Graph Learning based Recommender Systems: A Review

Shoujin Wang[1] , **Liang Hu**[2,3] , **Yan Wang**[1∗] , **Xiangnan He**[4] , **Quan Z. Sheng**[1] , **Mehmet A. Orgun**[1] , **Longbing Cao**[5] , **Francesco Ricci**[6] and **Philip S. Yu**[7]
[1] Macquarie University,
[2] DeepBlue Academy of Sciences,
[3] Tongji University,
[4]University of Science and Technology of China,
[5]University of Technology Sydney,
[6]Free University of Bozen-Bolzano,
[7]University of Illinois at Chicago

Abstract

Recent years have witnessed the fast development of the emerging topic of Graph Learning based Recommender Systems (GLRS). GLRS employ advanced graph learning approaches to model users' preferences and intentions as well as items' characteristics for recommendations. Differently from other RS approaches, including content-based filtering and collaborative filtering, GLRS are built on graphs where the important objects, e.g., users, items, and attributes, are either explicitly or implicitly connected. With the rapid development of graph learning techniques, exploring and exploiting homogeneous or heterogeneous relations in graphs are a promising direction for building more effective RS. In this paper, we provide a systematic review of GLRS, by discussing how they extract important knowledge from graph-based representations to improve the accuracy, reliability and explainability of the recommendations. First, we characterize and formalize GLRS, and then summarize and categorize the key challenges and main progress in this novel research area.

1 Introduction

Recommender Systems (RS) are one of the most popular and important applications of Artificial Intelligence (AI). They have been widely adopted to help the users of many popular content sharing and e-Commerce web sites to more easily find relevant content, products or services. Meanwhile, Graph Learning (GL), which relates to machine learning applied to graph structure data, is an emerging technique of AI which is rapidly developing and has shown its great capability in recent years [Wu *et al.*, 2021]. In fact, by benefiting from these capabilities to learn relational data, an emerging RS paradigm built on GL, i.e., Graph Learning based Recommender Systems (GLRS), has been proposed and studied extensively in the last few years [Guo *et al.*, 2020]. In this paper we offer a systematic review of the challenges and progresses in this emerging area.

∗Corresponding author, email: yan.wang@mq.edu.au.

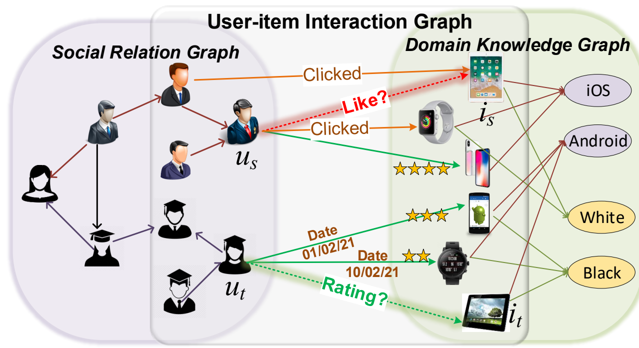

Figure 1: The demonstration of graph learning based recommender systems

Motivation: why graph learning for RS?

Most of the data in RS has essentially a graph structure. In the real world, most of the objects around us are explicitly or implicitly connected with each other; in other words, we are living in a world of graphs. Such characteristic is even more obvious in RS where the objects here considered including users, items, attributes, context, are tightly connected with each other and influence each other via various relations [Hu *et al.*, 2014], as shown in Figure 1. In practice, various kinds of graphs arise from the data used by RS, and they can significantly contribute to the quality of the recommendations.

Graph learning has the capability to learn complex relations. As one of the most promising machine learning techniques, GL has shown great potential in deriving knowledge embedded in different kinds of graphs. Specifically, many GL techniques, such as random walk and graph neural networks, have been developed to learn the particular type of relations modeled by graphs, and have demonstrated to be quite effective [Wu *et al.*, 2021]. Consequently, employing GL to model various relations in RS is a natural and compelling choice.

Formalization: how does graph learning can help RS?

To date, there is no unified formalization of GLRS. We generally formalize GLRS from a high-level perspective.

We construct a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ with the data of an RS where the objects, e.g., users and items, are represented as nodes in $\mathcal{V}$ and the relations between them, e.g., purchases, are represented as edges in $\mathcal{E}$. Then, a GLRS model $M(\Theta)$ is constructed and trained to generate optimal recommendation results $\mathcal{R}$ with optimized model parameters Θ that are learned from the topological and content information of $\mathcal{G}$. Formally,

$$\mathcal{R} = \arg\max_{\Theta} f(M(\Theta)|\mathcal{G}). \quad (1)$$

Depending on the specific recommendation data and scenarios, the graph $\mathcal{G}$ and the recommendation target $\mathcal{R}$ can be defined in various forms, e.g., $\mathcal{G}$ can be homogeneous sequences or heterogeneous networks while $\mathcal{R}$ can be predicted ratings or ranking over items. The objective function f can be the maximum utility [Wang *et al.*, 2019f] or the maximum probability to form links between nodes [Verma *et al.*, 2019].

Contributions. The main contributions of this work are summarized below:

- We systematically analyze the key challenges presented by various GLRS graphs and categorize them from a data driven perspective, providing a useful view to better understand the important characteristics of GLRS.
- We summarize the current research progress in GLRS by systematically categorizing the more technical state-of-the-art literature.
- We share and discuss some open research directions of GLRS for giving references to the community.

2 Data Characteristics and Challenges

Different objects are managed by an RS, e.g., users, items, attributes. All of them are inter-connected with various types of relations [Hu *et al.*, 2019], e.g., social relations between users, or interactions between users and items. This results in different types of graphs that may be considered in an RS. In this section, we first classify the various types of data used in RS by considering their source and characteristics. For each class, we analyze its characteristics, then discuss how to better represent it with graphs, and finally indicate challenges that these characteristics pose to building GLRS. A brief summary of these data types is provided in Table 1.

It is well known that the three key objects managed by an RS are *user*, *item* and *user-item interaction* (interaction for short), and thus all the data managed by an RS is related to them. There are two broad types of data: *user-item interaction data*, e.g., clicks and ratings of users for items, and *side information data*, e.g., users' profiles and items' attributes [Shi *et al.*, 2014]. Depending on whether the temporal order of the interactions is recorded or not, interaction data can be classified into sequential interaction data and general interaction data. Hence, we classify the data of an RS into three classes: (1) *general interaction data*, (2) *sequential interaction data*, and (3) *side information data*. Each class can be further divided into multiple sub-classes (cf. Table 1).

