leskovec 2016) to learn the embeddings of all users in the social network, and then takes the averaged embeddings associated with the k most similar neighbours for each user, as the input of a MLP architecture, so as to learn non-linear features of each user from the social relations. Finally, the learned user features are integrated into the probabilistic matrix factorization for rating prediction. Base on this, they further proposed a novel attention neural network (GraphRec) (Fan et al. 2019) to jointly model the user-item interactions and user social relations. First, the user embedding is learned via aggregating both (1) the user's opinions towards interacted items with different attention scores and (2) the influence of her social friends with different attention scores. Analogously, the item embedding is learned via attentively aggregating the opinions of users who have interacted with the item. All attention scores are automatically learned through a two-layer attention neural network. Finally, the user and item embeddings are concatenated and fed into a MLP for rating prediction.

Summary of DLMs+NFs. First, similar as DLMs+FFs, the usage of NFs in DLMs evolves from simple pre-filtering, to moderate summation via SVDFeature, and finally to deep projection. This further helps support our point that the exploitation of side information in LFMs could provide better clues and guidance for DLMs. Second, extensive experimental results have demonstrated that DLMs+NFs consistently defeat LFMs+NFs in terms of their recommendation accuracy. However, we also notice that the studies on fusing NFs into DLMs are quite fewer than those on LFMs+NFs. On the other hand, there is still a great demand on social friend recommendations with the rapid development of social network platforms, such as Facebook, Twitter and so forth, which calls for more in-depth investigation on DLMs+NFs. Third, as the NFs can be considered as a graph, other graph related DLMs can also be adopted, including *graph convolutional networks* (GCN) (Duvenaud et al. 2015; Niepert et al. 2016) and *graph neural network* (GNN) (Scarselli et al. 2008). They have attracted considerable and increasing attention due to their superior learning capability on the graph-structured data. Last, the idea of leveraging distrust information and trust propagation in LFMs+NFs is still worthy of exploration in DLMs+NFs.

4.4 Deep learning models with feature hierarchies (DLMs+FHs)

Tuan and Phuong (2017) proposed a 3D-CNN framework that combines session clicks and content features (i.e., item descriptions and category hierarchies) into a 3-dimensional CNN with character-level encoding of all the input data. To utilize the information encoded in the hierarchy, they concatenated the current category with all its ancestors up to the root and use the resulting sequence of characters as the category feature, for example, "apple/iphone/iphone7/accessories". However, this method cannot distinguish the different impacts of categories at different layers of the hierarchy. In IARN (Pei et al. 2017b), the feature encoder is used to fuse a set of categories K that item i belongs to, where a category k is modeled as a transformation function $\mathbf{M}_i^k$ that projects the item embedding $\mathbf{e}_i$ into a new space. In terms of the hierarchical categories, it considers the recursive parent-children relations between categories from the root to leaf layer, that is, $\prod_{k=1}^{L} \mathbf{M}_i^k \cdot \mathbf{e}_i$. With the recursive projection, the item can be gradually mapped into a more general feature space.

Summary of DLMs+FHs. DLMs+FHs is also inadequately studied compared with either LFMs+FHs or DLMs with other side information (e.g., FFs, KGs, TFs, IFs), although their effectiveness for high-quality recommendations has been empirically verified (Tuan and Phuong 2017; Pei et al. 2017b). We argue that the advantages of investigation on DLMs+FHs lie in three aspects: (1) FHs are more easily and cost-effectively obtained in real-world applications (e.g., Amazon, Tmall) compared with other complex side information, such as knowledge graphs, textual reviews and visual images; (2) FHs provide human- and machine-readable descriptions about a set of features and their relations (e.g., affinity, alternation, and complement). The rich information encoded in FHs could enable more accurate and diverse recommendations; (3) the volumes of FHs are generally much smaller than other complicated side information, such as KGs, text reviews and visual images. Thus, the integration of FHs will deliver a flexible DLM with less computational cost and high efficiency. In addition, the categories at different layers of the hierarchy have different impacts on characterizing user-item interactions, as already proven by LFMs+FHs. To achieve a better exploitation on FHs for more effective recommendation results, such different impacts may be learned by deep learning based advances (e.g., attention mechanism) in an automatic fashion.

4.5 Deep learning models with knowledge graphs (DLMs+KGs)

According to how KGs are exploited, three types of approaches under DLMs+KGs are included: graph embedding based methods, path embedding based methods and propagation based methods.

Graph embedding based methods. Many KG-aware recommendation approaches directly make use of conventional graph embedding methods, such as TransE (Bordes et al. 2013), TransR (Lin et al. 2015), TransH (Yang et al. 2014) and TransD (Ji et al. 2015), to learn the embeddings for the KG. The learned embeddings are then incorporated into recommendation models. For instance, Zhang et al. (2016) proposed a *collaborative knowledge graph embedding* method (CKE) that leverages TransR to learn better item representations. This is jointly trained with the item visual and textual representation learning models in a unified Bayesian framework. Wang et al. (2018c) proposed DKN for news recommendation. It exploits CNN to generate embedding for a news i based on its title t_i, where a sub-KG related to t_i is extracted and learned based on conventional graph embedding methods like TransD. Then the learned embeddings of entities and corresponding context in the sub-KG, as well as the embeddings of words in t_i, are treated as different channels and stacked together as the input of a CNN to learn the news embeddings.

Recently, Cao et al. (2019) proposed KTUP to jointly learn the recommendation model and knowledge graph completion. Inspired by TransH, they came up with a new translation-based recommendation model (TUP) that automatically induces a preference for a user-item pair, and learns the embeddings of preference $\mathbf{p}$, user $\mathbf{u}$ and item $\mathbf{i}$, satisfying $\mathbf{u} + \mathbf{p} \approx \mathbf{i}$. KTUP further extends TUP to jointly optimize TUP (for the item recommendation) and TransH (for the KG completion) to enhance the item and preference modeling by transferring knowledge of entities and relations from the KG, respectively. Similarly, Xin et al. (2019) proposed RCF to generate recommendations based on both the user-item interaction history and the item relational data in the KG. They developed a two-level hierarchical attention mechanism to model user preference: the first-level attention discriminates which types of relations are more important; and the second-level attention considers the specific relation values to estimate the contribution of an interacted item. Finally, they jointly modeled the user preference via the hierarchical attention mechanism and item relations via the KG embedding method (i.e., DistMult) (Yang et al. 2014).

Other methods adopt deep learning advances to learn the embeddings for the KG. For example, Wang et al. (2019c) developed a multi-task learning approach, MKR, which utilizes the KG embedding task as the explicit constraint term to provide regularization for the recommendation task. The recommendation module takes a user and an item as input, and uses MLP and cross&compress units to output the predicted scores; the KG embedding module also uses MLP to extract features from the head h and relation r of a knowledge triple $\langle h, r, t \rangle$, and outputs the representation of the predicted tail t; and the two modules are bridged by the cross&compress units, which automatically share the latent features and learn the high-order interactions between the items in recommender systems and the entities in the KG.

Path embedding based methods. Typically, path embedding based methods extract connected paths with different semantics between the user-item pairs, and then encode these paths via DLMs. For instance, Sun et al. (2018) proposed a *recurrent knowledge graph embedding* method (RKGE). It first extracts connected paths between a user-item pair in the KG, representing various semantic relations between the user and the item. Then these extracted paths are encoded via a batch of RNNs to learn the different path influences on characterizing the user-item interactions. After that, a pooling layer is incorporated to help distinguish the different saliency of these paths on modeling the user-item interaction. Finally, a fully-connected layer is employed to estimate the user's preference for each item. Following the same idea, Wang et al. (2018a) later designed another KG-aware recommender, KPRN, by further taking into account different entity types and relations when encoding the extracted paths via RNNs.

Propagation based methods. These KG-aware recommendation approaches take advantage of both graph embedding based methods and path based methods. Instead of directly extracting paths in a KG, *propagation* is adopted to discover high-order interactions between items in recommender systems and entities in the KG, which is equivalent to the automatic path mining.

Wang et al. (2018b, 019a, 2019b, 2019d) designed a series of KG-aware recommendation methods based on the idea of *propagation*. Specifically, Wang et al. (2018b, 019a) developed RippleNet which naturally incorporates graph embedding based methods (i.e., three-way tensor factorization) by preference propagation. In particular, it treats user u's interacted items as a set of seeds in the KG, and extends iteratively along the KG links to discover her l-order interests ($1 \leqslant l \leqslant 3$). Based on this, it learns user u's l-order preference with respect to item v, respectively, which are then accumulated together as user u's hierarchical preference to item v. Later, they further proposed KGCN (Wang et al. 2019d) based on *graph convolutional networks* (GCN) (Niepert et al. 2016; Duvenaud et al. 2015), which outperforms RippleNet. It computes the user-specific item embeddings by first applying a trainable function that identifies important KG relations for a given user and then transforming the KG into a user-specific weighted graph. It further applies GCN to compute the embedding of an item by propagating and aggregating neighborhood information in the KG. After that, to provide better inductive bias, they upgraded KGCN to KGCN-LS (Wang et al. 2019b) by using *label smoothness* (LS), which provides regularization over edge weights and has proven to be equivalent to label propagation scheme on a graph.

Recently, Wang et al. (2019a) designed KGAT, which recursively and attentively propagates the embeddings from a node to its neighbors in the KG to refine the node's embedding. Specifically, it first uses TransR (Lin et al. 2015) to learn the KG embeddings, and then employs the *graph convolution network* (GCN) (Duvenaud et al. 2015; Niepert et al. 2016) to recursively propagate the embeddings along the high-order connectivity, and meanwhile generates attentive weights to reveal the saliency of such connectivity via graph attention network (Veličković et al. 2017).

Summary of DLMs+KGs. Table 14 summarizes the classifications of DLMs+KGs in terms of fundamental model types and KG usage types. We further summarize the studies from the following

Table 14: Classifications of DLMs+KGs, where 'AE' denotes auto-encoder; 'Attn' means attention; 'KGE' refers to knowledge graph embedding; and 'Prop.' indicates propagation.

