Recommendation System Based on Heterogeneous Feature: A Survey

HUI WANG1,2, ZICHUN LE3, AND XUAN GONG1
1College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310023, China
2College of Applied Science, Jiangxi University of Science and Technology, Ganzhou 341000, China
3College of Science, Zhejiang University of Technology, Hangzhou 310023, China
Corresponding author: Zichun Le (lzc@zjut.edu.cn)

This work was supported in part by the Special Funding of the Belt and Road International Cooperation of Zhejiang Province under Grant 2015C04005.

ABSTRACT Recommendation systems have become an important field of research in computer science and physics. In recent years, breakthroughs have been achieved in social, biological, and research cooperation networks. With the popularization of big data and deep learning technology development, graph structures are increasingly being used to represent large-scale and complex data in the real world. In this paper, we reviewed the progress made in recommendation systems research in the past 20 years and comprehensively classified recommender systems based on the heterogeneous input features. We introduced layering in the classification of recommendation systems. Furthermore, we proposed a new hierarchical classification model of recommendation systems divided into three layers: feature input, feature learning, and output layers. In the feature learning layer, existing recommendation systems were divided into graph-based, text-based, behavior-based, spatiotemporal-based, and hybrid recommendation systems. Additionally, we provided evaluation index, open-source implementation, experimental comparison and the relative merits for each recommendation method. Subsequently, future development directions of recommendation systems are discussed.

INDEX TERMS Behavior features, feature, graph features, recommendation systems, text features.

I. INTRODUCTION The rapid development of the Internet has brought us from an era of relatively low information to an era of information explosion. Users often experience information overload due to daily exponential growth in the amount of information, the increasing number of information resources, and rapidly developing transmission mode. Although users receive massive amounts of information daily from major portals, social media channels, and other platforms, they often ignore most of the information received because they do not meet their needs and preferences. Frequently, needed information is submerged in the ocean of information, making it difficult to be discovered. Consequently, users waste time and network resources searching for needed information. Recommendation systems were introduced to mitigate this problem.

A recommendation system can determine the degree to which a target user likes a product based on the user’s history and personal data and determine whether to recommend the product to the user based on the ranking.

Multiple recommendation systems have been developed for different applications. The first paper on recommendation systems was published in 1997. Presently, 76372 scientific papers are published under the theme of recommendation system in the Web of Science, of which 7733 (16%) are review papers. Most of the existing recommendation systems are divided into collaborative filtering (CF), content-based, and mixed recommendation methods in the review papers. However, these methods are based on the shallow model, which fails to learn the deep features of users and items. Furthermore, there are some challenges and difficulties in processing texts, images, and other heterogeneous information. In 2020, Aminu and Naomie [1] published a representative review paper that divided the recommendation system into a traditional recommendation system and a deep learning-based recommendation and introduced the recommendation algorithm thoroughly. Notably, recommendation systems based on restricted boltzmann machine (RBM), generative adversarial network (GAN), recurrent neural network (RNN), convolutional neural network (CNN), and graph autoencoder (GAE) were introduced thoroughly. However, they did not consider recommendation systems based on graph
models. In 2020, Zhou et al. [2] published a review paper in which they divided deep learning-based recommendation methods into four categories, namely deep neural network (DNN), CNN, recurrent neural network (RNN) and short-term and long-term memory (LSTM), and graph neural network (GNN)-based recommendation methods. Although they mentioned recommendation systems based on graph structure features, the introduction was not comprehensive and detailed.

To close the gap in existing research, we introduced a layered design approach in this paper. Furthermore, we proposed a new recommendation system classification model based on heterogeneous features that divides existing recommendation systems into graph-based, text-based, behavior-based, spatiotemporal-based, and hybrid recommendation systems.

The structure of this paper is as follows: In Section II, we defined the recommendation system. A new hierarchical classification model of recommendation systems based on heterogeneous features, covering multiple classic and emerging recommendation methods, is proposed in Section III. In Section IV, we summarized the evaluation index. An experimental comparison is established in Section V. We analyzed the advantages and disadvantages of each recommendation method in Section VI. In Section VII, we presented challenges and future development directions. The conclusion is in Section IX.

II. THE DEFINITION OF RECOMMENDATION SYSTEM
In 1997, Resnick and Varian [3] proposed the term recommendation system. Subsequently, the term recommendation system has been widely cited, which has become an important area of research. Recommendation system simulate human behavior using relevant algorithms to analyze and process information provided by others and then present the results to users. Burke [4] state that any system could output personalized recommendation information or guide users in a personalized way to find interesting or valuable information embedded in a large amount of information, which can be called a recommendation system. Schafer et al. [5] conclude that the recommendation system is a part of e-commerce websites. By analyzing and learning customers’ consumption behaviors on the websites, their future consumption behaviors can be predicted. Therefore, e-commerce websites can provide personalized services on the basis of the preferences of each customer.