2.1 GLRS Built on General Interaction Data

Interactions between users and items are usually represented as an interaction matrix, where each row indicates one user and each column indicates one item. Each entry in the matrix captures an information about the type of occurred interaction. Depending on the interaction type, the interaction data can be divided into the explicit (i.e., users' ratings on items) and the implicit (e.g., click, view) [Zhang *et al.*, 2019b]. Then, the recommendation task based on this general interaction data is usually formulated as a matrix completion task [Zhang and Chen, 2020].

An interaction matrix can be naturally represented as a *user-item bipartite graph* [Zha *et al.*, 2001]. In this graph, the user nodes and the item nodes constitute the two "parts" respectively, while the interactions are represented as edges connecting two nodes in different parts. Furthermore, an explicit interaction matrix can be represented as a *weighted bipartite graph* where each edge is labeled with a weight to indicate the rating value. An implicit interaction matrix can be represented as an *unweighted bipartite graph* where en edge indicates a implicit interaction. Hence, from this graph based perspective, the recommendation task is converted to link prediction on the RS bipartite graph [Li and Chen, 2013].

The advantage of building a GLRS on a bipartite graph is obvious. Since most users often interacted with only a small proportion of the large amount of items, matrix completion methods generally face data sparsity related and cold-start problems [Jamali and Ester, 2009]. A bipartite graph based approach mitigates these issues by enabling the information propagating widely among nodes to enrich the information of those users and items with less interactions [Wu *et al.*, 2020b]. However, it is challenging *how to effectively and efficiently propagate the information between users or items.* This is particularly challenging in a bipartite graph since no direct links exist between users or items, and thus the information should be propagated via multi-hop neighbour nodes. For instance, to propagate some information from user u_1 to a similar user u_2 one needs to first propagate it to a bridge item v_1 connecting both users, and then to u_2 from v_1.

While targeting this challenge, a variety of GLRS approaches have been developed. For weighted bipartite graphs, there are mainly graph auto-encoder (e.g., graph convolutional matrix completion [Berg *et al.*, 2017]), Graph Convolutional Networks (GCN) (e.g., multi-graph convolutional neural networks [Monti *et al.*, 2017], stacked and reconstructed GCN [Zhang *et al.*, 2019a]), and Graph SAmple and aggreGatE (GraphSage) (e.g., inductive graph-based matrix completion [Zhang and Chen, 2020]). For unweighted bipartite graphs, there are mainly random walk (e.g., RecWalk [Nikolakopoulos and Karypis, 2019]), graph embedding (e.g., high-order proximity for implicit recommendation [Yang *et al.*, 2018], collaborative similarity embedding [Chen *et al.*, 2019]), GCN (e.g., spectral collaborative filtering [Zheng *et al.*, 2018], lightGCN [He *et al.*, 2020], low-pass collaborative filter [Yu and Qin, 2020], multi-behavior GCN [Jin *et al.*, 2020]), and GraphSage (e.g., neural graph collaborative filtering [Zheng *et al.*, 2018]). The gist of these approaches will be discussed in Section 3.

2.2 GLRS Built on Sequential Interaction Data

A sequential interaction data set is a collection of sequences of user-item interactions (e.g., click, purchase) registered dur-

Data Class	Data Subclass	Representing Graph	Representative Approach Category
General interaction	Explicit interaction	Weighted bipartite graph	Graph auto-encoder[1], GCN[2], GraphSage[3]
	Implicit interaction	Unweighted bipartite graph	Random walk[4], graph embedding[5], GCN[6], GraphSage[7]
Sequential interaction	Single-type interactions	Directed homogeneous graph	GGNN[8], GraphSage[9], GAT[10]
	Multi-type interactions	Directed heterogeneous graph	GraphSage[11]
Side information	Attribute information	Heterogeneous graph	Graph embedding[12], GAT[13]
	Social information	Homogeneous graph	Random walk[14], graph embedding[15], GAT[16]
	External knowledge	Tree or heterogeneous graph	Graph embedding[17], GCN[18]

[1][Berg *et al.*, 2017];[2][Monti *et al.*, 2017];[3][Zhang and Chen, 2020];[4][Nikolakopoulos and Karypis, 2019];[5][Chen *et al.*, 2019];[6][He *et al.*, 2020];[7][Zheng *et al.*, 2018];[8][Wu *et al.*, 2019b];[9][Ma *et al.*, 2020] [10][Qiu *et al.*, 2019];[11][Wang *et al.*, 2020c];[12][Shi *et al.*, 2018];[13][Wang *et al.*, 2019f];[14][Jamali and Ester, 2009];[15][Wen *et al.*, 2018];[16][Fan *et al.*, 2019];[17][Gao *et al.*, 2019];[18][Wang *et al.*, 2019b]

Table 1: A summary of data in RS, the representing graph, and the corresponding GLRS approach

ing a given time period, and ordered by their time stamp. According to the number of interaction types included in a sequence, a sequential interaction data set can be divided into *single-type interaction data set* where only one type of interactions is included, and *multi-type interaction data set* where multiple types of interactions are included. Multi-type interactions like view, click and purchase co-happening in one sequence are very common in practice [Wang *et al.*, 2020c]. For a given user u, a single-type interaction sequence is usually recorded as a sequence of interacted with items (denoted as v), e.g., $\{v_1, ..., v_n\}$, while a multi-type interaction sequence is recorded as a sequence of $\langle interaction\ type, item\rangle$ pairs, e.g., $\{click\ v_1, click\ v_2, ..., purchase\ v_n\}$. An RS built on sequential interaction data is formalized as a Sequential Recommender System (SRS) which takes a sequence of historical interactions as input to predict the possible next interaction(s) [Quadrana *et al.*, 2018; Wang *et al.*, 2019e].

A sequential interaction data set can be represented as a directed graph where each interaction sequence corresponds to one path in the graph [Wu *et al.*, 2019b]. In each path, the interactions serve as the nodes and a directed edge between any adjacent nodes indicates the order of interactions. In a multi-type interaction sequence, each element is a $\langle interaction\ type, item\rangle$ pair, which results in a compound node composed of two parts. Note that in some cases one user may have multiple identical interactions happening in a sequence (e.g., click the same items multiple times), resulting in a path consisting of one or more loops [Wu *et al.*, 2019b].