Method	Model Type					KG Usage Type			Reference
	AE	MLP	CNN	RNN	Attn	KGE	Path	Prop.	
CKE	✓					✓			Zhang et al. 2016
KTUP						✓			Cao et al. 2019
RCF		✓			✓	✓			Xin et al. 2019
DKN			✓		✓	✓			Covington et al. 2016
MKR		✓				✓			Wang et al. 2019c
RKGE				✓			✓		Sun et al. 2018
KPRN				✓			✓		Wang et al. 2018a
KGAT			✓		✓			✓	Wang et al. 2019a
RippleNet					✓			✓	Wang et al. 2018b
KGCN			✓					✓	Wang et al. (2019b, 2019d)

perspectives. First, DLMs+KGs show a great superiority compared to LFMs+KGs in terms of recommendation accuracy. However, the high computational cost limits the scalability of DLMs+KGs on large-scale datasets. Hence, a promising direction for boosting DLMs+KGs approaches should focus on improving their scalability, and thus reduce time complexity. Second, empirical studies have demonstrated the strength of propagation (i.e., hybrid) based methods against those either exploiting graph embedding or path embedding only. Third, regardless of the KG usage types, most of these methods rely on conventional KG embedding methods such as TransE/R/H/D to incorporate KG for better recommendations. In particular, they learn embeddings for the KG based on the triple $\langle h, r, t \rangle$, where h, t separately represent head and tail entities, and r denotes entity relations. In contrast, several methods attempt to employ deep learning advances, for example, *graph convolution network* (GCN) is developed to further boost the quality of recommendations. Thus, more further efforts should be devoted to this topic. Lastly, a better exploitation of the heterogeneity of KGs will facilitate more accurate recommendation results. For instance, by additionally distinguishing entity types and relation types, KPRN (Wang et al. 2018a) performs better than RKGE (Sun et al. 2018), and by attentively identifying the saliency of different relation types, KGCN (Wang et al. 2019b) outperforms RippleNet (Wang et al. 2018b).

4.6 Deep learning models with text features (DLMs+TFs)

There are a number of studies that incorporate textual features (e.g., reviews, tips, and item descriptions) into DLMs for better recommendations.

Auto-Encoder based methods. Wang et al. (2015c) proposed CDL based on the *stacked denoising autoencoder* (SDAE) (Vincent et al. 2010) to learn the item representation with item content (i.e., paper abstract and movie plots) as input. With the learned item representations by SDAE as bridge, CDL simultaneously minimizes the rating estimation error via matrix factorization and SDAE reconstruction error. In CKE (Zhang et al. 2016), SDAE is used to extract item textual representations from textual knowledge (i.e., movie and book summaries). This is jointly trained with the matrix factorization and visual representation learning models. In addition, ENR (Okura et al. 2017) adopts a variant of a denoising autoencoder (Vincent et al. 2008) to learn the distributed representations of articles for news recommendation. Yin et al. (2017) proposed SH-CDL which uses *deep belief network* (DBN), an auto-encoder model (Hinton et al. 2006), to learn the item hidden representations $\mathbf{f}_i$ by feeding the textual content (i.e., categories, descriptions and comments). It unifies matrix factorization and DBN by linking the item latent vector $\mathbf{q}_i$ and its hidden representation $\mathbf{f}_i$ under the assumption that $\mathbf{q}_i$ follows a normal distribution with the mean $\mathbf{f}_i$.

MLP based methods. In the video recommendation method (DNN) (Davidson et al. 2010), a user's watch/impression history, that is, a set of IDs associated with the videos that a user has visited before, and side information (e.g., the user's tokenized queries) are concatenated into a wide layer, followed by several layers of fully connected ReLU (Nair and Hinton 2010) to rank the recommendation candidates. In NEXT (Zhang et al. 2017b), the embedding of an item is influenced by that of its description (split into multiple words), and modeled via one fully-connected feed-forward layer supercharged by ReLU (Nair and Hinton 2010). Zhang et al. (2017a) developed a joint representation learning approach (JRL), which jointly learns user and item representations from three types of sources: (1) textual reviews via PV-DBOW (Le and Mikolov 2014); (2) visual images via CNN; and (3) numerical ratings with a two-layer fully-connected neural network. The learned user and item representations from different sources are respectively concatenated together for pair-wise item ranking.

CNN based methods. The 3D-CNN framework (Tuan and Phuong 2017) combines session clicks and content features (i.e., item descriptions and category hierarchies) into a 3-dimensional CNN with character-level encoding of all input data. Zheng et al. (2017b) designed DeepCoNN to learn both user and item representations in a joint manner using textual reviews. Parallel and identical user and item networks are involved: in the first layer, all reviews written by a user (or written for an item) are represented as matrices of word embeddings to capture the semantic information. The next layer employs CNN to extract the textual features for learning user and item representations. Finally, a shared layer is introduced on the top to couple these two networks together, which enables the learned user and item representations to interact with each other in a similar manner as factorization machine (Rendle 2010, 2012).

Catherine and Cohen (2017) further extended DeepCoNN by proposing TransNets to help address the issue that a pair-wise review for the target user to the target item may not be available in testing procedure. It thus uses an additional transform layer to transform the latent representations of user and item into those of their pair-wise review. In the training, this layer is regularized to be similar to the real latent representation of the pair-wise review learned by the target network. Therefore, in the testing, an approximate representation of the pair-wise review can be generated and used for making predictions.

Seo et al. (2017) developed D-Attn to jointly learn better user and item representations using CNN with local (L-Attn) and global (G-Attn) attentions. It uses the embeddings of words in the review as input, and adopts both L-Attn and G-Attn to learn the saliency of words with respect to a local window and the entire input texts. These are then fed into two CNNs to learn the respective L-Attn and G-Attn representations of a user (an item), followed by a concatenate layer to get the final user (item) representation. Finally, the inner product of user and item representations is used to estimate a user's preference to an item.

RNN based methods. Yao et al. (2017) proposed SERM for next POI recommendation, which jointly learns the embeddings of multiple features (user, location, time, keyword extracted from text messages) and the transition parameters of the RNN, to capture the spatial-temporal regularities, the activity semantics, as well as the user preferences in a unified way. The embedding layer transforms all features into low-dimensional dense representations, and then concatenates them into a unified representation as the input of RNN module. Wu et al. (2017a) presented a joint *review-rating recurrent recommender network* (RRN-Text). The rating model uses two LSTMs to capture the temporal dynamics of the user and movie representations, which are further combined with stationary user and item representations. Review texts are modeled by a character-level LSTM, and the input character embeddings are fused with both dynamic and stationary user and item representations in the rating model by the bottleneck layer.

Lu et al. (2018) designed a multi-task learning model, MT, for explainable recommendation. It extends *matrix factorization* (MF) model by using the textual features extracted from reviews to serve as the regularizers for the user and item representations. In particular, the embedding of each word in the relevant review is sequentially fed into a bidirectional GRU (Cho et al. 2014a) to learn the textual features for the user (item). Similarly, Li et al. (2017) introduced NRT that simultaneously predicts ratings and generates abstractive tips. It consists of two modules: (1) the neural rating regression module takes the user and item representations as input, and utilizes MLP to predict ratings; and (2) the tips generation module adopts GRU to generate concise tips. Its hidden state is initialized by the user and item representations, and the vectorization of the predicted ratings, as well as the hidden variables from the review text.

Attention based methods. D-Attn (Seo et al. 2017) uses two attention mechanisms to learn the importance of words with respect to a local window and the entire input text for better recommendation performance. Different from the existing DLMs with text features, which either adopt traditional network structures (e.g., TransNets (Catherine and Cohen 2017) and DeepCoNN (Zheng et al. 2017b)), or utilize vanilla attention mechanisms to boost these structures (e.g., D-Attn (Seo et al. 2017)), Tay et al. (2018) proposed a new network architecture, MPCN, to dynamically distinguish the importance of different reviews, instead of treating them equally. In this method, each user (or item) is represented as a sequence of reviews, and each review is constructed from a sequence of words. It first leverages the review-level co-attention to select the most informative review pairs from the review bank of each user and item. Then, it adopts word-level co-attention to model the selected review pairs at word-level. Finally, the learned representations for each review pair at different levels are separately concatenated and passed into the factorization machine for rating prediction.

Summary of DLMs+TFs. Table 15 summarizes all the DLMs+TFs methods. First, compared with LFMs+TFs, which heavily depend on external toolkits to extract knowledge from TFs, DLMs based recommendation approaches seamlessly fuses TFs via deep learning

Table 15: Classification of state-of-the-art methods in DLMs+TFs, where 'AE' denotes auto-encoder; 'Attn' means attention; the methods with early fusion are marked by '*', and the rest are late fusion methods.

Type	Representative Method
AE	(1) CDL (Wang et al. 2015c) (2) CKE (Zhang et al. 2016) (3) ENR (Okura et al. 2017)* (4) SH-CDL (Yin et al. 2017)
MLP	(1) DNN (Davidson et al. 2010)* (2) NEXT (Zhang et al. 2017b)* (3) JRL (Zhang et al. 2017a)
CNN	(1) 3D-CNN (Tuan and Phuong 2017)* (2) DeepCoNN (Zheng et al. 2017b)* (3) TransNets (Catherine and Cohen 2017)* (4) D-Attn (Seo et al. 2017)*
RNN	(1) SERM (Yao et al. 2017)* (2) RRN-Text (Wu et al. 2017a) (3) MT (Lu et al. 2018) (4) NRT (Li et al. 2017)
Attn	(1) D-Attn (Seo et al. 2017)*; (2) MPCN (Tay et al. 2018)*

advances such as CNN and RNN. The homogeneity of underlying methodologies facilitates unified and elegant approaches with excellent recommendation accuracy. Second, the exploitation of TFs by DLMs+TFs is in a deeper and more fine-grained fashion. Recall that most LFMs+TFs leverage TFs in word-, aspect- and sentiment-level. For instance, they either simply apply averaged words embeddings to represent the text, or utilize the results of aspect extraction and sentiment analysis on the text to help infer user preferences. The topic-level methods, though, model latent topic distribution of text with fine granularity via conventional topic modelling models (e.g., LDA (Blei et al. 2003)), cannot capture the complex relations (e.g., non-linearity) encoded in the text better. In contrast, DLMs+TFs take the text embedding at word-level as input, and feed it into deep learning advances (e.g., SDAE, MLP, CNN, RNN), and extract features of the text via multiple non-linear transformations. Meanwhile, neural attention mechanisms can be adopted to distinguish the saliency of each word for the text, and saliency of each review for users and items, so as to support more accurate recommendations. Third, in DLMs+TFs, text features are generally utilized to learn user (item) contextual representations. Based on the stage that text features are integrated, they can be classified into two types: (1) *early fusion* and (2) late fusion. With early fusion, for instance, many methods concatenate the text embedding in word-level with the user (item) embedding, which are together fed into the network. This occurs with DNN (Covington et al. 2016), NEXT (Zhang et al. 2017b), 3D-CNN (Tuan and Phuong 2017) and SERM (Yao et al. 2017). Other methods directly use relevant text embeddings of user (item) as input to learn user- and item-textual representations without concatenating user and item embeddings, like ENR (Okura et al. 2017), DeepCoNN (Zheng et al. 2017b), TransNets (Catherine and Cohen 2017), and D-Attn (Seo et al. 2017), MPCN (Tay et al. 2018). Late fusion methods, in contrast, are often composed of two parallel modules, namely text module and rating module. The text module employs deep learning advances (e.g., SDAE, MLP, CNN) to help learn user (item) textual representations with the relevant text embedding as input; and the rating module takes another deep neural architecture to help learn the plain user and item representations with the user-item interaction data as input. The text module is responsible for regularizing the rating module to assist in learning better user and item representations, thus achieving outstanding recommendation results. These two modules are jointly trained and mutually enhanced. Typical approaches include CDL (Wang et al. 2015c), CKE (Zhang et al. 2016), SH-CDL (Yin et al. 2017) and JRL (Zhang et al. 2017a).