III. A NEW HIERARCHICAL MODEL OF RECOMMENDATION SYSTEM BASED ON HETEROGENEOUS FEATURES
Experts in physics, computer science, engineering, and other fields have proposed various general and basic recommendation methods. Most of the proposed methods use text, graph structure, behavior, an spatiotemporal features to compute node similarity and utilize feature extraction to learn the implicit representation of users and items. The implicit representation is used to generate item recommendations for users. On the basis of the variety of input features, we proposed a new hierarchical model of recommendation system based on heterogeneous features, as shown in Figure 1. The proposed model includes feature input, feature learning, and output layers. Each layer and its corresponding recommendation method are discussed systematically below.

A. FEATURE INPUT LAYER
The feature input layer is the first layer of the hierarchical model of the recommendation system based on heterogeneous features. The data fed into the input layer include textual features such as attribute features, ratings, and likes/dislikes showing the feedback, behavior features of implicit feedback such as browsing, and clicking, graph structure features, and spatiotemporal features. The four commonly used input features are described below.

1) GRAPH STRUCTURAL FEATURES
The graph structure features include eight popular heuristics such as Common Neighbors (CN), Jaccard, Preferential Attachment (PA), Adamic-Adar (AA), Resource Allocation (RA), Katz, PageRank, and SimRank and node centrality scores (degree, closeness, and betweenness).

2) TEXT FEATURES
Multiple studies have shown that using node and link content, attributes, and image features such as user age, interest, and photos as input features can significantly enhance the recommendation accuracy. The attribute information of nodes exists in the form of text. Therefore, the similarity between two nodes can be computed using text and string-based similarity indicators, which can be sorted to achieve high recommendation accuracy.

3) BEHAVIOR FEATURES
Behavior features include user click, browse, search, purchase, and comment behavior. Among these behavior features, effective analysis and mining of user comment behavior are the current research hotspot and difficulty. Sentiment tendencies, precise customer needs, and accurate recommendations can be generated from the customer comments, including text, voice, video, and images [6]. In 2016, Lei and Zhu [7] applied sentiment analysis to movie recommendations, which considered the effect of comment behavior on the similarity of interest among users. They achieved a better recommendation by analyzing comments’ emotion and computing users’ emotional tendency toward different movies.

4) SPATIOTEMPORAL FEATURES
Spatiotemporal features include both temporal and spatial features. It considers space and time dependencies, which is becoming more important in applications such as traffic and human activity prediction. Zhao et al. [8] proposed a new recurrent neural network model, named spatiotemporal LSTM ST-LSTM, to model users’ sequential visiting...
behavior. Time intervals and distance intervals features of neighbor check-ins were modeled using time and distance gate, which further integrate coupled input and forget gates with the proposed model. Extensive experiments confirm that the model outperformed the state-of-the-art recommendation methods.

**B. FEATURE LEARNING LAYER**

The feature learning layer is the second layer of the hierarchical classification model of the recommendation system based on heterogeneous features. The main function of this layer is to learn useful features from the upper layer. This layer divides the conventional recommendation algorithms into graph-based, text-based, behavior-based, spatiotemporal-based, and hybrid recommendation systems based on the heterogeneous feature.

1) **GRAPH-BASED RECOMMENDATION SYSTEM**

Graph-based recommendation systems is a research area in recommendation systems. It represents items and users as nodes, and items-to-items, users-to-users, users-to-items, and content information as node status information. Considering the complex relationship between objects during recommendations can enhance the representation of objects and interpretability of the recommendation system using graph reasoning; consequently, providing high-quality recommendation results. In this paper, graph-based recommendation systems are divided into graph semi-supervised, graph-unsupervised, graph-supervised, and graph-based reinforcement learning recommendation systems. These methods are introduced, as shown in Figure 2.

1) **Graph semi-supervised learning recommendation system**

The basic idea of graph-based semi-supervised learning is to construct graphs for all samples (including labeled and unlabeled), where the samples are considered as nodes of the graph. The similarity between the samples generated based on a specific predefined measure is the edge of the graph. If two samples $x_i$ and $x_j$ are considered to be similar under the measure, then an edge exists between $i$ and $j$; otherwise, there is no edge. After building the graph, various derived algorithms can infer the labels of unlabeled data based on the basic assumption about graphs that the labels of nodes
are smooth across the graph, implying that similar samples tend to have the same labels. This paper divides the graph-based semi-supervised learning recommendation system into graph convolution network (GCN)-based, graph neural network (GNN)-based, graph attention network (GAT)-based, graph autoencoder (GAE)-based, and graph RNN (GRNN)-based recommendation systems.

**GCN-based recommendation system**

Multiple works have studied GCN and have achieved useful theoretical and practical results. After describing a problem with graph signals, the corresponding GCN is designed to solve the graph problem based on the characteristics of the problem. Specific to the recommendation system, the users and items are regarded as the vertices of the graph, and the user’s ratings of the items are regarded as the links. Furthermore, the users and items features are regarded as signals distributed on the vertices. Consequently, the recommendation problem is transformed into the link prediction of the graph problem.