The advantages of building SRS on directed graph lies in the strong capability of graph learning to represent and model even the most complicated transitions in a sequence of interactions. There are usually complicated transitions which deviate from simple one-way consecutive time series patterns [Wang *et al.*, 2020a] over sequential interactions, especially when there are multiple identical interactions in one sequence [Wu *et al.*, 2019b; Wang *et al.*, 2020b]. Such transitions can be well represented by the multi-direction connections in a graph and well learned by the information aggregation from neighbour nodes of different directions in graph learning [Wu *et al.*, 2019b; Xu *et al.*, 2019]. However, building SRS on a directed graph is still challenging. In particular it is critical *how to construct a graph to effectively represent the sequential interaction data with minimal information loss, and how to propagate information on the graph to effectively model even the most complicated transitions.*

While targeting these challenges, various SRS have been built based on graph learning. Most of the studied approaches focus on single-type interaction data, including Gated Graph Neural Networks (GGNN) (e.g., session-based recommendation with GNN [Wu *et al.*, 2019b] and graph contextualized self-attention networks [Xu *et al.*, 2019]), GraphSage (e.g., memory augmented GNN [Ma *et al.*, 2020]), and Graph ATtention networks (GAT) (e.g., full graph neural network [Qiu *et al.*, 2019]). Limited approaches for multi-type interaction data include GraphSage (e.g., multi-relational GNN for session-based prediction [Wang *et al.*, 2020c]).

2.3 GLRS Incorporating Side Information Data

Interaction data is often sparse [Hu *et al.*, 2019], thus is not sufficient for correctly capturing the users' preferences and item characteristics. Hence, various types of side information, e.g., attribute information and social information, have been used to alleviate such an issue. In this section, we discuss three main types of side information: (1) *attribute information*, (2) *social information*, and (3) *external knowledge*.

GLRS Incorporating Attribute Information

Attribute information mainly includes user attributes (e.g., gender, age), and item attributes (e.g., category, price) [Wang *et al.*, 2017; Han *et al.*, 2018]. A user (item) attribute data set is usually recorded as a user (item) information table where each row indicates one user (item) and each column is one attribute. Attribute information is often combined with general or sequential interaction data to perform recommendations. Given a data set, the combination of interaction data and attribute data naturally results in a *heterogeneous graph*. In such a graph, three types of nodes, i.e., user node, item node and attribute value node, and at least two types of edges exist. Specifically, in the combination of general interaction data and attribute data, in addition to user-item edges (cf. Sec. 2.1), there are user (or item)-attribute value edge representing the relations between user (or item) and attributes. In the combination of sequential interaction data and attribute data, in addition to the directed interaction-interaction edges (cf. Sec. 2.2), there are also item-attribute value edges. Consequently, the recommendation task here becomes the prediction of the interactions by learning the complex relations embedded in the above mentioned heterogeneous graph.

Heterogeneous graphs combine two different types of information, i.e., interaction information and attribute information, hence enabling information propagation among different types of nodes, and better coping with the mentioned data sparsity problem. However, it is challenging *to selectively aggregate those useful attribute information to improve the recommendation performance.*

GLRS targeting such a challenge include (heterogeneous) graph embedding (e.g., entity2rec [Palumbo *et al.*, 2017] based on node2vec, heterogeneous preference embedding [Chen *et al.*, 2016] and heterogeneous network embedding for recommendation [Shi *et al.*, 2018]), and GAT (e.g., knowledge graph attention network [Wang *et al.*, 2019f]).

GLRS Incorporating Social Information

Social information relates to the commonly existing social relations between users. A particular type of social relation among the users in a data set naturally forms a homogeneous social graph where each user corresponds to a node and each social link (e.g., friend relation) between two users corresponds to an edge. In an RS, the social graph can be mainly used for two tasks: (1) *social recommendation* (recommending items to users by incorporating social information) [Fan *et al.*, 2019], and (2) *friend recommendation* (recommending users to a given user by predicting the possible social links) [Huang *et al.*, 2015].

Social recommendation. Social relations enable social influence diffusion among users [Wu *et al.*, 2020a] and thus help better understand users' preferences. The combination of social information and general or sequential user-item interaction data naturally results in a heterogeneous graph comprising two parts. The first is the bipartite graph derived from the general interaction data (cf. Sec. 2.1) or the directed graph extracted from the sequential interaction data (cf. Sec. 2.2), while the second part is the social graph connecting the users. Obviously, two heterogeneous types of information (i.e., interaction information and social information) are contained in the graph. Hence, the RS must be able to effectively leverage this heterogeneous graph to predict the unknown user-item interactions. Such an approach helps better understand a user's preference by considering the influence of her neighbours in a social graph. However, on one hand, it is not clear how many orders of neighbours should be considered to correctly compute this influence on a given user. On the other hand, different neighbours usually influence a user to different degrees [Wu *et al.*, 2020b]. Hence, it is a challenge *to appropriately model the influence of other users to a given user.* Typical approaches targeting this challenge include random walk (e.g., Trust-walker [Jamali and Ester, 2009]), graph embedding [Wen *et al.*, 2018] and GAT (e.g., GraphRec [Fan *et al.*, 2019] and improved diffusion network [Wu *et al.*, 2020a]). All these works focus on combining social graph and general interactions, while limited works combine social graph and sequential interactions [Song *et al.*, 2019].

Friend recommendation. By using the aforementioned homogeneous social graph, friend recommendation is performed as a link prediction task on such graph [Yin *et al.*, 2010]. Specifically, given a target user and the known social links on the graph, friend recommendation first infers the possible links between other users and the target user and then recommends those users with high probabilities to link with the target user to her. The main challenge lies in *how to appropriately model the mutual-influence between users.* According to our analysis, random walk based approaches [Backstrom and Leskovec, 2011; Bagci and Karagoz, 2016] are more common in order to address this challenge. Other approaches include graph embedding [Verma *et al.*, 2019].

GLRS Incorporating External Knowledge

External knowledge, e.g., item taxonomy and semantic relations between concepts, related to users and items usually contributes to a deeper understanding of the users' preference and item characteristics [Wang *et al.*, 2018a], and ultimately improving recommendation performance. Such knowledge is usually represented as a knowledge graph where various types of objects (e.g., users, movies, movie directors) are represented as nodes and the relations between them (e.g., movie-director relation) are represented as edges [Wang *et al.*, 2019f]. This graph is often combined with the graph composed by the general or sequential interaction data, giving rise to a more complex and heterogeneous graph. There are mainly two types of external knowledge commonly utilized in RS: item/user ontology and common knowledge.

GLRS incorporating ontology knowledge. The ontology of users or items is usually represented as a hierarchical tree-like graph where the hierarchical relations between users or items are recorded. A type of commonly utilized ontology knowledge for recommendations is item taxonomy information [Huang *et al.*, 2019]. An example of such a tree graph is used in Amazon.com, where the category information of products is used to organize all the items offered by the platform. In that graph, the root node corresponds to the coarsest-grained category and the leaf nodes represent specific items. The incorporation of item ontology knowledge enables a better understanding of the users' multi-level preferences towards items, and thus helps improving the explainability of the recommendations [Gao *et al.*, 2019]. However, it remains a challenge *to propagate users' preferences over items along the hierarchy tree graph to extract the multi-level preferences.* Representative works targeting such a challenge include graph embedding based approaches [Wang *et al.*, 2018b; Gao *et al.*, 2019], aimed at learning more informative item embedding for general recommendations, and memory network on graphs to learn coarse-grained-preference representation for sequential recommendations [Huang *et al.*, 2019].