4.7 Deep learning models with image features (DLMs+IFs)

In CKE (Zhang et al. 2016), the stacked convolutional auto-encoder (SCAE) (Masci et al. 2011) is utilized to extract item visual representations from images (i.e., movie poster and book cover). This is jointly trained with the matrix factorization and textural representation learning models to achieve high-quality recommendations. In CDL-Image (Lei et al. 2016), CNN is adopted to extract the high-level features and learn representations of images. The learned image representation together with the user representation learned via four fully-connected layers are fed into a distance calculation net to estimate the user's preference for each image. Another image recommender, NPR (Niu et al. 2018) utilizes the image visual features learned via CNN for better recommendations. After dimension reduction, the image visual representations are fed into a fully-connected layer to learn the representation of each user's contextual preference. In JRL (Zhang et al. 2017a), it jointly learns the user and item representations from three types of sources, namely reviews, images and ratings. It also utilizes a fully-connected layer to learn the image representation which is guided by the raw image features obtained via CNN.

Alashkar et al. (2017) proposed a deep neural network for makeup recommendation with homogeneous style (Exp-Rul). The facial traits are classified automatically and coded as feature vectors, which are then fed into MLP to generate recommendations for each makeup element. The network is trained by examples and guided by rules. In particular, for an automatic analysis on the facial traits, 83 facial landmarks are detected on 900 facial images using the face++ framework (www.faceplusplus.com), and different regions of interest are extracted for different facial attributes. Also, Chen et al. (2017) introduced ACF, a neural network consisting of two attention modules: (1) the component-level attention module learns the user's preference to the selected informative components inside each item, that is, the regions of image and frame of video; and (2) the item-level attention module helps learn the user's preference for the entire item by incorporating the learned component-level attentions with weighted combination. They make use of the widely-used architecture, ResNet-152 (He et al. 2016c), to extract visual features from both the regions of images and frames of videos. The idea is quite similar to D-Attn (Seo et al. 2017) that learns user and item representations from the perspectives of local and global attentions towards the relevant reviews.

Summary of DLMs+IFs. Image features play a crucial role for recommendation tasks in domains such as fashion, restaurants and hotels, as well as image related platforms, such as Flickr and Instagram. They can be used to improve the attractiveness of the recommended items in addition to accuracy. Due to the superior capability of capturing local features, DLMs+IFs is mainly dominated by CNN structures through the use of CNNs to extract visual features from images to generate user (item) visual representations. These visual features are then fed into recommendation frameworks, so as to help regularize and learn high-quality user (item) representations. The major difference between LFMs+IFs and DLMs+IFs is the recommendation framework. That is, LFMs+IFs simply feeds the extracted visual features into linear latent factor models, whereas DLMs+IFs designs proper deep neural architectures with multiple non-linear hidden layers to better accommodate the visual features, and thus achieve more effective recommendation results.

4.8 Deep learning models with video features (DLMs+VFs)

The research studies on fusing videos into DLMs for recommendation tasks are far fewer than research about other types of side information. This may be mainly due to two reasons: (1) the video features are much difficult and time-consuming to be obtained and managed compared with other side information; and (2) the volume of video features is generally quite large, requiring a huge computational cost. One representative approach is the ACF (Chen et al. 2017) mentioned above. In particular, it adopts ResNet-152 (He et al. 2016c) to extract visual features for the frames of videos. To further simplify the process, it uses the output of pool5 layer in ResNet-152, which is actually the mean pooling of the feature maps, as the feature vector for each frame of videos.

4.9 Discussion of DLMs with side information

The extensive and insightful analysis on DLMs with side information in the previous subsections leads to the following conclusions: (1) with deep architectures and non-linear transformations, DLMs have been empirically proven to be extraordinary effective in capturing the highly complex user-item interactions compared with the conventional models including MMs, LFMs and RLMs. On the other hand, it also should be acknowledged that the performance improvements by deep learning advances are often accompanied with heavy computational cost and much longer training time due to the complexity of deep learning models. To meet this end, expensive computation devices such as powerful graph cards are necessary for effective training and inferring. In other words, conventional models are far more efficient than DLMs regarding time complexity in most cases; and (2) due to their high flexibility, DLMs can be easily extended to incorporate various side information. They have shown overwhelming superiority in coping with complex structural data like knowledge graphs and non-structural data including text features and image features. Moreover, conventional models normally need to first do feature engineering to fuse the side information and train the model, while DLMs seamlessly combine these two phases in an end-to-end manner.

We now offer further details on DLMs with side information and provide a summary from the following perspectives. (1) Regarding how to integrate various side information into DLMs, there are mainly three ways: (a) *pre-filtering* is the simplest way to leverage side information to do data pre-processing; (b) *concatenation* is the most straightforward approach because it directly concatenates all side information together; and (c) *projection* is a more fine-grained way by mapping users (items) into low-dimensional space regarding the side information to learn the related contextual representations. (2) In terms of when to integrate the side information into DLMs, they can be broadly grouped into two types: (a) early fusion combines all available side information with user (item) in the input layer, and then feeds them into the network architecture to extract more high-level and complex features; and (b) late fusion uses two parallel modules, with the feature module that aims to learn user (item) contextual representations with respect to the side information, and the rating module that aims to learn plain user and item representations via user-item historical interaction data. The two modules are jointly trained and mutually benefit where the feature module regularizes the rating module, while the rating module in turn guides the feature module. (3) Different types of side information improve recommendation performance from different aspects and are incorporated in different ways. For the first issue, in addition to accuracy, FHs and KGs can help with diversity, while text features facilitate explainable recommendations and image features may assist in attractiveness of recommendation. For the latter issue, simple side information like FFs is generally fused by concatenation in the early fusion, while the graph-structural data (e.g., NFs and KGs) can be accommodated with graph related deep learning advances (e.g., CNN, GCN, GNN). Similar to image features, text features (i.e., word embedding matrices) can be treated as images also, both thus can be fed into CNNs for feature extraction.

5 FUTURE DIRECTIONS

In this Research Commentary, we surveyed recent developments in recommendation with side information. Despite all the progress, there are many challenges to be addressed and plenty of room for improvement. In this section, we identify key challenges and opportunities which we believe can shape future research on this topic. We mainly discuss future directions by considering the following research questions:

- How to further improve deep learning based recommendation with side information in complex structures?
- How to obtain high-quality side information to improve recommendation?
- For which recommendation techniques can side information play an important role, thus should be taken into account?
- In which recommendation scenarios can side information be most valuable?

To answer these questions, we discuss the following challenges and future research directions: deep learning with structured information, leveraging crowdsourcing as means to solicit side information for recommendation; side information for specific recommendation techniques such as reinforcement learning and adversarial recommendation; and side information as an important data source for improving recommendation in specific recommendation scenarios such as cross-domain and package recommendation.

Deep recommenders with structured side information. Integrating side information into deep learning based recommendation is currently an active research topic. Existing approaches, as we have discussed, are highly limited in exploiting the full potential of structured side information for recommendation. The challenges mainly arise from two complications: the intrinsic complexity of structured side information and the difficulty in adapting deep learning models for incorporating structured information.

With knowledge graphs as an example, the current deep learning based recommendation approaches are limited to the usage of the most basic information in a knowledge graph, for example, paths (Sun et al. 2018) or meta-paths (Hu et al. 2018). There is room for improvement by considering higher-level information such as meta-graphs, that is, a collection of linked meta-paths; and hyper-graphs, that is, an abstract view to knowledge graphs that considers entities of the same type as a hyper-node and the connections between hyper-nodes as hyper-edges. Recent work can be found on the hyper-graph based approach for recommendation in [33], though the authors only considered random walk in the hyper-graph for location recommendation in LBSNs. More research is needed to take advantage of meta-graphs or hyper-graphs for deep learning based recommendation.

We observe that the majority of existing methods process structured information into a data format that is consumable by common neural network architectures (e.g., convolutional or recurrent). An alternative approach to using structured side information in deep learning based recommendation is adapting deep neural networks such that they can directly model structured information. Seminal work has been carried out by Wang et al. (2019b, 2019d), in which graph convolutional networks (Kipf and Welling 2016) have been used to incorporate knowledge graphs for recommendation. We note that graph convolutional networks are a special class of graph networks (Battaglia et al. 2018), which are designed to process structured information as an intrinsic capability. The investigation of graph networks is an active ongoing research topic in the machine learning community and we expect that developments on this topic can nurture future advances in deep learning based recommendation with structured side information.

Crowdsourcing side information for recommendation. Crowdsourcing provides an efficient and cost-effective mean for data collection and augmentation. User feedback that is used as the main input to various recommendation methods, can be viewed as the result of crowdsourced feedback collection where the crowd is the large amount of users in the recommender system. From this perspective, the main forms of user feedback that have been considered in recommender systems are rather restricted: we either consider explicit feedback such as ratings or implicit feedback such as clicks, views, or check-ins. The development of recommendation techniques that incorporate various side information opens up new research directions to leverage crowdsourcing for collecting much more types of data as side information for improving recommendation. In this sense, crowdsourcing has the potential to become an integral component of recommender systems enhanced by side information.

Existing work on the intersection of recommendation and crowdsourcing mainly studies recommendation within crowdsourcing platforms or leverages existing crowdsourced data on the Web for recommendation. For example, Leal et al. (2019) studied the problem of recommending wiki pages – a popular example of crowdsourced knowledge repository – with different publisher profiling strategies. [16] proposed a stream recommendation engine that leverages crowdsourced information for hotel recommendation. They specifically considered crowdsourced data streams contributed by tourists in tourism information sharing platforms such as TripAdvisor, Expedia or Booking.com. In contrast, much less work has been carried out to actively involve users to provide the data that is most beneficial for boosting recommendation. We note that this topic is related to research on interactive, controllable recommender systems (Zhao et al. 2013), where users are explicitly asked to express their preferences. Research on this topic, so far, has been mostly theoretical and limited to small scale user studies. It remains key research questions which type of side information is most suitable to be crowdsourced and how to best leverage crowdsourcing techniques to improve recommendation. To bring forward research in this direction, existing literature in crowdsourcing can be inspirational, for example, (deep) active learning from crowds (Yang et al. 2018; Ostapuk et al. 2019) and gamification (Morschheuser et al. 2016).

Side information for reinforcement & adversarial recommendation. Reinforcement learning is an effective approach to quickly identify items for recomendation in a dynamic Web-based environment where new items are continuously generated (Li et al. 2010; Li et al. 2011), for example, news in a news recommender system. The basic idea is balancing between exploration and exploitation, that is, recommending most relevant items to users to maximize user satisfaction, while collecting user feedback on less relevant items so as to improve recommendation performance in the long run. A major challenge in reinforcement learning based recommendation is the large space of possible actions to choose from, namely, which items to present to the users. Being able to leverage similarities between items for knowledge transfer is, therefore, of key importance to reduce the action space. In this respect, side information can play a critical role for enhancing reinforcement learning based recommenders. We note that the importance of content features of items have been widely recognized in this context. Existing research, however, has not tapped into the rich structure of side information.