In 2018, Ying et al. [9] proposed a scalable GCN algorithm for GCN called PinSage. The PinSage model adopted local convolution to reduce the complexity of the model in training and calculation. Efficient random walk and graph convolution are combined to generate node embedding. The node embeddings combine graph structure and node feature information. This model reveals the influence of the graph convolution method on the production recommendation system and enhances the recommendation performance. In 2019, Zhang et al. [10] proposed a new layered and reconstructed GCN (STAR-GCN) structure to learn node representation, enhance the performance of the recommendation system, and especially enhance the recommendation effect under cold start conditions. STAR-GCN uses a set of GCN encoder-decoders, combined with intermediate monitoring to enhance the final prediction performance, and takes low-dimensional users and item potential factors as inputs to suppress the model space complexity. Furthermore, STAR-GCN generates node embedding of new nodes by reconstructing the mask input node embedding, which solves the cold start problem.

**GNN-based recommendation system**

The GNN recommendation considers how to map different entities in the recommendation system to graph nodes in the GNN, map the connections between different entities to the corresponding edges in the GNN graph, and use neural network functions to fit different nodes in the GNN. In the GNN recommendation model, whether it is a user or an item node, embedding processing is performed using relevant information to generate a vectorized description of the corresponding node.

In 2019, Cui et al. [11] used GNN for clothing recommendation, service categories are used as nodes, and the relationship between each category as edges. Various services are placed in corresponding nodes to form subgraphs, which form an undirected unit graph GNN recommendation model. The target service with a higher matching degree of designated services is used to achieve the matching of packages, which achieved good results. In 2019, Fan et al. [12] used three undirected graphs, namely, the user–item relationship binary graph (all items purchased by the user), the user–friend relationship unit graph (the user’s social relationship), and the item–user relationship binary graph (the record of items purchased by the user), to rate and predict the items. Furthermore, given the possible behaviors of users, which are related to the items they once bought and closely related to their social relations, this paper proposed a new GNN recommendation system called GraphRec. The framework is mainly composed of three modules, namely, the user–item, user–friend, and item–user relationships. The user–friend relationship constitutes the user module, the item–user relationship constitutes the item module, and the recommendation module is constituted by a multilayer perceptron. The nodes in these three graphs are calculated using the information propagation update algorithm, and score predictions are generated based on the results. This method integrates the user’s social relationship into the neural network. Furthermore, the user features and the items are related, and the introduction of the attention mechanism helped achieve good results. However, the two GNN models in the user module cannot be calculated in parallel, which increased the model’s time complexity.

**GAT-based recommendation system**

The attention mechanism has become the standard in sequence tasks. Its value lies in being able to focus on the most important part of the object. This mechanism proved to be useful in the field of machine translation and recommendation systems. The attention mechanism becomes
more beneficial to information recommendation as the model capacity increases.

In 2019, Xu et al. [13] proposed a graph contextualized self-attention model (GC-SAN), which uses both GNN and self-attention mechanism, for session-based recommendations. They dynamically constructed a graph structure for session sequence and captured deep local dependencies using GNN. Subsequently, each session learns long-range dependencies by applying the self-attention mechanism. Each session is represented as a linear combination of the global preference and the session’s current interest. Extensive experiments on two real data sets show that GC-SAN performed better than the state-of-the-art technical methods. In 2019, Wang et al. [14] examined the utility of the knowledge graph (KG), which breaks the assumption of independence by associating the item with its attributes. They presented a new knowledge graph attention network (KGAT) method that explicitly models the high-order connections in terms of KG in an end-to-end model. It recursively propagates from the node’s neighbors (which can be users, items, or attributes) to refine the node’s embedding and uses an attention mechanism to distinguish the importance of the neighbors. KGAT is conceptually applicable to existing KG-based recommendation approaches, which can take advantage of high-level relationships by extracting paths, or implicitly modeled by regularization. Empirical results from three public benchmarks showed that KGAT is significantly superior to the neural factorization machine (FM) and RippleNet methods. Further studies validate the efficacy of embedding propagation for high-order relation modeling and the interpretability benefits provided by the attention mechanism.

- GAE-based recommendation system

Van et al. [15] consider matrix completion for recommender systems based on link prediction on graphs. Building on recent progress in deep learning and graph-structured data, the author proposed a GAE framework based on differentiable message passing on the bipartite interaction graph. Formulating matrix completion as a link prediction task on a bipartite graph becomes possible when recommender graphs are accompanied by structured external information such as social networks. Combining such external information with interaction data can mitigate performance bottlenecks related to the cold start problem. The author demonstrated that the GAE model efficiently combines interaction data with side information, without resorting to recurrent frameworks.

- GRNN-based recommendation systems

RNN is a neural network that can process sequence data. Most RNNs can handle variable-length sequences. The introduction of RNN solved the limitations of traditional neural networks in processing sequence information. When RNN acts on the map, we call this architecture GraphRNN, its main function is to define the figure or node status information. In the recommendation system, the user’s historical interaction records can be abstracted into sequence data. Subsequently, RNN can be used to mine the user’s interest preferences and other changes over time. Recommendation systems based on the GraphRNN are classified as semi-supervised and graph-unsupervised learning recommendation systems.

You et al. [16] applied GraphRNN to graph generation problems. They used two RNNs, one for generating new nodes and the other for auto-regressively generating edges for newly added nodes. The results show that the hierarchical RNN structure can effectively learn from the input graph and enhance the recommendation performance within an acceptable time complexity.