GLRS incorporating common knowledge. Common knowledge refers to the wide range of relations between the various entities managed by an RS. It includes, but is not limited to, general semantic relations between entities (e.g., the relations among bread, food, bakery item from Microsoft Concept Graph[1]) [Sheu and Li, 2020], and domain-specific relations between entities (e.g., the relations between movies, directors, genre) [Gao *et al.*, 2020]. Due to the diversity of these entities and their relations, common knowledge is usually represented as a heterogeneous and complex graph where different types of nodes and edges exist [Guo *et al.*, 2020]. The incorporation of common knowledge benefits the exploration and exploitation of various external implicit

[1]https://concept.research.microsoft.com/

relations between users and/or items, improving recommendation performance. However, it remains a challenge *to effectively propagate information between different types of entities via different types of links between them*, to obtain coherent and useful information for the recommendations. Representative works targeting this challenge include graph embedding methods [Wang *et al.*, 2019c] (especially meta-path based embedding [Zhao *et al.*, 2017; Sun *et al.*, 2018; Shi *et al.*, 2018; Wang *et al.*, 2019g]) to wisely learn the embedding of heterogeneous entities and relations, and GNN based methods (especially GCN [Wang *et al.*, 2019b] an GAT [Wang *et al.*, 2019f]) to iteratively aggregate the information from neighbour nodes.

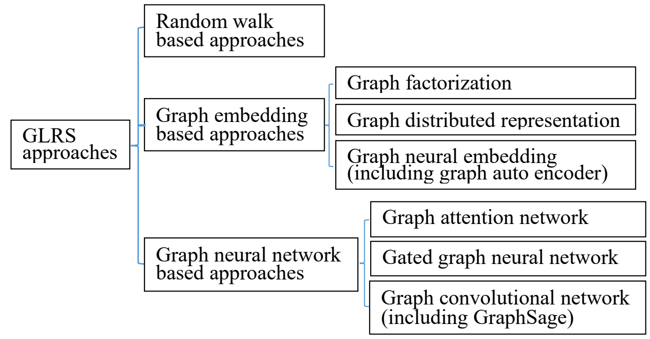

Figure 2: Classifying GLRS approaches from technical perspective

3 Graph Learning Approaches for RS

In this section, we introduce graph learning based techniques, which offer solutions to the challenges faced by GLRS, which were discussed in Section 2. We first provide a technical categorization of the solutions, and then we discuss the gist of each solution together with the achieved progresses.

The categorization of the approaches to GLRS is presented in Figure 2. GLRS are divided into three categories, and some categories are further divided into sub-categories.

3.1 Random Walk based Approach

Random walk based RS have been extensively studied in the past years and have been widely employed on various types of graphs (e.g., social graphs, sequence graphs). Generally, a random walk based RS first let a random walker to walk on a given graph with a predefined transition probability for each step, in order to model the implicit preference or interaction propagation among users and/or items, and then takes the probability the walker lands on nodes after certain steps to rank these candidate nodes for recommendations. Random walk based RS are particularly suitable for capturing the complex, higher-order and indirect relations among various types of nodes (e.g., users and items) on the graph, and thus, can address important challenges for GLRS especially those built on heterogeneous graphs.

There are different variants of random walk based RS. Besides the basic random walk based RS [Baluja *et al.*, 2008], random walk with restart based RS [Bagci and Karagoz, 2016; Jiang *et al.*, 2018] is a representative type of several variants. It sets a constant probability to jump back to the starting node in each transition and it is generally used in graphs containing many nodes to avoid leaving the particular context of the starting node.

Although widely applied, the drawbacks of random walk based RS are clear: (1) they need to generate ranking scores on all candidate items at each step for each user, leading to low efficiency; (2) unlike most of the learning-based paradigms, they are heuristic-based, lacking model parameters to optimize the recommendation objective.

3.2 Graph Embedding based Approach

Graph embedding is an effective technique to analyze the complex relations embedded on graphs and has been rapidly developing in recent years. It maps each node into a low-dimension embedding vector which encodes the graph structure information. Researchers introduced graph embedding to model the complex relations between various nodes (e.g., users, items) and they came up with the novel approach of Graph Embedding based RS (GERS). Depending to the specific embedding approach that is used, GERS can be divided into three classes: (1) Graph Factorization based RS (GFRS), (2) Graph Distributed Representation based RS (GDRRS), and (3) Graph Neural Embedding based RS (GNERS).

Graph Factorization based RS (GFRS). GFRS first factorizes the inter-node commuting matrix based on meta-path on the graph in order to obtain the embedding of each node (e.g., a user or an item), which are then used as input of the subsequent recommendation task [Wang *et al.*, 2019h]. By doing so, the complex relations between nodes in the graph are encoded into the embedding to improve the recommendations. Due to their capability to handle the heterogeneity of the nodes, GFRS have been widely applied to capture relations between different types of nodes, e.g., users and items. However, although being simple and effective, such models may easily suffer from the sparsity of the observed data.

Graph Distributed Representation based RS (GDRRS). Differently from GFRS, GDRRS usually follow Skip-gram model [Mikolov *et al.*, 2013] in order to learn a distributed representation of each user or item in a graph. They encode information about the user or item and its adjacent relations into a low-dimensional vector [Shi *et al.*, 2018], which is then used for the subsequent recommendation step. Specifically, GDRRS usually first use random walk to generate a sequence of nodes that co-occurred in one meta-path and then employ the skip-gram or similar models to generate node representations for recommendations. By exploiting its powerful capability to encode the inter-node connections in a graph, GDRRS are widely applied to both homogeneous and heterogeneous graphs for capturing the relations between the objects managed by the RS [Cen *et al.*, 2019]. GDRRS have shown their great potential in recent years due to their simplicity, efficiency and efficacy.

Graph Neural Embedding based RS (GNERS). GNERS utilize neural networks, like Multilayer-perceptron, auto-encoder, to learn users or items embedding. Neural embedding models are easy to integrated with other downstream neural recommendation models (e.g., RNN based ones) to build an end-to-end RS [Han *et al.*, 2018]. To this end, GNERS have been widely applied to a variety of graphs like attributed graphs [Han *et al.*, 2018], interaction combined with knowledge graphs [Hu *et al.*, 2018; Cen *et al.*, 2019].