Adversarial recommendation is a more recent recommendation technique, where the goal is to improve recommendation performance by leveraging adversarial examples, either by directly sampling from the item pool (Wang et al. 2017e) or by performing perturbations on the embedding parameters of items (He et al. 2018b). In this context, side information has the potential to help better sample the adversarial examples or choose items for parameter perturbations. Research on this topic is still at the infant stage and there are plenty of gaps to be filled.

Side information for cross-domain & package recommendation. Here, we discuss two recommendation scenarios where side information can play an important role to enhance recommendation performance, namely, cross-domain recommendation and package recommendation. Cross-domain recommendation addresses the problem of leveraging data in different domains to generate recommendations (Berkovsky et al. 2007; Fernández-Tobías et al. 2012). Assuming that there is certain overlap of information between a source and a target domain, the main idea underlying this class of recommendation methods is to transfer knowledge from the source domain to the target domain, thus addressing data sparsity or cold start problems in the target domain. In this context, side information that describes the common type information of items in the two domains can be highly valuable for knowledge transfer, for example, a topic taxonomy for both books and movies. Besides, side information about users that is indicative of user preferences in different domains can also bridge the source and the target domains, thus useful to improve cross-domain recommendation.

Package recommendation is relevant for scenarios where a package of items is necessary to be recommended (Adomavicius et al. 2011; Wibowo et al. 2017), for example, a list of POIs for tourism or a basket of products. For this kind of recommendation scenarios, it is important to take into account the relationships between items in the recommended package, for example, the geographical proximity of POIs in the recommended POI list or the complementary relationships among products in the recommended basket. Existing research on individual item recommendation has shown that structured side information can be highly beneficial for identifying relationships among items (Yang et al. 2016a; Sun et al. 2017b), in improving recommendation performance and in providing explanations. Uncovering item relationships encoded in structured side information specifically for package recommendation is, however, an underdeveloped topic, which calls for more attention from the research community.

6 CONCLUSION

This Research Commentary surveyed a considerable amount of state-of-the-art recommendation algorithms with the incorporation of side information from two orthogonal angles: (1) different fundamental methodologies of recommendation, including memory-based methods, latent factor, representation learning and deep learning models; and (2) different representations of side information including structural data (flat features, network features, hierarchical features and knowledge graphs) and non-structural data (text features, image features and video features). In addition, we further discussed the challenges and provided new potential directions in recommendation with side information. By doing so, a comprehensive and systematic survey was delivered to benefit both researchers and practitioners in the area of recommender systems.

7 ACKNOWLEDGEMENTS

This work was partly conducted within the Delta-NTU Corporate Lab for Cyber-Physical Systems with funding support from Delta Electronics Inc. and the National Research Foundation (NRF) Singapore under the Corp Lab@University Scheme, and also supported by the funding awarded to Dr. Jie Zhang by the BMW Tech Office Singapore. We also gratefully acknowledge the support of National Natural Science Foundation of China (Grant No. 71601104, 71601116, 71771141 and 61702084) and the support of the Fundamental Research Funds for the Central Universities in China under Grant No. N181705007.

REFERENCES

[1] Gediminas Adomavicius, Nikos Manouselis, and YoungOk Kwon. 2011. Multi-criteria recommender systems. In *Recommender Systems Handbook*. Boston, MA: Springer, 769–803.

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

[3] Taleb Alashkar, Songyao Jiang, Shuyang Wang, and Yun Fu. 2017. Examples-Rules Guided Deep Neural Network for Makeup Recommendation. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*. Menlo Park, CA: AAAI Press, 941–947.

[4] Jie Bao, Yu Zheng, and Mohamed F Mokbel. 2012. Location-based and preference-aware recommendation using sparse geo-social networking data. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*. New York: ACM Press, 199–208.

[5] Yang Bao, Hui Fang, and Jie Zhang. 2014. Topicmf: Simultaneously exploiting ratings and reviews for recommendation. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence*. Menlo Park, CA: AAAI Press, 2–8.

[6] Oren Barkan and Noam Koenigstein. 2016. Item2vec: neural item embedding for collaborative filtering. In *2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP)*. Washington, DC: IEEE Computing Society Press, 1–6.

[7] Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, Caglar Gulcehre, Francis Song, Andrew Ballard, Justin Gilmer, George Dahl, Ashish Vaswani, Kelsey Allen, Charles Nash, Victoria Langston, Chris Dyer, Nicolas Heess, Daan Wierstra, Pushmeet Kohli, Matt Botvinick, Oriol Vinyals, Yujia Li, and Razvan Pascanu. 2018. Relational inductive biases, deep learning, and graph networks. arXiv preprint arXiv:1806.01261.

[8] Alejandro Bellogín, Iván Cantador, Fernando Díez, Pablo Castells, and Enrique Chavarriaga. 2013. An empirical comparison of social, collaborative filtering, and hybrid recommenders. *ACM Transactions on Intelligent Systems and Technology (TIST)* 4, 1 (2013), 14.

[9] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of Machine Learning Research* 3, Feb (2003), 1137–1155.

[10] Shlomo Berkovsky, Tsvi Kuflik, and Francesco Ricci. 2007. Cross-domain mediation in collaborative filtering. In *International Conference on User Modeling*. Berlin-Heidelberg, Germany: Springer, 355–359.

[11] Preeti Bhargava, Thomas Phan, Jiayu Zhou, and Juhan Lee. 2015. Who, what, when, and where: Multi-dimensional collaborative recommendations using tensor factorization on sparse user-generated data. In *Proceedings of the 24th International Conference on World Wide Web*. New York: ACM Press, 130–140.

[12] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent Dirichlet allocation. *Journal of Machine Learning Research* 3, Jan (2003), 993–1022.

[13] Jesús Bobadilla, Fernando Ortega, Antonio Hernando, and Abraham Gutiérrez. 2013. Recommender systems survey. *Knowledge-Based System* 46 (2013), 109–132.

[14] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems*. South Lake Tahoe, CA, December, 2787–2795.

[15] John S Breese, David Heckerman, and Carl Kadie. 1998. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*. San Francisco: CA, Morgan Kaufmann Publishers Inc., 43–52.

[16] Veloso Bruno, Leal Fatima, Malheiro Benedita, and Carlos Burguillo Juan. 2019. On-line Guest Profiling and Hotel Recommendation. *Electronic Commerce Research and Applications* 34 (2019), 100832.

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

[18] Hongyun Cai, Vincent W Zheng, and Kevin Chen-Chuan Chang. 2018. A comprehensive survey of graph embedding: Problems, techniques, and applications. *IEEE Transactions on Knowledge and Data Engineering* 30, 9 (2018), 1616–1637.

[19] Yixin Cao, Xiang Wang, Xiangnan He, Zikun Hu, and Tat-Seng Chua. 2019. Unifying Knowledge Graph Learning and Recommendation: Towards a Better Understanding of User Preferences. arXiv preprint arXiv:1902.06236.

[20] Rose Catherine and William Cohen. 2016. Personalized recommendations using knowledge graphs: A probabilistic logic programming approach. In *Proceedings of the 11th ACM Conference on Recommender Systems*. New York: ACM Press, 325–332.

[21] Rose Catherine and William Cohen. 2017. Transnets: Learning to transform for recommendation. In *Proceedings of the 12th ACM Conference on Recommender Systems*. New York: ACM Press, 288–296.

[22] Jingyuan Chen, Hanwang Zhang, Xiangnan He, Liqiang Nie, Wei Liu, and Tat-Seng Chua. 2017. Attentive collaborative filtering: Multimedia recommendation with item-and component-level attention. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*. New York: ACM Press, 335–344.

[23] Tianqi Chen, Weinan Zhang, Qiuxia Lu, Kailong Chen, Zhao Zheng, and Yong Yu. 2012. SVDFeature: a toolkit for feature-based collaborative filtering. *Journal of Machine Learning Research* 13, Dec (2012), 3619–3622.

[24] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, Rohan Anil, Zakaria Haque, Lichan Hong, Vihan Jain, Xiaobing Liu, and Hemal Shah. 2016. Wide & deep learning for recommender systems. In *Proceedings of the 11th ACM Conference on Recommender Systems*. New York: ACM Press, 7–10.

[25] Kyunghyun Cho, Bart Van-Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the Properties of Neural Machine Translation: Encoder-Decoder Approaches. arXiv preprint arXiv:1409.1259.

[26] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation.