2) Graph-unsupervised learning recommendation system

In unsupervised learning, an algorithm tends to learn or draw inferences from a large number of unlabeled samples. There are many specific unsupervised learning algorithms. The graph-based unsupervised learning method has been a research hotspot in recent years, which mainly extracts the graph structure features of the network. The unsupervised indicators can reveal the relationship between two nodes in the network, which can be used to make recommendations. In this paper, we divided the graph-based unsupervised learning recommendation system into DeepWalk-based, Node2vec-based, random walk-based, and graph clustering recommendation systems.

- Deepwalk-based recommendation system

DeepWalk is a widely used network embedding algorithm that learns the potential vector representation of nodes in a network. It takes the graph as input and outputs the potential vector representation of all nodes in the graph. DeepWalk can be used to get the author vector. In DeepWalk, random paths in a network are used as sentences and nodes as words. Subsequently, applying sentence sequences to skip-gram, the algorithm can learn the distributed representation of nodes.

In 2019, Guo et al. [17] proposed a citation recommendation method based on content-aware bibliographic network representation, called CR-CA, which used the DeepWalk to obtain the vector representation of the paper and vectorize the candidate and target papers. The author used Euclidean distances to make recommendations on the target paper. The experimental results show that the method can handle problems caused by highly cited papers. Furthermore, the DeepWalk-based method performed better than the baseline method, which enhanced citation recommendation performance.

- Node2vec-based recommendation system

Node2vec is an enhanced version of DeepWalk. It enhanced the random walk strategy by utilizing parameters p and q and considered the macroscopic and microcosmic information simultaneously. Node2vec controls the walker’s transition probability, where p represents the returning probability and q denotes the probability of jumping to the next node. Conversely, DeepWalk sets p and q to 1 and ignores other factors that influence the generation of sentence sequences.
In 2018, Enrio et al. [18] proposed Node2Vec to generate item recommendations by learning knowledge graph embeddings, efficiently solving important problems such as new items and data sparsity. Node2Vec uses node dependencies on MovieLens and DBpedia data sets to generate items recommendations. The results show that node2vec performs better than other CF algorithms.

- Random Walk-based recommendation system

Graph model scheme is another way to build a model to implement recommendations, which can effectively calculate the similarity of nodes in the graph and explore the correlation between nodes. Furthermore, it can capture the modeling of high-order relationships, interactive propagation, and implicit preferences of various entities on the graph. The disadvantages are low efficiency and lack of model parameters due to optimization. Many researchers have applied random walk algorithms in link prediction and recommendation system and achieved good results.

The earliest graph recommendation algorithm is the random walk-based topic-sensitive PageRank (TSPR) algorithm proposed by Haveliwala [19] in 2003. The algorithm uses a first-order Markovchain to calculate the walk probability. The method is as follows: starting from each user node, performing a round of random walk on the “user–item” bipartite graph. Whenever a node is reached, it is important to determine whether to continue walking downstream with probability $\beta$ or return to the starting point with a probability of $1-\beta$ to repeat the walk process until the probabilities (similarity) of the user and the items on the bipartite graph converge to a stable value. Subsequently, a recommendation list of items for each user is generated according to the user’ probability of traveling on the items. The TSPR algorithm is simple to implement and has high recommendation accuracy. However, the algorithm’s time complexity is high, and the performance degradation caused by data sparsity also exists.

In 2019, Nzeko et al. [20] proposed a novel graph-based framework GraFC2T2, which enable easy combination and comparison of top-N recommended edge information. This method encodes the content’s features, time, and trust information into a complex graph on which recommendations are performed using a personalized PageRank. The author tested the proposed method on Epinions and Ciao data sets, and the results showed that the proposed method enhanced the recommendation effect.

- Graph clustering recommendation system

Graph clustering is a relatively new clustering method. First, it expresses the data as a graph and then transforms the clustering problem into a graph partition problem. Multiple studies have confirmed that graph clustering methods perform better than other clustering methods. In recent years, implement clustering algorithms on the structural features of graphs and apply them to recommendation systems has gained much research attention, and a series of research results have been obtained.

In 2016, Liu et al. [21] proposed a user clustering recommendation algorithm based on a bipartite graph trust network, which clustered users with similar scores to form several user groups. Connection was established within each group using bipartite graph, which effectively alleviated the recommendation sparsity problem. Then, intragroup and intergroup trust connections were established using the trust mechanism to make the constructed recommendation results more accurate.

In 2019, Cao and Li [22] combined trust information and graph clustering method in the recommendation system, which expressed the user/item information as a graph. It then applies the approximate densest subgraph search algorithm on the graph to find the initial cluster center. Furthermore, it applies an iterative method to update the cluster center until the cluster is merged. The found clusters are used as neighbors to predict the rating of unknown items and recommend the top N items to active users. Multiple experiments have shown that the graph clustering recommendation algorithm performs better than other clustering algorithms.

3) Graph-supervised learning recommendation system

With the rapid development of graph deep learning technology, some researchers choose to use graph-based supervised learning technology for information recommendation. The supervised learning method can automatically learn the classification model based on the extracted graph’s explicit or implicit features, which overcomes some of the defects of unsupervised indicators. In this paper, graph-based supervised learning methods were included as random forest-based recommendation methods.