3.3 Graph Neural Network based Approach

Graph Neural Networks (GNN) apply neural networks techniques on graph data. Leveraging the strength of GNN in learning informative representations, several RS have used GNN to address the most important challenges posed by GLRS. By considering a model perspective, GNN based RS can be mainly categorized into three classes: (1) Graph ATtention network based RS (GATRS), (2) Gated Graph Neural Network based RS (GGNNRS), and (3) Graph Convolutional Network (including GraphSage) based RS (GCNRS).

Graph ATtention network based RS (GATRS). Graph ATtention networks (GAT) introduce attention mechanisms into GNN to discriminatively learn the different relevance and influence degree of other users (items) w.r.t. the target user (item) on a given graph. GATRS are based on GAT for precisely learning inter-user or item relations. In such a case, the influence of the more important users or items, w.r.t. a specific user or item, is emphasized, which is more in line with the real-world cases and this has been shown to be beneficial for the recommendations. Due to their good discrimination capability, GAT are widely used in different kinds of graphs including social graphs [Fan *et al.*, 2019], item session graphs [Xu *et al.*, 2019], and knowledge graphs [Wang *et al.*, 2019f].

Gated Graph Neural Network based RS (GGNNRS). Gated graph neural networks (GGNN) introduce the Gated Recurrent Unit (GRU) into GNN to learn the optimized node representations by iteratively absorbing the influence of other nodes in a graph to comprehensively capture the inter-node relations. GGNNRS are built on GGNN to learn the user or item embeddings for recommendations by comprehensively considering the complex inter-user or inter-item relations. Due to their capability to capture the complex relations between nodes, GGNN are widely used to model the complex transitions between items in a sequence graph for sequential recommendations [Wu *et al.*, 2019b], or to model the complex interactions between different categories of fashion products for fashion recommendations [Cui *et al.*, 2019], and they have achieved superior recommendation performance.

Graph Convolutional Network based RS (GCNRS). Graph Convolutional Networks (GCN) generally learn how to iteratively aggregate feature information from local graph neighbor nodes by leveraging both graph structure and node feature information. In general, by utilizing the convolution and pooling operations, GCNs are capable of learning informative embeddings of users and items by effectively aggregating information from their neighborhoods in graphs. GCNRS are built on GCN to learn the user or item embeddings in a graph while exploiting both the complex relations between users or/and items and their own content information for recommendations [Ying *et al.*, 2018]. Thanks to the powerful feature extraction and learning capability, particularly their strength in combining the graph structure and node content information, GCN are widely applied to a variety of graphs in RS to build GCNRS and are demonstrated to be very effective. For instance, GCN are used for influence diffusion on social graphs in social recommendations [Wu *et al.*, 2019a], for mining the hidden user-item connection information on user-item interaction graphs, for alleviating the data sparsity problemn in collaborative filtering [Wang *et al.*, 2019a], and for capturing inter-item relatedness by mining their associated attributes on knowledge graphs [Wang *et al.*, 2019b].

4 GLRS Algorithms and Datasets

The source code of most of the representative GLRS algorithms is publicly accessible. In Table 2, to facilitate the access for empirical analysis, we summarize source codes of algorithms for GLRS which take various input data and use different learning approaches for different learning tasks. The listed algorithms are carefully selected and are commonly used as baselines in existing work.

In addition to algorithms, datasets are another important part for empirical analysis of GLRS approaches. In order to facilitate the empirical analysis of the surveyed algorithms, in Table 3 we also list public and real-world datasets with different characteristics from various domains. These datasets are commonly used for evaluating GLRS algorithms.

5 Open Research Directions

GLRS are fast developing. Although substantial results have been achieved, some challenges still remain. By matching the demonstrated challenges to the research progress already achieved, we have identified some open research directions.

Self-evolutionary RS with dynamic-graph learning. In real-world RS, users, items and the interactions between them, keep evolving over time [Wang *et al.*, 2019d]. This originates graphs with dynamic topology, and such dynamics could have direct impacts on the user and requirement modeling, causing even a clear change of recommendation results over time. However, this issue is still underestimated in existing GLRS. Therefore, it is a promising future research direction to design self-evolutionary RS over dynamic graphs.

Explainable RS with causal graph learning. Causal inference is a major technique used to discover the causal relations between objects or actions. Although some progress has been achieved in explainable RS, we are is still far away from achieving a complete understanding of the reasons and intents behind user choice behaviours, which is a critical step to make reliable and explainable recommendations [Zhang and Chen, 2018]. To this end, it is another promising direction to construct explainable RS with causal graph learning.

Cross-domain RS with multiplex graph learning. In reality, the data and interactions for recommendation could be derived from multiple domains, including various sources, systems, and modalities [Zhu *et al.*, 2019]. These are intercorrelated and must collaboratively contribute to the recommendations [Zhu *et al.*, 2021]. Consequently, the interactions in cross-domain RS can be represented by multiplex networks where nodes may or may not be interconnected with other nodes in other layers. As a result, the new generation cross-domain RS potentially works with multiplex graph learning.

Algorithm	Input Data	Learning Task	Learning Approach	Venue	Link
GC-MC[1]	Explicit interaction	Rating prediction	Graph auto-encoder	KDD'2018 DL	https://github.com/riannevdberg/gc-mc
MGCNN[2]	Explicit interaction	Rating prediction	GCN	NIPS'2017	https://github.com/fmonti/mgcnn
IGMC[3]	Explicit interaction	Rating prediction	GraphSage	ICLR'2020	https://github.com/muhanzhang/IGMC
RecWalk[4]	Implicit interaction	Click prediction	Random walk	WSDM '2019	https://github.com/nikolakopoulos/RecWalk
PinSage[5]	Implicit interaction	Click prediction	GraphSage	KDD'2018	https://github.com/yoonjong12/pinsage
CSE[6]	Implicit interaction	Click prediction	Graph embedding	WWW'2019	https://github.com/cnclabs/proNet-core
LightGCN[7]	Implicit interaction	Click prediction	GCN	SIGIR'2020	https://github.com/kuandeng/LightGCN
SpectralCF[8]	Implicit interaction	Click prediction	GraphSage	RecSys'2018	https://github.com/lzheng21/SpectralCF
SR-GNN[9]	Single-type sequential interaction	Next-item prediction	GGNN	AAAI'2019	https://github.com/CRIPAC-DIG/SR-GNN
MA-GNN[10]	Single-type sequential interaction	Next-item prediction	GraphSage	AAAI'2020	https://github.com/cynricfu/MAGNN
FGNN[11]	Single-type sequential interaction	Next-item prediction	GAT	CIKM'2019	https://github.com/RuihongQiu/FGNN
MGNN-SPred[12]	Multi-type sequential interaction	Next-item prediction	GraphSage	WWW'2020	https://github.com/Autumn945/MGNN-SPred
HERec[13]	Explicit interaction + Attribute information	Rating prediction	Graph embedding	TKDE'2018	https://github.com/librahu/HERec
KGAT[14]	Implicit interaction + Attribute information	Click prediction	GAT	KDD'2019	https://github.com/xiangwang1223/
TrustWalker[15]	Explicit interaction + Trust relation	Rating prediction	Random walk	KDD'2009	https://github.com/Antili/TrustWalker
GraphRec[16]	Explicit interaction + Social relation	Rating prediction	GAT	WWW'2019	https://github.com/wenqifan03/GraphRec
KGNN-LS[17]	Implicit interaction + External knowledge	Click prediction	GAT	KDD'2019	https://github.com/hwwang55/KGNN-LS