arXiv preprint arXiv:1406.1078.
[27] Wei-Ta Chu and Ya-Lun Tsai. 2017. A hybrid recommendation system considering visual information for predicting favorite restaurants. *World Wide Web* 20, 6 (2017), 1313–1331.
[28] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research* 12, Aug (2011), 2493–2537.
[29] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *Proceedings of the 11th ACM Conference on Recommender Systems*. New York: ACM Press, 191–198.
[30] James Davidson, Benjamin Liebald, Junning Liu, Palash Nandy, Taylor Van Vleet, Ullas Gargi, Sujoy Gupta, Yu He, Mike Lambert, Blake Livingston, et al. 2010. The YouTube video recommendation system. In *Proceedings of the 5th ACM Conference on Recommender Systems*. New York: ACM Press, 293–296.
[31] Christian Desrosiers and George Karypis. 2011. A comprehensive survey of neighborhood-based recommendation methods. In *Recommender systems handbook*. Boston MA: Springer, 107–144.
[32] Daizong Ding, Mi Zhang, Shao-Yuan Li, Jie Tang, Xiaotie Chen, and Zhi-Hua Zhou. 2017. BayDNN: Friend Recommendation with Bayesian Personalized Ranking Deep Neural Network. In *Proceedings of the 26th ACM International Conference on Information and Knowledge Management*. New York: ACM Press, 1479–1488.
[33] Yang Dingqi, Bingqing Qu, Yang Jie, and Philippe Cudré-Mauroux. 2019. Revisiting user mobility and social relationships in lbsns: a hypergraph embedding approach. In *Proceedings of the 2019 World Wide Web Conference on World Wide Web*. New York: ACM Press, 2147–2157.
[34] Xin Dong, Lei Yu, Zhonghuo Wu, Yuxia Sun, Lingfeng Yuan, and Fangxi Zhang. 2017. A hybrid collaborative filtering model with deep structure for recommender systems. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*. Menlo Park, CA: AAAI Press, 1309–1315.
[35] David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. 2015. Convolutional networks on graphs for learning molecular fingerprints. In *Advances in Neural Information Processing Systems*. Montreal, Canada, 2224–2232.
[36] Michael D Ekstrand, John T Riedl, Joseph A Konstan, et al. 2011. Collaborative filtering recommender systems. *Foundations and Trends® in Human Computer Interaction* 4, 2 (2011), 81–173.
[37] Wenqi Fan, Qing Li, and Min Cheng. 2018. Deep Modeling of Social Relations for Recommendation. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*. Menlo Park, CA: AAAI Press, 8075–8076.
[38] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. 2019. Graph Neural Networks for Social Recommendation. arXiv preprint arXiv:1902.07243.
[39] Hui Fang, Yang Bao, and Jie Zhang. 2014. Leveraging decomposed trust in probabilistic matrix factorization for effective recommendation. In *Proceedings of the 28st AAAI Conference on Artificial Intelligence*, Vol. 350. Menlo Park, CA: AAAI Press, 30–36.
[40] Hui Fang, Guibing Guo, and Jie Zhang. 2015. Multi-faceted trust and distrust prediction for recommender systems. *Decision Support Systems* 71 (2015), 37–47.
[41] Jie Feng, Yong Li, Chao Zhang, Funing Sun, Fanchao Meng, Ang Guo, and Depeng Jin. 2018. DeepMove: Predicting Human Mobility with Attentional Recurrent Networks. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web*. New York: ACM Press, 1459–1468.
[42] Shanshan Feng, Gao Cong, Bo An, and Yeow Meng Chee. 2017. POI2Vec: Geographical Latent Representation for Predicting Future Visitors. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*. Menlo Park, CA: AAAI Press, 102–108.
[43] Ignacio Fernández-Tobías, Iván Cantador, Marius Kaminskas, and Francesco Ricci. 2012. Cross-domain recommender systems: A survey of the state of the art. In *Spanish Conference on Information Retrieval*. 1–12.
[44] Rana Forsati, Mehrdad Mahdavi, Mehrnoush Shamsfard, and Mohamed Sarwat. 2014. Matrix factorization with explicit trust and distrust side information for improved social recommendation. *ACM Transactions on Information Systems (TOIS)* 32, 4 (2014), 17.
[45] Huiji Gao, Jiliang Tang, Xia Hu, and Huan Liu. 2015. Content-Aware Point of Interest Recommendation on Location-Based Social Networks. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*. Menlo Park, CA: AAAI Press, 1721–1727.
[46] Carlos A Gomez-Uribe and Neil Hunt. 2016. The netflix recommender system: Algorithms, business value, and innovation. *ACM Transactions on Management Information Systems (TMIS)* 6, 4 (2016), 13.
[47] László Grad-Gyenge, Peter Filzmoser, and Hannes Werthner. 2015. Recommendations on a knowledge graph. In *1st International Workshop on Machine Learning Methods for Recommender Systems (MLRec)*. Vancouver, Canada, April 30-May 2, 13–20.
[48] Mihajlo Grbovic, Vladan Radosavljevic, Nemanja Djuric, Narayan Bhamidipati, Jaikit Savla, Varun Bhagwan, and Doug Sharp. 2015. E-commerce in your inbox: Product recommendations at scale. In *Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York: ACM Press, 1809–1818.
[49] Aditya Grover and Jure Leskovec. 2016. Node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York: ACM Press, 855–864.
[50] Guibing Guo. 2012. Resolving data sparsity and cold start in recommender systems. In *International Conference on User Modeling, Adaptation, and Personalization*. Berlin-Heidelberg, Germany: Springer, 361–364.
[51] Guibing Guo. 2013. Integrating trust and similarity to ameliorate the data sparsity and cold start for recommender systems. In *Proceedings of the 7th ACM Conference on Recommender Systems*. New York: ACM Press, 451–454.
[52] Guibing Guo, Jie Zhang, and Daniel Thalmann. 2012. A simple but effective method to incorporate trusted neighbors in recommender systems. In *International Conference on User Modeling, Adaptation, and Personalization*. Berlin-Heidelberg, Germany: Springer, 114–125.
[53] Guibing Guo, Jie Zhang, and Daniel Thalmann. 2014. Merging trust in collaborative filtering to alleviate data sparsity and cold start. *Knowledge-Based Systems* 57 (2014), 57–68.
[54] Guibing Guo, Jie Zhang, and Neil Yorke-Smith. 2015. Leveraging multiviews of trust and similarity to enhance clustering-based recommender systems. *Knowledge-Based Systems* 74 (2015), 14–27.
[55] Guibing Guo, Jie Zhang, and Neil Yorke-Smith. 2015. TrustSVD: Collaborative Filtering with Both the Explicit and Implicit Influence of User Trust and of Item Ratings. In *International Joint Conference on Artificial Intelligence*, Vol. 15. Menlo Park: AAAI Press, 123–125.
[56] Qing Guo, Zhu Sun, and Yin-Leng Theng. 2019. Exploiting side information for recommendation. In *International Conference on Web Engineering*. Berlin-Heidelberg, Germany: Springer, 569–573.
[57] Qing Guo, Zhu Sun, Jie Zhang, Qi Chen, and Yin-Leng Theng. 2017. Aspect-aware point-of-interest recommendation with geo-social influence. In *International Conference on User Modeling, Adaptation, and Personalization*. New York: ACM Press, 17–22.
[58] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Washington, DC: IEEE Computer Society Press, 770–778.
[59] Ruining He, Chen Fang, Zhaowen Wang, and Julian McAuley. 2016. Vista: A visually, socially, and temporally-aware model for artistic recommendation. In *Proceedings of the 11th ACM Conference on Recommender Systems*. New York: ACM Press, 309–316.
[60] Ruining He, Chunbin Lin, Jianguo Wang, and Julian McAuley. 2016. Sherlock: sparse hierarchical embeddings for visually-aware one-class collaborative filtering. In *International Joint Conference on Artificial Intelligence*. Menlo Park: AAAI Press, 3740–3746.
[61] Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proceedings of the 2016 World Wide Web Conference on World Wide Web*. New York: ACM Press, 507–517.
[62] Ruining He and Julian McAuley. 2016. VBPR: Visual Bayesian Personalized Ranking from Implicit Feedback. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*. Menlo Park, CA: AAAI Press, 144–150.
[63] Xiangnan He, Tao Chen, Min-Yen Kan, and Xiao Chen. 2015. Trirank: Review-aware explainable recommendation by modeling aspects. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. New York: ACM Press, 1661–1670.
[64] Xiangnan He, Xiaoyu Du, Xiang Wang, Feng Tian, Jinhui Tang, and Tat-Seng Chua. 2018. Outer product-based neural collaborative filtering. arXiv preprint arXiv:1808.03912.
[65] Xiangnan He, Zhankui He, Xiaoyu Du, and Tat-Seng Chua. 2018. Adversarial personalized ranking for recommendation. In *Proceedings of the 41st International ACM SIGIR conference on Research and Development in Information Retrieval*. New York: ACM Press, 355–364.
[66] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 2017 World Wide Web Conference on World Wide Web*. New York: ACM Press, 173–182.
[67] John R Hershey and Peder A Olsen. 2007. Approximating the Kullback Leibler divergence between Gaussian mixture models. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing*, Vol. 4. Los Alamitos, CA: IEEE Computer Society Press, IV–317.
[68] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2015. Session-based recommendations with recurrent neural networks. arXiv preprint arXiv:1511.06939.
[69] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. 2006. A fast learning algorithm for deep belief nets. *Neural Computation* 18, 7 (2006), 1527–1554.
[70] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation* 9, 8 (1997), 1735–1780.
[71] Seyedabbas Hosseini, Ali Khodadadi, Keivan Alizadeh, Ali Arabzadeh, Mehrdad Farajtabar, Hongyuan Zha, and Hamid RR Rabiee. 2018. Recurrent poisson

factorization for temporal recommendation. IEEE Transactions on Knowledge and Data Engineering. in press.

[72] Binbin Hu, Chuan Shi, Wayne Xin Zhao, and Philip S Yu. 2018. Leveraging meta-path based context for top-n recommendation with a neural co-attention model. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York: ACM Press, 1531–1540.

[73] Longke Hu, Aixin Sun, and Yong Liu. 2014. Your neighbors affect your ratings: on geographical neighborhood influence to rating prediction. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*. New York: ACM Press, 345–354.

[74] Zhiting Hu, Poyao Huang, Yuntian Deng, Yingkai Gao, and Eric Xing. 2015. Entity hierarchy embedding. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (1, Long Papers)*, Vol. 1. Menlo Park: AAAI Press, 1292–1300.

[75] Won-Seok Hwang, Ho-Jong Lee, Sang-Wook Kim, and Minsoo Lee. 2012. On using category experts for improving the performance and accuracy in recommender systems. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*. New York: ACM Press, 2355–2358.

[76] Mohsen Jamali and Martin Ester. 2010. A matrix factorization technique with trust propagation for recommendation in social networks. In *Proceedings of the 5th ACM Conference on Recommender Systems*. New York: ACM Press, 135–142.

[77] Rodolphe Jenatton, Julien Mairal, Guillaume Obozinski, and Francis R Bach. 2010. Proximal Methods for Sparse Hierarchical Dictionary Learning. In *International Conference on Machine Learning*. Haifa, Israel: JMLR.org, 487–494.

[78] Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Knowledge graph embedding via dynamic mapping matrix. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (1, Long Papers)*, Vol. 1. 687–696.

[79] Ke Ji, Hong Shen, Hui Tian, Yanbo Wu, and Jun Wu. 2014. Two-phase layered learning recommendation via category structure. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Berlin-Heidelberg, Germany: Springer, 13–24.

[80] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. 2014. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM International Conference on Multimedia*. New York: ACM Press, 675–678.

[81] How Jing and Alexander J Smola. 2017. Neural survival recommender. In *Proceedings of the 11th ACM International Conference on Web Search and Data Mining*. New York: ACM Press, 515–524.

[82] Bhargav Kanagal, Amr Ahmed, Sandeep Pandey, Vanja Josifovski, Jeff Yuan, and Lluis Garcia-Pueyo. 2012. Supercharging recommender systems using taxonomies for learning user purchase behavior. *Proceedings of the VLDB Endowment* 5, 10 (2012), 956–967.

[83] Wang-Cheng Kang and Julian McAuley. 2018. Self-Attentive Sequential Recommendation. In *2018 IEEE International Conference on Data Mining*. Washington, DC: IEEE Computer Society Press, 197–206.

[84] Alexandros Karatzoglou, Xavier Amatriain, Linas Baltrunas, and Nuria Oliver. 2010. Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering. In *Proceedings of the 4th ACM conference on Recommender systems*. New York: ACM Press, 79–86.

[85] Choonho Kim and Juntae Kim. 2003. A recommendation algorithm using multi-level association rules. In *Proceedings IEEE/WIC International Conference on Web Intelligence*. Washington, DC: IEEE Computer Society Press, 524–527.

[86] Seyoung Kim and Eric P Xing. 2010. Tree-guided group lasso for multi-task regression with structured sparsity. In *International Conference on Machine Learning*. Haifa, Israel: JMLR.org, 543–550.

[87] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907.

[88] Noam Koenigstein, Gideon Dror, and Yehuda Koren. 2011. Yahoo! music recommendations: modeling music ratings with temporal dynamics and item taxonomy. In *Proceedings of the 5th ACM Conference on Recommender Systems*. New York: ACM Press, 165–172.

[89] Yehuda Koren. 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York: ACM Press, 426–434.