In 2018, Li and Hong [23] proposed a bipartite graph recommendation algorithm based on random forest correction. By making a more detailed division of the scoring weights, the weights of the projects were balanced, and the mining of popular projects was fully considered. The similarity calculation and accuracy of recommendations were enhanced. After standardizing the feature data, a random forest model is established, and the prediction list generated in the bipartite graph algorithm stage is corrected and verified for the scoring interval. The algorithm proposed in this paper enhances the accuracy of recommendation while ensuring better diversity recommended accuracy than the recommendation algorithm based on weighted bipartite graph (Network based inference, NBI) and the weighted bipartite graph recommendation algorithm ISWNBI based on increasing similarity coefficient.

4) Graph-based reinforcement learning recommendation system

Reinforcement learning is one of the machine learning paradigm and methodology used to describe agents in the process of interaction with the environment maximizes the reward of learning strategies to achieve or recommended problem. In this paper, the recommendation system of graph-based reinforcement learning is divided into graph convolutional policy network (GCPN) based recommendation system and graph transformation policy network (GTPN)-based recommendation system.
- GCPN based recommendation system

In 2018, You and Liu [24] proposed GCPN, which uses reinforcement learning to perform goal-oriented modular graph generation tasks to deal with non-differentiable goals and constraints. Notably, they modeled graph generation as a Markov decision process and used the generative model as a reinforcement learning agent running in the graph generation environment. GCPN uses similar agent actions as link prediction problems, domain-specific and confrontation rewards, and GCN to learn node representations, thereby implementing end-to-end training through the strategy gradient method. Experimental results show the effectiveness of GCPN in multiple graph generation problems. Furthermore, the application of GCPN can go far beyond molecular production. In many cases, the algorithm can be used to generate graphics, such as circuits and social network recommendation systems, and explore diagrams that can optimize domain-specific properties.

- GTPN based recommendation system

In 2018, Kien et al. [25] proposed GTPN, which predicts chemical reaction products. GTPN is a novel generic method that combines the strengths of GNNs and reinforcement learning to learn the reactions directly from data with minimal chemical knowledge. Compared with the previous approach, GTPN has some attractive attributes, such as end-to-end learning, and does not assume the graph transformation length or order. The experimental results on the large USPTO data set show that GTPN achieved an enhanced prediction accuracy and performance compared with the state-of-the-art methods.

2) TEXT-BASED RECOMMENDATION SYSTEM

The text-based recommendation system is the earliest recommendation algorithm used, which comes from the field of information acquisition. It is based on project information. Users’ interest data are obtained from examples of description of text attributes, and recommendation results should be generated based on users’ preferences and item feature information.

Zhang et al. [26] used multi-modal information to make hashta recommendations on Weibo. Thei work considers text and images using CNN and RNN to extract features from images and text and then combines the features of both aspects for label recommendation. Simultaneously, considering that tags are only related to some information in images and text, they used attention mechanisms to model local correlation. Yang et al. [27] proposed a context-aware citation recommendation model based on LSTMs to recommend relevant and appropriate scientific paper citations for users. First, the model learns the distribution representation of citation context and scientific papers based on LSTM. Second, it incorporates author, venue, and content information into the distributed vector representation of scientific papers. Third, it measures the relevance of scientific papers based on the learned citation context distribution. Fourth, it select scientific papers with higher relevance scores as the recommended list. Lee et al. [28] first used a CN to model the dialogue text information in Twitter and then added LSTM to the CNN layer to establish the dialogue timing relationship. Then, superimposed the Softmax unit layer on the top to perform predictions to complete the quotation recommendation on Twitter. In [29], deep belief network (DB) was directly used as the information recommendation model. In this paper, features such as users’ age, price range, brand material, design, and sewing method are used as DBN input data for brand recommendation, and the scores are regressed by mapping multilayer ReLU neurons to the softmax cell layer. The experimental results of DBN are higher than those of previous machine learning methods. Bansal et al. [30] proposed a text recommendation method based on an RNN model. The model used scoring data, text content information, and text label data for recommendations. First, the model uses gated recurrent unit (GRU) to learn the vector representation of text content and then use a multi-task learning framework to build a joint optimization objective function based on the hidden factor model and optimize the model parameters under the supervised learning framework. Compared with CDL and CRAE, although the model also uses the hidden vector representation of the item-related text data learning items, the model is a complete end-to-end supervised learning architecture.

The above examples show that the text-based recommendation system largely depends on the effective extraction of text features. Using deep learning technology for text recommendation, the model can learn the features of the data through an end-to-end process and automatically obtain a high-level representation of the data without relying on manual design functions. It can effectively ease the difficulty of the user and item feature extraction and cold start of new items and can integrate the user and item feature extraction and recommendation process into a unified framework but cannot solve the text-based recommendation method for new users, which has its problems. Furthermore, it can only find resources that are similar to the user’s existing interests.

3) BEHAVIOR-BASED RECOMMENDATION SYSTEM

There are multiple user behavior patterns, and the famous “Beer and Diaper” story is a good example of user behavior patterns. User behavior-based recommendation does not need to obtain feature data of users or products in advance. It uses only historical behavior data (explicit or implicit feedback) of existing users to predict items of interest to the current user. Explicit feedback is mainly user evaluation, and implicit feedback includes browsing and searching. In the field of recommendation, the behavior-based recommendation is the mainstream of research, and it is currently the most widely used recommendation method.