[1][Berg *et al.*, 2017];[2][Monti *et al.*, 2017];[3][Zhang and Chen, 2020];[4][Nikolakopoulos and Karypis, 2019];[5][Ying *et al.*, 2018]; [6][Chen *et al.*, 2019];[7][He *et al.*, 2020];[8][Zheng *et al.*, 2018];
[9][Wu *et al.*, 2019b];[10][Ma *et al.*, 2020];[11][Qiu *et al.*, 2019];[12][Wang *et al.*, 2020c];[13][Shi *et al.*, 2018];[14][Wang *et al.*, 2019f];[15][Jamali and Ester, 2009];[16][Fan *et al.*, 2019];[17][Wang *et al.*, 2019b]

Table 2: A list of representative open-source GLRS algorithms

Dataset	Domain	Information Included	# Interactions	Reference	Link
MovieLens-1M	Movie	Explicit interaction	1,000,209	[Zheng *et al.*, 2018]	https://grouplens.org/datasets/
HetRec	Movie	Explicit interaction	855,598	[Zheng *et al.*, 2018]	https://grouplens.org/datasets/
Amazon instant video	Video	Explicit interaction	583,933	[Zheng *et al.*, 2018]	http://jmcauley.ucsd.edu/data/amazon/
Gowalla	POI	Implicit interaction	1,027,370	[He *et al.*, 2020]	http://snap.stanford.edu/data/loc-gowalla.html
Yelp 2018	POI	Implicit interaction	1,561,406	[He *et al.*, 2020]	https://www.yelp.com/dataset
Amazon-book	E-commerce	Implicit interaction	2,984,108	[He *et al.*, 2020]	https://github.com/uchidalab/book-dataset
Yoochoose 1/4	E-commerce	Stream of clicks	8,326,407	[Wu *et al.*, 2019b]	http://2015.recsyschallenge.com/challege.html
Diginetica	E-commerce	Stream of clicks	982,961	[Wu *et al.*, 2019b]	https://competitions.codalab.org/competitions/11161
Book-crossing	Book	Ratings and attribute information	1,000,000	[Wang *et al.*, 2019b]	http://www2.informatik.uni-freiburg.de/~cziegler/BX/
Last.FM	Music	Implicit interaction, social and tag	92,834	[Wang *et al.*, 2019b]	https://grouplens.org/datasets/hetrec-2011/
Epinions	E-commerce	Rating, trust relation	764,352	[Fan *et al.*, 2019]	http://alchemy.cs.washington.edu/data/epinions/
Ciao	E-commerce	Rating, trust relation	283,319	[Fan *et al.*, 2019]	https://www.cse.msu.edu/~tangjili/datasetcode/truststudy.htm
Amazon-toys and games	E-commerce	Implicit interaction	167,597	[Gao *et al.*, 2019]	http://jmcauley.ucsd.edu/data/amazon
Amazon-digital music	Music	Implicit interaction	64,706	[Gao *et al.*, 2019]	http://jmcauley.ucsd.edu/data/amazon

Table 3: A list of commonly used and publicly accessible real-world datasets for GLRS

High-efficiency online RS with large-scale graph learning. An inevitable issue in real RS is the scale of data, which is often large and leads to high cost in terms of both time and space. This issue is even more important in GLRS since the graph structure data is usually even larger and requires more time and space to be processed, let alone to perform complex machine learning techniques on it to generate recommendations. Therefore, it is necessary to study more efficient algorithms to speed up large-scale online processing and learning to keep updating model to generate timely recommendations.

6 Conclusions

As one of the most important applications of Artificial Intelligence (AI), Recommender Systems (RS) can be found nearly at every corner of our daily lives. Graph Learning (GL), as one of the most promising AI techniques, has shown a great capability to learn the complex relations among the various objects managed by an RS. This has launched a totally new RS paradigm: Graph Learning based Recommender Systems (GLRS), which is of great potential to be the next-generation of RS. It is our hope that this review has provided a comprehensive and self contained overview of the recent progress, challenges as well as future research directions in GLRS to both the academia and industry.

Acknowledgements

This work was supported by ARC Project DP180102378.

References

[Backstrom and Leskovec, 2011] Lars Backstrom and Jure Leskovec. Supervised random walks: predicting and recommending links in social networks. In *WSDM*, pages 635–644, 2011.

[Bagci and Karagoz, 2016] Hakan Bagci and Pinar Karagoz. Context-aware friend recommendation for location based social networks using random walk. In *WWW*, pages 531–536, 2016.

[Baluja *et al.*, 2008] Shumeet Baluja, Rohan Seth, Dharshi Sivakumar, and et al. Video suggestion and discovery for youtube: taking random walks through the view graph. In *WWW*, pages 895–904, 2008.

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

[Cen *et al.*, 2019] Yukuo Cen, Xu Zou, Jianwei Zhang, and et al. Representation learning for attributed multiplex heterogeneous network. In *SIGKDD*, pages 1358–1368, 2019.

[Chen *et al.*, 2016] Chih-Ming Chen, Ming-Feng Tsai, Yu-Ching Lin, and Yi-Hsuan Yang. Query-based music recommendations via preference embedding. In *RecSys*, pages 79–82, 2016.

[Chen *et al.*, 2019] Chih-Ming Chen, Chuan-Ju Wang, Ming-Feng Tsai, and Yi-Hsuan Yang. Collaborative similarity embedding for recommender systems. In *WWW*, pages 2637–2643, 2019.

[Cui *et al.*, 2019] Zeyu Cui, Zekun Li, Shu Wu, and et al. Dressing as a whole: Outfit compatibility learning based on node-wise graph neural networks. In *WWW*, pages 307–317, 2019.

[Fan *et al.*, 2019] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, and et al. Graph neural networks for social recommendation. In *WWW*, pages 417–426, 2019.