[90] Yehuda Koren. 2009. Collaborative filtering with temporal dynamics. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York: ACM Press, 447–456.

[91] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 8 (2009), 30–37.

[92] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*. Harrahs and Harveys, Lake Tahoe, 1097–1105.

[93] Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International Conference on Machine Learning*. Beijing China: JMLR.org, 1188–1196.

[94] Fátima Leal, Bruno M Veloso, Benedita Malheiro, Horacio González-Vélez, and Juan Carlos Burguillo. 2019. Scalable modelling and recommendation using wiki-based crowdsourced repositories. *Electronic Commerce Research and Applications* 33 (2019), 100817.

[95] Yann LeCun and Yoshua Bengio. 1995. Convolutional networks for images, speech, and time series. Cambridge, MA: MIT Press.

[96] Chenyi Lei, Dong Liu, Weiping Li, Zheng-Jun Zha, and Houqiang Li. 2016. Comparative deep learning of hybrid representations for image recommendations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Washington, DC: IEEE Computer Society Press, 2545–2553.

[97] Huayu Li, Yong Ge, Richang Hong, and Hengshu Zhu. 2016. Point-of-interest recommendations: Learning potential check-ins from friends. In *Proceedings of the 22th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York: ACM Press, 975–984.

[98] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. 2010. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 2010 World Wide Web Conference on World Wide Web*. New York: ACM Press, 661–670.

[99] Lihong Li, Wei Chu, John Langford, and Xuanhui Wang. 2011. Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. In *Proceedings of the 4th ACM International Conference on Web Search and Data Mining*. New York: ACM Press, 297–306.

[100] Piji Li, Zihao Wang, Zhaochun Ren, Lidong Bing, and Wai Lam. 2017. Neural rating regression with abstractive tips generation for recommendation. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York: ACM Press, 345–354.

[101] Defu Lian, Cong Zhao, Xing Xie, Guangzhong Sun, Enhong Chen, and Yong Rui. 2014. GeoMF: joint geographical modeling and matrix factorization for point-of-interest recommendation. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York: ACM Press, 831–840.

[102] Dawen Liang, Jaan Altosaar, Laurent Charlin, and David M Blei. 2016. Factorization meets the item embedding: Regularizing matrix factorization with item co-occurrence. In *Proceedings of the 11th ACM Conference on Recommender Systems*. New York: ACM Press, 59–66.

[103] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, Vol. 15. Menlo Park, CA: AAAI Press, 2181–2187.

[104] Greg Linden, Brent Smith, and Jeremy York. 2003. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing* 1, 1 (2003), 76–80.

[105] Christoph Lippert, Stefan Hagen Weber, Yi Huang, Volker Tresp, Matthias Schubert, and Hans-Peter Kriegel. 2008. Relation prediction in multi-relational domains using matrix factorization. In *Proceedings of the NIPS 2008 Workshop: Structured Input-Structured Output, Vancouver, Canada*. 1–4.

[106] Qiang Liu, Shu Wu, and Liang Wang. 2017. DeepStyle: Learning user preferences for visual recommendation. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*. New York: ACM Press, 841–844.

[107] Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. 2016. Predicting the Next Location: A Recurrent Model with Spatial and Temporal Contexts. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*. Menlo Park, CA: AAAI Press, 194–200.

[108] Xin Liu, Yong Liu, Karl Aberer, and Chunyan Miao. 2013. Personalized point-of-interest recommendation by mining users' preference transition. In *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management*. New York: ACM Press, 733–738.

[109] Xin Liu, Yong Liu, and Xiaoli Li. 2016. Exploring the Context of Locations for Personalized Location Recommendations. In *International Joint Conference on Artificial Intelligence*. Menlo Park: AAAI Press, 1188–1194.

[110] Pasquale Lops, Marco De Gemmis, and Giovanni Semeraro. 2011. Content-based recommender systems: State of the art and trends. In *Recommender Systems Handbook*. Boston MA: Springer, 73–105.

[111] Yichao Lu, Ruihai Dong, and Barry Smyth. 2018. Why I like it: multi-task learning for recommendation and explanation. In *Proceedings of the 13th ACM Conference on Recommender Systems*. New York: ACM Press, 4–12.

[112] Chen Luo, Wei Pang, Zhe Wang, and Chenghua Lin. 2014. Hete-cf: Social-based collaborative filtering recommendation using heterogeneous relations. In *2014 IEEE International Conference on Data Mining*. Washington, DC: IEEE Computer Society Press, 917–922.

[113] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. arXiv preprint arXiv:1508.04025.

[114] Hao Ma. 2013. An experimental study on implicit social recommendation. In *Proceedings of the 36th International ACM SIGIR conference on Research and Development in Information Retrieval*. New York: ACM Press, 73–82.

[115] Hao Ma, Irwin King, and Michael R Lyu. 2009. Learning to recommend with social trust ensemble. In *Proceedings of the 32nd International ACM SIGIR conference on Research and Development in Information Retrieval*. New York: ACM Press, 203–210.

[116] Hao Ma, Irwin King, and Michael R Lyu. 2011. Learning to recommend with explicit and implicit social relations. *ACM Transactions on Intelligent Systems and Technology (TIST)* 2, 3 (2011), 29.

[117] Hao Ma, Michael R Lyu, and Irwin King. 2009. Learning to recommend with trust and distrust relationships. In *Proceedings of the 4th ACM Conference on Recommender Systems*. New York: ACM Press, 189–196.

[118] Hao Ma, Haixuan Yang, Michael R Lyu, and Irwin King. 2008. Sorec: social recommendation using probabilistic matrix factorization. In *Proceedings of the 17th ACM International Conference on Information and Knowledge Management*. New York: ACM Press, 931–940.

[119] Hao Ma, Dengyong Zhou, Chao Liu, Michael R Lyu, and Irwin King. 2011. Recommender systems with social regularization. In *Proceedings of the 4th ACM International Conference on Web Search and Data Mining*. New York: ACM Press, 287–296.

[120] Jonathan Masci, Ueli Meier, Dan Cireşan, and Jürgen Schmidhuber. 2011. Stacked convolutional auto-encoders for hierarchical feature extraction. In *International Conference on Artificial Neural Networks*. Berlin-Heidelberg, Germany: Springer, 52–59.

[121] Julian McAuley and Jure Leskovec. 2013. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 8th ACM Conference on Recommender Systems*. New York: ACM Press, 165–172.

[122] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th International ACM SIGIR conference on Research and Development in Information Retrieval*. New York: ACM Press, 43–52.

[123] Prem Melville, Raymond J Mooney, and Ramadass Nagarajan. 2002. Content-boosted collaborative filtering for improved recommendations. *Proceedings of the 16th AAAI Conference on Artificial Intelligence* 23 (2002), 187–192.

[124] Aditya Krishna Menon, Krishna-Prasad Chitrapura, Sachin Garg, Deepak Agarwal, and Nagaraj Kota. 2011. Response prediction using collaborative filtering with hierarchies and side-information. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York: ACM Press, 141–149.

[125] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

[126] Andriy Mnih. 2011. Taxonomy-informed latent factor models for implicit feedback. In *Proceedings of the 2011 International Conference on KDD Cup 2011, 18*. JMLR.org, 169–181.

[127] Andriy Mnih and Ruslan R Salakhutdinov. 2008. Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems*. Vancouver, B.C., Canada, 1257–1264.

[128] Andriy Mnih and Yee W Teh. 2012. Learning label trees for probabilistic modelling of implicit feedback. In *Advances in Neural Information Processing Systems*. Harrahs and Harveys, Lake Tahoe, 2816–2824.

[129] Benedikt Morschheuser, Juho Hamari, and Jonna Koivisto. 2016. Gamification in crowdsourcing: a review. In *The 49th Hawaii International Conference on System Sciences (HICSS)*. Washington, DC: IEEE Computer Society Press, 4375–4384.

[130] Vinod Nair and Geoffrey E Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *International Conference on Machine Learning*. Haifa, Israel: JMLR.org, 807–814.

[131] Mathias Niepert, Mohamed Ahmed, and Konstantin Kutzkov. 2016. Learning convolutional neural networks for graphs. In *International Conference on Machine Learning*. New York City, NY: JMLR.org, 2014–2023.

[132] Wei Niu, James Caverlee, and Haokai Lu. 2018. Neural Personalized Ranking for Image Recommendation. In *Proceedings of the 11th ACM International Conference on Web Search and Data Mining*. New York: ACM Press, 423–431.

[133] Shumpei Okura, Yukihiro Tagami, Shingo Ono, and Akira Tajima. 2017. Embedding-based news recommendation for millions of users. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York: ACM Press, 1933–1942.

[134] Natalia Ostapuk, Jie Yang, and Philippe Cudre-Mauroux. 2019. ActiveLink: Deep Active Learning for Link Prediction in Knowledge Graphs. In *Proceedings of the 2019 World Wide Web Conference on World Wide Web*. New York: ACM Press, 1398–1408.

[135] Nikolaos Pappas and Andrei Popescu-Belis. 2013. Sentiment analysis of user comments for one-class collaborative filtering over ted talks. In *Proceedings of the 36th International ACM SIGIR conference on Research and Development in Information Retrieval*. New York: ACM Press, 773–776.

[136] Seung-Taek Park, David Pennock, Omid Madani, Nathan Good, and Dennis DeCoste. 2006. Naïve filterbots for robust cold-start recommendations. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York: ACM Press, 699–705.

[137] Rajiv Pasricha and Julian McAuley. 2018. Translation-based factorization machines for sequential recommendation. In *Proceedings of the 13th ACM Conference on Recommender Systems*. New York: ACM Press, 63–71.

[138] Arkadiusz Paterek. 2007. Improving regularized singular value decomposition for collaborative filtering. In *Proceedings of KDD Cup and Workshop*, Vol. 2007. New York: ACM Press, 5–8.

[139] Wenjie Pei, Tadas Baltrusaitis, David MJ Tax, and Louis-Philippe Morency. 2017. Temporal attention-gated model for robust sequence classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Washington, DC: IEEE Computer Society Press, 6730–6739.

[140] Wenjie Pei, Jie Yang, Zhu Sun, Jie Zhang, Alessandro Bozzon, and David MJ Tax. 2017. Interacting attention-gated recurrent networks for recommendation. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. New York: ACM Press, 1459–1468.

[141] Štefan Pero and Tomáš Horváth. 2013. Opinion-driven matrix factorization for rating prediction. In *International Conference on User Modeling, Adaptation, and Personalization*. Berlin-Heidelberg, Germany: Springer, 1–13.

[142] Fatemeh Pourgholamali, Mohsen Kahani, Ebrahim Bagheri, and Zeinab Noorian. 2017. Embedding unstructured side information in product recommendation. *Electronic Commerce Research and Applications* 25 (2017), 70–85.

[143] Juan Ramos et al. 2003. Using tf-idf to determine word relevance in document queries. In *Proceedings of the 1st Instructional Conference on Machine Learning*, Vol. 242. Piscataway, NJ, 133–142.