Wang [31] integrated human personality into the CF of users and proposed a CF system based on personalized information to solve the cold start problem often encountered by recommendation systems. The 2013 MovieLens user data
was used as experimental data for evaluation, which show that the cold start problem can be significantly alleviated. Zhao and Shang [32] proposed a user-based CF algorithm on the cloud computing platform Hadoop to solve CF’ scalability problem. The experimental results show that using simple methods to group users according to two basic principles can enhance the recommendation effect to a certain extent. Baralis and Garza [33] proposed an item-based CF algorithm to recommend target items that may be of interest to target users and used chi-square measures to measure the effectiveness of the algorithm. Wang [34] considered the rapid growth of items and users in e-commerce and aimed to solve the problems of data sparseness and scalability, using CF-based item clustering. They verified the effectiveness of the algorithm using the Movielens data set. Zhang et al. [35] proposed a CF recommendation algorithm model combined with DNN, which comprises a feature representation module and a score prediction module. First, the model enhances the traditional matrix factorization algorithm. It uses a quadratic polynomial regression model to capture the potential feature representations to make the potential feature representations obtained by the model more accurate and then inputs these potential features into the DNN to predict the score. This model can effectively enhance the recommended performance. Zhang et al. [36] proposed the DBNCF method in the massive open online courses (MOOC) environment, which utilizes DB’ high performance in function approximation, feature extraction, prediction classification, and other aspects. It combines the MOOC platform user course feature vector to mine the user’ course interests. Meanwhile, it uses the score of courses as the class label of DBN supervised learning. Through unsupervised pre-training and supervised feedback fine-tuning, the DBN recommendation model training can be achieved. The experiment was conducted in the real MOOC platform and was compared with several traditional recommendation methods. The experimental results show that the DBNCF is more efficient than the traditional cooperative filtering method. Zheng [37] proposed a cooperative neural network model. User evaluation and item profile features were extracted as user hidden factors and item hidden factors respectively through CN, and the scoring matrix was reconstructed through matrix decomposition. The adjustment process of parameters was completed using alternating optimization. Robin and Bersini [38] explored the LSTM for the CF recommendation problem. The experimental results showed that the LSTM achieved good results on the Movielens and Netflix data sets, especially in terms of short-term prediction and item coverage. In 2019, Lu and Li [39] built sentiment trust relationship model between users through sentiment analysis of user comment and combined trust relationship in the recommendation process to make up for the lack of similarity as the only preference weight in the traditional CF recommendation system, to enhance the recommendation accuracy. In 2018, Yang et al. [40] proposed to deeply dig out the comments of patients online in a Chinese medical network, considering patient preferences and using intuitionistic fuzzy sets with the Bonferroni mean to address interdependencies, which can help recommend patients to find a satisfactory doctor. In 2019, Li et al. [41] proposed a new deep learning framework for deep collaborative hashing codes on user–item ratings, which adopts neural networks to learn user and item representation and make these close to binary codes. Simultaneously, the author extends the proposed method for dealing with new user, new items, and new ratings. Experiments show the effectiveness of the proposed method. In 2017, Yu and Zhou [42] proposed a CF recommendation method based on user interest clustering, which computes the interest degree of the user and divides the users into clusters and uses the CF method to recommend movies. The experimental results show that the proposed method effectively enhances data sparsity and recommendation accuracy. In 2018, David and Maria [43] added a clustering process to a critique-based recommender and defined a cognitive user preference model based on the preferences received by the user. Multiple experiments showed that using clustering in a critique-based recommender leads to an improvement in their recommendation efficiency. Chen et al. [44] proposed an attentive collaborative filtering (ACF) model by introducing two levels of attention mechanisms into the hidden factor model. ACF is a multilayer perceptron, including two levels of attention mechanism. An item-level attention mechanism is used to describe the degree to which different items match the user’ interest, and a component-level attention mechanism is used to capture the most characteristic features of users in multimedia auxiliary information.

In the above example, by introducing deep learning technology in the behavior-based recommendation system, a non-linear feature transformation is introduced to learn the implicit representation of users and items, which has better performance. However, these methods do not address the data sparseness problem of the traditional CF method and the cold start problem of new users and new items.

4) SPATIOTEMPORAL-BASED RECOMMENDATION SYSTEM

The temporal and spatial recommendation system is based on the traditional recommendation system. It integrates the temporal and spatial features, converts the temporal and spatial information into trajectory data, and can deeply dig out the trend of users’ interests and preferences changing over time, which plays a positive role in improving the accuracy of recommendations.