[Gao *et al.*, 2019] Jingyue Gao, Xiting Wang, Yasha Wang, and et al. Explainable recommendation through attentive multi-view learning. In *AAAI*, pages 3622–3629, 2019.

[Gao *et al.*, 2020] Yang Gao, Yi-Fan Li, Yu Lin, and et al. Deep learning on knowledge graph for recommender system: A survey. *arXiv preprint arXiv:2004.00387*, 2020.

[Guo *et al.*, 2020] Qingyu Guo, Fuzhen Zhuang, Chuan Qin, and et al. A survey on knowledge graph-based recommender systems. *IEEE Transactions on Knowledge and Data Engineering*, 2020. doi: 10.1109/TKDE.2020.3028705.

[Han *et al.*, 2018] Xiaotian Han, Chuan Shi, Senzhang Wang, and et al. Aspect-level deep collaborative filtering via heterogeneous information networks. In *IJCAI*, pages 3393–3399, 2018.

[He *et al.*, 2020] Xiangnan He, Kuan Deng, Xiang Wang, and et al. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *SIGIR*, pages 639–648, 2020.

[Hu *et al.*, 2014] Liang Hu, Jian Cao, Guandong Xu, and et al. Deep modeling of group preferences for group-based recommendation. In *AAAI*, pages 1861–1867, 2014.

[Hu *et al.*, 2018] Binbin Hu, Chuan Shi, Wayne Xin Zhao, and Philip S Yu. Leveraging meta-path based context for top-n recommendation with a neural co-attention model. In *SIGKDD*, pages 1531–1540, 2018.

[Hu *et al.*, 2019] Liang Hu, Songlei Jian, Longbing Cao, Zhiping Gu, and et al. Hers: Modeling influential contexts with heterogeneous relations for sparse and cold-start recommendation. In *AAAI*, pages 3830–3837, 2019.

[Huang *et al.*, 2015] Shangrong Huang, Jian Zhang, Lei Wang, and Xian-Sheng Hua. Social friend recommendation based on multiple network correlation. *IEEE transactions on multimedia*, 18(2):287–299, 2015.

[Huang *et al.*, 2019] Jin Huang, Zhaochun Ren, Wayne Xin Zhao, Gaole He, Ji-Rong Wen, and et al. Taxonomy-aware multi-hop reasoning networks for sequential recommendation. In *WSDM*, pages 573–581, 2019.

[Jamali and Ester, 2009] Mohsen Jamali and Martin Ester. Trustwalker: a random walk model for combining trust-based and item-based recommendation. In *SIGKDD*, pages 397–406, 2009.

[Jiang *et al.*, 2018] Zhengshen Jiang, Hongzhi Liu, Bin Fu, and et al. Recommendation in heterogeneous information networks based on generalized random walk model and bayesian personalized ranking. In *WSDM*, pages 288–296, 2018.

[Jin *et al.*, 2020] Bowen Jin, Chen Gao, Xiangnan He, Depeng Jin, and Yong Li. Multi-behavior recommendation with graph convolutional networks. In *SIGIR*, pages 659–668, 2020.

[Li and Chen, 2013] Xin Li and Hsinchun Chen. Recommendation as link prediction in bipartite graphs: A graph kernel-based machine learning approach. *Decision Support Systems*, 54(2):880–890, 2013.

[Ma *et al.*, 2020] Chen Ma, Liheng Ma, Yingxue Zhang, and et al. Memory augmented graph neural networks for sequential recommendation. In *AAAI*, pages 5045–5052, 2020.

[Mikolov *et al.*, 2013] Tomas Mikolov, Ilya Sutskever, Kai Chen, and et al. Distributed representations of words and phrases and their compositionality. In *NIPS*, pages 3111–3119, 2013.

[Monti *et al.*, 2017] Federico Monti, Michael M Bronstein, and Xavier Bresson. Geometric matrix completion with recurrent multi-graph neural networks. In *NIPS*, pages 3700–3710, 2017.

[Nikolakopoulos and Karypis, 2019] Athanasios N Nikolakopoulos and George Karypis. Recwalk: Nearly uncoupled random walks for top-n recommendation. In *WSDM*, pages 150–158, 2019.

[Palumbo *et al.*, 2017] Enrico Palumbo, Giuseppe Rizzo, and Raphaël Troncy. Entity2rec: Learning user-item relatedness from knowledge graphs for top-n item recommendation. In *RecSys*, pages 32–36, 2017.

[Qiu *et al.*, 2019] Ruihong Qiu, Jingjing Li, Zi Huang, and Hongzhi Yin. Rethinking the item order in session-based recommendation with graph neural networks. In *CIKM*, pages 579–588, 2019.

[Quadrana *et al.*, 2018] Massimo Quadrana, Paolo Cremonesi, and Dietmar Jannach. Sequence-aware recommender systems. *ACM Computing Surveys (CSUR)*, 51(4):1–36, 2018.

[Sheu and Li, 2020] Heng-Shiou Sheu and Sheng Li. Context-aware graph embedding for session-based news recommendation. In *RecSys*, pages 657–662, 2020.

[Shi *et al.*, 2014] Yue Shi, Martha Larson, and Alan Hanjalic. Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges. *ACM Computing Surveys (CSUR)*, 47(1):1–45, 2014.

[Shi *et al.*, 2018] Chuan Shi, Binbin Hu, Wayne Xin Zhao, and S Yu Philip. Heterogeneous information network embedding for recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 31(2):357–370, 2018.

[Song *et al.*, 2019] Weiping Song, Zhiping Xiao, Yifan Wang, and et al. Session-based social recommendation via dynamic graph attention networks. In *WSDM*, pages 555–563, 2019.

[Sun *et al.*, 2018] Zhu Sun, Jie Yang, Jie Zhang, Alessandro Bozzon, Long-Kai Huang, and Chi Xu. Recurrent knowledge graph embedding for effective recommendation. In *RecSys*, pages 297–305, 2018.

[Verma *et al.*, 2019] Janu Verma, Srishti Gupta, Debdoot Mukherjee, and et al. Heterogeneous edge embedding for friend recommendation. In *ECIR*, pages 172–179, 2019.

[Wang *et al.*, 2017] Shoujin Wang, Liang Hu, and Longbing Cao. Perceiving the next choice with comprehensive transaction embeddings for online recommendation. In *ECML-PKDD*, pages 285–302, 2017.

[Wang *et al.*, 2018a] Hongwei Wang, Fuzheng Zhang, Jialin Wang, Miao Zhao, Wenjie Li, and et al. Ripplenet: Propagating user preferences on the knowledge graph for recommender systems. In *CIKM*, pages 417–426, 2018.

[Wang *et al.*, 2018b] Xiang Wang, Xiangnan He, Fuli Feng, and et al. Tem: Tree-enhanced embedding model for explainable recommendation. In *WWW*, pages 1543–1552, 2018.