[144] Steffen Rendle. 2010. Factorization machines. In *2010 IEEE International Conference on Data Mining*. Washington, DC: IEEE Computer Society Press, 995–1000.

[145] Steffen Rendle. 2012. Factorization machines with libfm. *ACM Transactions on Intelligent Systems and Technology (TIST)* 3, 3 (2012), 57.

[146] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*. Montreal, Canada: AUAI Press, 452–461.

[147] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 2010 World Wide Web Conference on World Wide Web*. New York: ACM Press, 811–820.

[148] Francesco Ricci, Lior Rokach, and Bracha Shapira. 2015. Recommender systems: introduction and challenges. In *Recommender Systems Handbook*. Boston MA: Springer, 1–34.

[149] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. 1985. *Learning internal representations by error propagation*. Technical Report. La Jolla Institute for Cognitive Science, University of California, San Diego.

[150] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 2001 World Wide Web Conference on World Wide Web*. New York: ACM Press, 285–295.

[151] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2008. The graph neural network model. *IEEE Transactions on Neural Networks* 20, 1 (2008), 61–80.

[152] J Ben Schafer, Dan Frankowski, Jon Herlocker, and Shilad Sen. 2007. Collaborative filtering recommender systems. In *The Adaptive Web*. Berlin-Heidelberg, Germany: Springer, 291–324.

[153] Jürgen Schmidhuber. 2015. Deep learning in neural networks: An overview. *Neural Networks* 61 (2015), 85–117.

[154] Suvash Sedhain, Aditya Krishna Menon, Scott Sanner, and Lexing Xie. 2015. Autorec: Autoencoders meet collaborative filtering. In *Proceedings of the 2015 World Wide Web Conference on World Wide Web*. New York: ACM Press, 111–112.

[155] Sungyong Seo, Jing Huang, Hao Yang, and Yan Liu. 2017. Interpretable convolutional neural networks with dual local and global attention for review rating prediction. In *Proceedings of the 12th ACM Conference on Recommender Systems*. New York: ACM Press, 297–305.

[156] Mohak Sharma, P Krishna Reddy, R Uday Kiran, and Thirumalaisamy Ragunathan. 2011. Improving the performance of recommender system by exploiting the categories of products. In *International Workshop on Databases in Networked Information Systems*. Berlin-Heidelberg, Germany: Springer, 137–146.

[157] Chuan Shi, Jian Liu, Fuzhen Zhuang, S Yu Philip, and Bin Wu. 2016. Integrating heterogeneous information via flexible regularization framework for recommendation. *Knowledge and Information Systems* 49, 3 (2016), 835–859.

[158] Chuan Shi, Zhiqiang Zhang, Ping Luo, Philip S Yu, Yading Yue, and Bin Wu. 2015. Semantic path based personalized recommendation on weighted heterogeneous information networks. In *Proceedings of the 24th ACM International Conference on Information and Knowledge Management*. New York: ACM Press, 453–462.

[159] Yue Shi, Martha Larson, and Alan Hanjalic. 2011. Tags as bridges between domains: Improving recommendation with tag-induced cross-domain collaborative filtering. In *International Conference on User Modeling, Adaptation, and Personalization*. Berlin-Heidelberg, Germany: Springer, 305–316.

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

[161] Ajit P Singh and Geoffrey J Gordon. 2008. Relational learning via collective matrix factorization. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York: ACM Press, 650–658.

[162] Alexander J Smola and Risi Kondor. 2003. Kernels and regularization on graphs. In *Learning Theory and Kernel Machines*. Berlin-Heidelberg, Germany: Springer, 144–158.

[163] Richard Socher, Cliff C Lin, Chris Manning, and Andrew Y Ng. 2011. Parsing natural scenes and natural language with recursive neural networks. In *International Conference on Machine Learning*. Bellevue, Washington: JMLR.org, 129–136.

[164] Yading Song, Simon Dixon, and Marcus Pearce. 2012. A survey of music recommendation systems and future perspectives. In *9th International Symposium on Computer Music Modeling and Retrieval*, Vol. 4. London, UK, 395–410.

[165] Xiaoyuan Su and Taghi M Khoshgoftaar. 2009. A survey of collaborative filtering techniques. *Advances in Artificial Intelligence* 2009, 1 (2009), 395–410.

[166] Yizhou Sun, Jiawei Han, Xifeng Yan, Philip S Yu, and Tianyi Wu. 2011. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. *Proceedings of the VLDB Endowment* 4, 11 (2011), 992–1003.

[167] Zhu Sun. 2015. Exploiting Item and User Relationships for Recommender Systems. In *International Conference on User Modeling, Adaptation, and Personalization*. Berlin-Heidelberg, Germany: Springer, 397–402.

[168] Zhu Sun, Guibing Guo, and Jie Zhang. 2016. Effective recommendation with category hierarchy. In *Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization*. New York: ACM Press, 299–300.

[169] Zhu Sun, Guibing Guo, Jie Zhang, and Chi Xu. 2017. A Unified Latent Factor Model for Effective Category-Aware Recommendation. In *International Conference on User Modeling, Adaptation, and Personalization*. New York: ACM Press, 389–390.

[170] Zhu Sun, Jie Yang, Jie Zhang, and Alessandro Bozzon. 2017. Exploiting both vertical and horizontal dimensions of feature hierarchy for effective recommendation. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*. Menlo Park, CA: AAAI Press, 189–195.

[171] Zhu Sun, Jie Yang, Jie Zhang, Alessandro Bozzon, Yu Chen, and Chi Xu. 2017. MRLR: Multi-level Representation Learning for Personalized Ranking in Recommendation. In *International Joint Conference on Artificial Intelligence*. Menlo Park: AAAI Press, 2807–2813.

[172] Zhu Sun, Jie Yang, Jie Zhang, Alessandro Bozzon, Long-Kai Huang, and Chi Xu. 2018. Recurrent Knowledge Graph Embedding for Effective Recommendation. In *Proceedings of the 13th ACM Conference on Recommender Systems*. 297–305.

[173] Panagiotis Symeonidis, Alexandros Nanopoulos, and Yannis Manolopoulos. 2010. A unified framework for providing recommendations in social tagging systems based on ternary semantic analysis. *IEEE Transactions on Knowledge and Data Engineering* 22, 2 (2010), 179–192.

[174] Jiaxi Tang and Ke Wang. 2018. Personalized top-n sequential recommendation via convolutional sequence embedding. In *Proceedings of the 11th ACM International Conference on Web Search and Data Mining*. New York: ACM Press, 565–573.

[175] Yi Tay, Anh Tuan Luu, and Siu Cheung Hui. 2018. Multi-pointer co-attention networks for recommendation. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. New York: ACM Press, 2309–2318.

[176] Maria Terzi, Matthew Rowe, Maria-Angela Ferrario, and Jon Whittle. 2014. Text-based user-knn: Measuring user similarity based on text reviews. In *International Conference on User Modeling, Adaptation, and Personalization*. Berlin-Heidelberg, Germany: Springer, 195–206.

[177] Trinh Xuan Tuan and Tu Minh Phuong. 2017. 3D convolutional networks for session-based recommendation with content features. In *Proceedings of the 12th ACM Conference on Recommender Systems*. New York: ACM Press, 138–146.

[178] Flavian Vasile, Elena Smirnova, and Alexis Conneau. 2016. Meta-prod2vec: Product embeddings using side-information for recommendation. In *Proceedings of the 11th ACM Conference on Recommender Systems*. New York: ACM Press, 225–232.

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

[180] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and On the Properties of Neural Machine Translation: Encoder-Decoder Approaches Bengio. 2017. Graph attention networks. arXiv preprint arXiv:1710.10903.

[181] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. 2008. Extracting and composing robust features with denoising autoencoders. In *International Conference on Machine Learning*. Helsinki, Finland: JMLR.org, 1096–1103.

[182] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. 2010. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research* 11, Dec (2010), 3371–3408.

[183] Hao Wang, Yanmei Fu, Qinyong Wang, Hongzhi Yin, Changying Du, and Hui Xiong. 2017. A location-sentiment-aware recommender system for both home-town and out-of-town users. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York: ACM Press, 1135–1143.

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

[185] Hongwei Wang, Fuzheng Zhang, Jialin Wang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. 2018. RippleNet: Propagating user preferences on the knowledge graph for recommender systems. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. New York: ACM Press, 417–426.

[186] Hongwei Wang, Fuzheng Zhang, Jialin Wang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. 2019a. Exploring High-Order User Preference on the Knowledge Graph for Recommender Systems. *ACM Transactions on Information Systems (TOIS)* 37, 3 (2019a), 32.

[187] Hongwei Wang, Fuzheng Zhang, Xing Xie, and Minyi Guo. 2018. DKN: Deep knowledge-aware network for news recommendation. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web*. New York: ACM Press, 1835–1844.

[188] Hongwei Wang, Fuzheng Zhang, Mengdi Zhang, Jure Leskovec, Miao Zhao, Wenjie Li, and Zhongyuan Wang. 2019. Knowledge Graph Convolutional Networks for Recommender Systems with Label Smoothness Regularization. arXiv preprint arXiv:1905.04413.

[189] Hongwei Wang, Fuzheng Zhang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. 2019. Multi-Task Feature Learning for Knowledge Graph Enhanced Recommendation. arXiv preprint arXiv:1901.08907.

[190] Hongwei Wang, Miao Zhao, Xing Xie, Wenjie Li, and Minyi Guo. 2019. Knowledge graph convolutional networks for recommender systems. arXiv preprint arXiv:1904.12575.

[191] Jun Wang, Lantao Yu, Weinan Zhang, Yu Gong, Yinghui Xu, Benyou Wang, Peng Zhang, and Dell Zhang. 2017. Irgan: A minimax game for unifying generative and discriminative information retrieval models. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*. New York: ACM Press, 515–524.

[192] Pengfei Wang, Jiafeng Guo, Yanyan Lan, Jun Xu, Shengxian Wan, and Xueqi Cheng. 2015. Learning hierarchical representation model for nextbasket recommendation. In *Proceedings of the 38th International ACM SIGIR conference on Research and Development in Information Retrieval*. New York: ACM Press, 403–412.

[193] Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. 2017. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering* 29, 12 (2017), 2724–2743.

[194] Suhang Wang, Jiliang Tang, Yilin Wang, and Huan Liu. 2015. Exploring Implicit Hierarchical Structures for Recommender Systems. In *International Joint Conference on Artificial Intelligence*. Menlo Park: AAAI Press, 1813–1819.

[195] Suhang Wang, Yilin Wang, Jiliang Tang, Kai Shu, Suhas Ranganath, and Huan Liu. 2017. What your images reveal: Exploiting visual contents for point-of-interest recommendation. In *Proceedings of the 2017 World Wide Web Conference on World Wide Web*. New York: ACM Press, 391–400.

[196] Wenya Wang, Sinno Jialin Pan, Daniel Dahlmeier, and Xiaokui Xiao. 2016. Recursive neural conditional random fields for aspect-based sentiment analysis. arXiv preprint arXiv:1603.06679.