Manotumruksa et al. [45] have proposed a novel situational attention loop architecture to solve the problem of different contexts havin different effects on user preferences. The architecture uses the feedback sequence and context information associated with the sequence to capture the user’ dynamic preferences. The loop architecture includes two types of gating mechanisms. One is the context attention gate, which controls the influence of ordinary context on the user’ context preferences. The other is a gate based on space-time, which controls the effects of hidden state from the previous check-in based on the transition context.
Hidasi et al. [46] adopted GRU to capture the dependencies between behaviors in the conversation. At each point in time, the mode’ input is the one-hot encoding of the currently clicked item and then compressed into a low-dimensional continuous vector through an embedding layer. In the middle are multiple basic GRU layers and a forward layer. The output layer uses softmax and other methods to calculate the click probability of each item. To achieve parallel computing, the research involved stitching different conversations and then using mini-batch processing. Liu et al. [47] studied the behavior prediction problem in location social networks. They used RNNs to capture the dependencies between sequential behaviors, thereby predicting the behavior at the next moment based on the user’ historical behavior sequence. Yang et al. [48] used an RNN to model the sequence movement pattern of users in a location social network, and simultaneously, integrated the social relationship of users to make location recommendations. Specifically, the model comprise two parts: the construction of a social network and the generation of a moving trajectory. Firs, a social network is built using the network embedding method. Then, the user’ access preferences, the influence of the user’ friends, and the user’ short-term sequence scenario, and long-term sequence scenario are used to jointly model the behavior sequence generation. To characterize the two kinds of context information, RNN and GRU are used to capture the sequence correlation in the user’ mobile mode in the short-term and long-term scenarios, respectively. Finally, the objective function is established by combining the construction of the social network and the generation of the moving trajectory, and the model is trained using the stochastic gradient descent method.

In the spatiotemporal recommendation system, integrating the user’ behavior sequence patterns and geographic location information into the recommendation system can help alleviate the data sparsity and cold start problems of the recommendation system and help enhance the quality of the recommendation system, especially suitable for transportation networks.

5) HYBRID RECOMMENDATION SYSTEM
UserCF’s recommendation results can recommend popular items with a wide range of interests but lack personalization. Conversely, ItemCF can find long-tailed items that are of personal interest to users but lacks diversity. Furthermore, a text-based recommendation system can solve sparse user behavior data and the cold start of new users but rely on the user and item features. Matrix factorization can automatically mine user and item features but fails to explain the recommendation results. Each recommendation method has its advantages and disadvantages and complements each other. Consequently, most recommendation systems are based on multiple recommendation algorithms. A hybrid recommendation strategy is often adopted in the real-world recommendation systems, which implies that the most suitable recommendation strategy is applied to specific scenarios. The purpose of a hybrid recommendation strategy is to use each recommendation algorithm’s advantages, learn from each other, and provide users with better recommendations. The basic principle is based on avoiding or complementing each recommendation strategy’s deficiencies by combining various recommendation algorithms.

Degemmis et al. [49] used WordNet to build a user profile based on semantics using machine learning and text classification algorithms, which contained user preference semantic information, not just keywords. The experimental results show that the method, based on the configuration file, significantly enhance the recommendation accuracy. In 2013, Lee et al. [50] extended the TSPR algorithm and proposed a hybrid recommendation algorithm PathRank that adapts to different application scenarios. The algorithm adds content or contextual information related to items and users based on the user–item bipartite graph. The nodes and associated edges are walked on the expansion graph using the enhanced random walk algorithm, which generates a mixed recommendation list based on various factors. Chen et al. [51] proposed a two-stage clustering recommendation algorithm based on the graph summary method and text-based similarity based on the data sparseness and cold start problems that often exist in social networks. Data captured by the Weibo open platform was used to verify the algorithm’s effectiveness and recommendation accuracy. Hsieh et al. [52] used metric learning to learn the similarity between data, combined metric learning with C and proposed a collaborative metric learning method. The study pointed out that the inner product operation does not have a distance. Furthermore, the transitivity of similarity is not a reasonable distance measure, which introduces inaccurate similarity learning in multimedia recommendations.

The above examples show that hybrid recommendation can effectively mitigate data sparsity and cold start problems to a certain degree. The next step is to build a unified hybrid recommendation framework for all data.

C. OUTPUT LAYER
The output layer is the third layer of the recommendation system’s hierarchical classification model based on heterogeneous features. This layer uses the inner product, softmax, and similarity calculation methods (Pearson correlation coefficient, cosine similarity, adjusted cosine similarity, and binary similarity) to generate the item recommendation list using the recommendation method selected by the second layer.

IV. EVALUATION INDEXES
There are multiple indexes for evaluating recommendation systems with varying application scenarios, which can objectively describe a recommendation system. Common evaluation indexes include mean absolute error (MAE) and root-mean-square error (RMSE), precision and recall, receiver operating characteristics (ROC) curve, mean reciprocal rank (MRR), hit ratio (HR), and coverage. Data sets are usually divided into training set $E^T$ and test set $E^P$ to evaluate the recommendation algorithm’s performance. We treat the training
set as information about known users and the test set as users who need to be recommended. Different recommendation algorithms need to select the appropriate recommendation index according to the user situation.