[Wang *et al.*, 2019a] Haoyu Wang, Defu Lian, and Yong Ge. Binarized collaborative filtering with distilling graph convolutional networks. In *IJCAI*, pages 4802–4808, 2019.

[Wang *et al.*, 2019b] Hongwei Wang, Fuzheng Zhang, and et al. Knowledge-aware graph neural networks with label smoothness regularization for recommender systems. In *SIGKDD*, pages 968–977, 2019.

[Wang *et al.*, 2019c] Hongwei Wang, Fuzheng Zhang, Miao Zhao, and et al. Multi-task feature learning for knowledge graph enhanced recommendation. In *WWW*, pages 2000–2010, 2019.

[Wang *et al.*, 2019d] Shoujin Wang, Longbing Cao, Yan Wang, and et al. A survey on session-based recommender systems. *arXiv preprint arXiv:1902.04864*, 2019.

[Wang *et al.*, 2019e] Shoujin Wang, Liang Hu, Yan Wang, and et al. Sequential recommender systems: challenges, progress and prospects. In *IJCAI*, pages 6332–6338, 2019.

[Wang *et al.*, 2019f] Xiang Wang, Xiangnan He, Yixin Cao, and et al. Kgat: Knowledge graph attention network for recommendation. In *SIGKDD*, pages 950–958, 2019.

[Wang *et al.*, 2019g] Xiang Wang, Dingxian Wang, Canran Xu, Xiangnan He, and et al. Explainable reasoning over knowledge graphs for recommendation. In *AAAI*, pages 5329–5336, 2019.

[Wang *et al.*, 2019h] Zekai Wang, Hongzhi Liu, Yingpeng Du, and et al. Unified embedding model over heterogeneous information network for personalized recommendation. In *IJCAI*, pages 3813–3819, 2019.

[Wang *et al.*, 2020a] Nan Wang, Shoujin Wang, Yan Wang, and et al. Modelling local and global dependencies for next-item recommendations. In *WISE*, pages 285–300, 2020.

[Wang *et al.*, 2020b] Shoujin Wang, Liang Hu, Yan Wang, and et al. Intention2basket: A neural intention-driven approach for dynamic next-basket planning. In *IJCAI*, pages 2333–2339, 2020.

[Wang *et al.*, 2020c] Wen Wang, Wei Zhang, Shukai Liu, Qi Liu, Bo Zhang, Leyu Lin, and Hongyuan Zha. Beyond clicks: Modeling multi-relational item graph for session-based target behavior prediction. In *The Web Conference*, pages 3056–3062, 2020.

[Wen *et al.*, 2018] Yufei Wen, Lei Guo, Zhumin Chen, and Jun Ma. Network embedding based recommendation method in social networks. In *WWW*, pages 11–12, 2018.

[Wu *et al.*, 2019a] Le Wu, Peijie Sun, Yanjie Fu, and et al. A neural influence diffusion model for social recommendation. In *SIGIR*, pages 235–244, 2019.

[Wu *et al.*, 2019b] Shu Wu, Yuyuan Tang, Yanqiao Zhu, and et al. Session-based recommendation with graph neural networks. In *AAAI*, pages 346–353, 2019.

[Wu *et al.*, 2020a] Le Wu, Junwei Li, Peijie Sun, and et al. Diffnet++: A neural influence and interest diffusion network for social recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 2020. doi: 10.1109/TKDE.2020.3048414.

[Wu *et al.*, 2020b] Shiwen Wu, Wentao Zhang, Fei Sun, and Bin Cui. Graph neural networks in recommender systems: A survey. *arXiv preprint arXiv:2011.02260*, 2020.

[Wu *et al.*, 2021] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1):4–24, 2021.

[Xu *et al.*, 2019] Chengfeng Xu, Pengpeng Zhao, and et al. Graph contextualized self-attention network for session-based recommendation. In *IJCAI*, pages 3940–3946, 2019.

[Yang *et al.*, 2018] Jheng-Hong Yang, Chih-Ming Chen, Chuan-Ju Wang, and Ming-Feng Tsai. Hop-rec: high-order proximity for implicit recommendation. In *RecSys*, pages 140–144, 2018.

[Yin *et al.*, 2010] Zhijun Yin, Manish Gupta, Tim Weninger, and Jiawei Han. Linkrec: a unified framework for link recommendation with user attributes and graph structure. In *WWW*, pages 1211–1212, 2010.

[Ying *et al.*, 2018] Rex Ying, Ruining He, and et al. Graph convolutional neural networks for web-scale recommender systems. In *SIGKDD*, pages 974–983, 2018.

[Yu and Qin, 2020] Wenhui Yu and Zheng Qin. Graph convolutional network for recommendation with low-pass collaborative filters. In *ICML*, pages 10936–10945, 2020.

[Zha *et al.*, 2001] Hongyuan Zha, Xiaofeng He, Chris Ding, Horst Simon, and Ming Gu. Bipartite graph partitioning and data clustering. In *CIKM*, pages 25–32, 2001.

[Zhang and Chen, 2018] Yongfeng Zhang and Xu Chen. Explainable recommendation: A survey and new perspectives. *arXiv preprint arXiv:1804.11192*, 2018.

[Zhang and Chen, 2020] Muhan Zhang and Yixin Chen. Inductive matrix completion based on graph neural networks. In *ICLR*, 2020.

[Zhang *et al.*, 2019a] Jiani Zhang, Xingjian Shi, Shenglin Zhao, and Irwin King. Star-gcn: Stacked and reconstructed graph convolutional networks for recommender systems. In *IJCAI*, 2019.

[Zhang *et al.*, 2019b] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. Deep learning based recommender system: A survey and new perspectives. *ACM Computing Surveys*, 52(1):1–38, 2019.

[Zhao *et al.*, 2017] Huan Zhao, Quanming Yao, Jianda Li, and et al. Meta-graph based recommendation fusion over heterogeneous information networks. In *SIGKDD*, pages 635–644, 2017.

[Zheng *et al.*, 2018] Lei Zheng, Chun-Ta Lu, Fei Jiang, and et al. Spectral collaborative filtering. In *RecSys*, pages 311–319, 2018.

[Zhu *et al.*, 2019] Feng Zhu, Chaochao Chen, Yan Wang, and et al. Dtcdr: A framework for dual-target cross-domain recommendation. In *CIKM*, pages 1533–1542, 2019.

[Zhu *et al.*, 2021] Feng Zhu, Yan Wang, Chaochao Chen, and et al. Cross-domain recommendation: Challenges, progress, and prospects. *arXiv preprint arXiv:2103.01696*, 2021.