[197] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019. KGAT: Knowledge Graph Attention Network for Recommendation. arXiv preprint arXiv:1905.07854.

[198] Xiang Wang, Dingxian Wang, Canran Xu, Xiangnan He, Yixin Cao, and Tat-Seng Chua. 2018. Explainable Reasoning over Knowledge Graphs for Recommendation. arXiv preprint arXiv:1811.04540.

[199] Yueyang Wang, Yuanfang Xia, Siliang Tang, Fei Wu, and Yueting Zhuang. 2017. Flickr group recommendation with auxiliary information in heterogeneous information networks. *Multimedia Systems* 23, 6 (2017), 703–712.

[200] Li-Tung Weng, Yue Xu, Yuefeng Li, and Richi Nayak. 2008. Exploiting item taxonomy for solving cold-start problem in recommendation making. In *2008 20th IEEE International Conference on Tools with Artificial Intelligence*, Vol. 2. Washington, DC: IEEE Computer Society Press, 113–120.

[201] Agung Toto Wibowo, Advaith Siddharthan, Chenghua Lin, and Judith Masthoff. 2017. Matrix Factorization for Package Recommendations. In *Proceedings of the RecSys 2017 Workshop on Recommendation in Complex Scenarios*. New York: ACM Press, 1–5.

[202] Chao-Yuan Wu, Amr Ahmed, Alex Beutel, and Alexander J Smola. 2017. Joint training of ratings and reviews with recurrent recommender networks. In *Workshop on International Conference on Learning Representations*. Toulon, France, 4–12.

[203] Chao-Yuan Wu, Amr Ahmed, Alex Beutel, Alexander J Smola, and How Jing. 2017. Recurrent recommender networks. In *Proceedings of the 10th ACM International Conference on Web Search and Data Mining*. New York: ACM Press, 495–503.

[204] Xin Xin, Xiangnan He, Yongfeng Zhang, Yongdong Zhang, and Joemon Jose. 2019. Relational Collaborative Filtering: Modeling Multiple Item Relations for Recommendation. arXiv preprint arXiv:1904.12796.

[205] Liang Xiong, Xi Chen, Tzu-Kuo Huang, Jeff Schneider, and Jaime G Carbonell. 2010. Temporal collaborative filtering with bayesian probabilistic tensor factorization. In *Proceedings of the 2010 Society for Industrial and Applied Mathematics International Conference on Data Mining*. Columbus, Ohio: SIAM, 211–222.

[206] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, and Misc Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. arXiv preprint arXiv:1502.03044.

[207] Xiaoying Xu, Kaushik Dutta, and Chunmian Ge. 2018. Do adjective features from user reviews address sparsity and transparency in recommender systems? *Electronic Commerce Research and Applications* 29 (2018), 113–123.

[208] Hong-Jian Xue, Xinyu Dai, Jianbing Zhang, Shujian Huang, and Jiajun Chen. 2017. Deep Matrix Factorization Models for Recommender Systems. In *International Joint Conference on Artificial Intelligence*. Menlo Park: AAAI Press, 3203–3209.

[209] Shuicheng Yan, Dong Xu, Benyu Zhang, Hong-Jiang Zhang, Qiang Yang, and Stephen Lin. 2007. Graph embedding and extensions: A general framework for dimensionality reduction. *IEEE Transactions on Pattern Analysis & Machine Intelligence* 1, 1 (2007), 40–51.

[210] Bo Yang, Yu Lei, Dayou Liu, and Jiming Liu. 2013. Social Collaborative Filtering by Trust. In *International Joint Conference on Artificial Intelligence*. Menlo Park: AAAI Press, 1633–1647.

[211] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2014. Embedding entities and relations for learning and inference in knowledge bases. arXiv preprint arXiv:1412.6575.

[212] Carl Yang, Lanxiao Bai, Chao Zhang, Quan Yuan, and Jiawei Han. 2017. Bridging collaborative filtering and semi-supervised learning: a neural approach for poi recommendation. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York: ACM Press, 1245–1254.

[213] Dingqi Yang, Daqing Zhang, Zhiyong Yu, and Zhu Wang. 2013. A sentiment-enhanced personalized location recommendation system. In *Proceedings of the 24th ACM Conference on Hypertext and Social Media*. New York: ACM Press, 119–128.

[214] Jie Yang, Thomas Drake, Andreas Damianou, and Yoelle Maarek. 2018. Leveraging crowdsourcing data for deep active learning an application: Learning intents in alexa. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web*. New York: ACM Press, 23–32.

[215] Jie Yang, Zhu Sun, Alessandro Bozzon, and Jie Zhang. 2016. Learning hierarchical feature influence for recommendation by recursive regularization. In *Proceedings of the 11th ACM Conference on Recommender Systems*. New York: ACM Press, 51–58.

[216] Xiwang Yang, Harald Steck, and Yong Liu. 2012. Circle-based recommendation in online social networks. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York: ACM Press, 1267–1275.

[217] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 1480–1489.

[218] Di Yao, Chao Zhang, Jianhui Huang, and Jingping Bi. 2017. SERM: A recurrent model for next location prediction in semantic trajectories. In *Proceedings of the 26th ACM International Conference on Information and Knowledge Management*. New York: ACM Press, 2411–2414.

[219] Mao Ye, Peifeng Yin, and Wang-Chien Lee. 2010. Location recommendation for location-based social networks. In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*. New York: ACM Press, 458–461.

[220] Hongzhi Yin, Yizhou Sun, Bin Cui, Zhiting Hu, and Ling Chen. 2013. LCARS: a location-content-aware recommender system. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York: ACM Press, 221–229.

[221] Hongzhi Yin, Weiqing Wang, Hao Wang, Ling Chen, and Xiaofang Zhou. 2017. Spatial-aware hierarchical collaborative deep learning for POI recommendation. *IEEE Transactions on Knowledge and Data Engineering* 29, 11 (2017), 2537–2551.

[222] Feng Yu, Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. 2016. A dynamic recurrent model for next basket recommendation. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. New York: ACM Press, 729–732.

[223] Wenhui Yu, Huidi Zhang, Xiangnan He, Xu Chen, Li Xiong, and Zheng Qin. 2018. Aesthetic-based clothing recommendation. In *Proceedings of the 2017 World Wide Web Conference on World Wide Web*. New York: ACM Press, 649–658.

[224] Xiao Yu, Xiang Ren, Quanquan Gu, Yizhou Sun, and Jiawei Han. 2013. Collaborative filtering with entity similarity regularization in heterogeneous information networks. *The 1st International Joint Conference on Artificial Intelligence Workshop on Heterogeneous Information Network Analysis* 27 (2013).

[225] Xiao Yu, Xiang Ren, Yizhou Sun, Quanquan Gu, Bradley Sturt, Urvashi Khandelwal, Brandon Norick, and Jiawei Han. 2014. Personalized entity recommendation: A heterogeneous information network approach. In *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*. New York: ACM Press, 283–292.

[226] Xiao Yu, Xiang Ren, Yizhou Sun, Bradley Sturt, Urvashi Khandelwal, Quanquan Gu, Brandon Norick, and Jiawei Han. 2013. Recommendation in heterogeneous information networks with implicit user feedback. In *Proceedings of the 8th ACM Conference on Recommender Systems*. New York: ACM Press, 347–350.

[227] Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. 2016. Collaborative knowledge base embedding for recommender systems. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York: ACM Press, 353–362.

[228] Jia-Dong Zhang and Chi-Yin Chow. 2015. GeoSoCa: Exploiting geographical, social and categorical correlations for point-of-interest recommendations. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York: ACM Press, 443–452.

[229] Shuai Zhang, Yi Tay, Lina Yao, and Aixin Sun. 2018. Next Item Recommendation with Self-Attention. arXiv preprint arXiv:1808.06414.

[230] Sheng Zhang, Weihong Wang, James Ford, and Fillia Makedon. 2006. Learning from incomplete ratings using non-negative matrix factorization. In *Proceedings of the 2006 SIAM International Conference on Data Mining*. SIAM, 549–553.

[231] Shuai Zhang, Lina Yao, and Aixin Sun. 2017. Deep learning based recommender system: A survey and new perspectives. arXiv preprint arXiv:1707.07435.

[232] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2019. Deep learning based recommender system: A survey and new perspectives. *ACM Computing Surveys (CSUR)* 52, 1 (2019), 5.

[233] Yongfeng Zhang, Qingyao Ai, Xu Chen, and W Bruce Croft. 2017. Joint representation learning for top-n recommendation with heterogeneous information sources. In *Proceedings of the 26th ACM International Conference on Information and Knowledge Management*. New York: ACM Press, 1449–1458.

[234] Yongfeng Zhang, Guokun Lai, Min Zhang, Yi Zhang, Yiqun Liu, and Shaoping Ma. 2014. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In *Proceedings of the 37th International ACM SIGIR conference on Research and Development in Information Retrieval*. New York: ACM Press, 83–92.

[235] Yongfeng Zhang, Haochen Zhang, Min Zhang, Yiqun Liu, and Shaoping Ma. 2014. Do users rate or review?: Boost phrase-level sentiment labeling with review-level sentiment classification. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*. New York: ACM Press, 1027–1030.

[236] Zhiqian Zhang, Chenliang Li, Zhiyong Wu, Aixin Sun, Dengpan Ye, and Xiangyang Luo. 2017. Next: a neural network framework for next POI recommendation. arXiv preprint arXiv:1704.04576.

[237] Xiaoxue Zhao, Weinan Zhang, and Jun Wang. 2013. Interactive collaborative filtering. In *Proceedings of the 22nd ACM International Conference on Information and Knowledge Management*. New York: ACM Press, 1411–1420.

[238] Zhi-Lin Zhao, Chang-Dong Wang, Yuan-Yu Wan, and Jian-Huang Lai. 2017. Recommendation in feature space sphere. *Electronic Commerce Research and Applications* 26 (2017), 109–118.

[239] Jing Zheng, Jian Liu, Chuan Shi, Fuzhen Zhuang, Jingzhi Li, and Bin Wu. 2017. Recommendation in heterogeneous information network via dual similarity regularization. *International Journal of Data Science and Analytics* 3, 1 (2017), 35–48.

[240] Lei Zheng, Vahid Noroozi, and Philip S Yu. 2017. Joint deep modeling of users and items using reviews for recommendation. In *Proceedings of the 10th ACM International Conference on Web Search and Data Mining*. New York: ACM Press, 425–434.

[241] Jiang Zhou, Rami Albatal, and Cathal Gurrin. 2016. Applying visual user interest profiles for recommendation and personalisation. In *International Conference on Multimedia Modeling*. Berlin-Heidelberg, Germany: Springer, 361–366.

[242] Cai-Nicolas Ziegler, Georg Lausen, and Lars Schmidt-Thieme. 2004. Taxonomy-driven computation of product recommendations. In *Proceedings of the 13th ACM International Conference on Information & Knowledge Management*. New York: ACM Press, 406–415.