A. MAE AND RMSE

MAE and RMSE are common indicators for verifying recommendation systems’ accuracy. A small MAE value indicates a more accurate prediction and higher recommended accuracy. Assuming that a recommendation algorithm is used to predict the target user’s rating value for K items of possible interest is \{p_1, p_2, \ldots, p_k\} and the user’s actual degree of interest for K items is \{r_1, r_2, \ldots, r_k\}, then MAE is calculated as follows:

\[
MAE = \frac{\sum_{i=1}^{k} |p_i - r_i|}{k}.
\]

RMSE is an improvement on MAE that further amplifies the error between the user’s actual rating and the system’s predicted rating. Since RMSE is sensitive to prediction scores’ outliers, RMSE better reflects the recommendation system’s accuracy.

\[
RMSE = \sqrt{\frac{\sum_{i=1}^{k} (p_i - r_i)^2}{k}}.
\]

B. PRECISION, RECALL, AND F1

Accuracy rate refers to the proportion of items selected by users in recommendation results. \(R(u)\) represents the list of recommendations generated by the recommendation system for each user on the test set. \(T(u)\) represents all items of interest for each user on the test set.

\[
\text{Precision} = \frac{\sum_{u \in U} |R(u) \cap T(u)|}{|R(u)|}.
\]

Recall rate refers to the proportion of items selected by users after system recommendation.

\[
\text{Recall} = \frac{\sum_{u \in U} |R(u) \cap T(u)|}{\sum_{u \in U} |T(u)|}.
\]

It is biased to evaluate the recommendation system’s performance based only on the accuracy or recall rate because there exists a contradiction between accuracy and recall rate. An increase in the number of items on the recommendation list reduces the accuracy and increases the recall rate and vice versa. Therefore, the balance between accuracy and recall rate generates an F1 score, which is a relatively comprehensive evaluation index.

\[
F1 = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}.
\]

C. ROC CURVE

The area under the ROC curve represents the area under the ROC curve (AUC). ROC is generally used to measure the ranking capability of the recommendation algorithm model. AUC is usually used for comparison because it is challenging to directly compare ROC curves. The higher the AUC value, the better the performance of the model. Assuming that there are M positive and N negative cases in a data set, \(\text{rank}(i)\) represents the ordinal number of sample i after ranking. The formula for computing AUC value is as follows:

\[
AUC = \frac{\sum_{i \in \text{positive}} \text{rank}(i) - \frac{M \times (M+1)}{2}}{M \times N}.
\]

D. MRR

The correct search result value ranks in the search result used to evaluate the performance of the search system.

\[
\text{MRR} = \frac{1}{Q} \sum_{i=1}^{Q} \frac{1}{\text{rank}_i}.
\]

where \(|Q|\) is the number of users, \(\text{rank}_i\) is the arrangement position of the first item in the ground truth result of the recommendation list for the i user.

E. HR

In the top-k recommendation, HR is a common index used to measure the recall rate. The calculation formula is

\[
HR@K = \frac{\text{NumberOfHits@k}}{GT}.
\]

The denominator is all test sets, and the numerator represents the sum of the number of test sets in each user’s top-k list.

F. COVERAGE

Coverage refers to the proportion of items recommended by the recommendation system to the total items. Assuming that the set of users is \(U\), the recommendation system recommends an item list \(R(u)\) of length N to each user. The coverage formula is

\[
\text{Coverage} = \frac{\bigcup_{u \in U} R(u)}{|U|}.
\]

V. OPEN-SOURCE IMPLEMENTATION

This section summarizes the open-source implementation of the hierarchical classification model based on the heterogeneous features reviewed in the survey to avoid duplication of research and improve the efficiency of research. Hyperlinks to the source code of recommendation methods are provided in Table 1.

VI. EXPERIMENTAL COMPARISON

The current evaluation indicators for measuring the performance of the recommendation method include RMSE, Precision, Recall, AUC, MRR, HR and Coverage. The experimental results of common recommendation methods are shown in Table 2.

VII. THE ADVANTAGES AND DISADVANTAGES OF EACH RECOMMENDATION METHOD

This section analyzes the advantages and disadvantages of the recommendation methods reviewed in this article, as shown in Table 3.
TABLE 1. A Summary of Open-source implementation.

| classification        | Model                                                                 | Github Link                                                                 |
|-----------------------|------------------------------------------------------------------------|----------------------------------------------------------------------------|
| Graph-based recommendation system | STAR-GCN:Stacked and reconstructed graph convolutional networks for recommender systems[10] | https://github.com/jennyzhang0215/STAR-GCN                                |
|                       | Dressing as a whole: Outfit Compatibility Learning based on Node-wise Graph Neural Networks[11] | https://github.com/CRIPAC-DIG/NGNN                                         |
|                       | Neural Networks for social Recommendation[12]                         |                                                                            |
|                       | Knowledge graph attention network for recommendation[14]              |                                                                            |
|                       | Graph convolutional matrix completion[15]                             |                                                                            |
|                       | Citation recommendation with a content-sensitive deepwalk based approach[17] |                                                                            |
|                       | A general graph-based for top-N recommendation using content, temporal and trust information[20] |                                                                            |
|                       | Graph convolutional policy network for goal-directed molecular graph generation[24] |                                                                            |
| Text-based recommendation system | A LSTM based model for personalized context-aware citation recommendation[27] | https://github.com/tebesu/NeuralCitationNetwork                           |
| Behavior-based recommendation system | User-based and Item-based Collaborative Filtering algorithm[32][34] | https://github.com/ChangUk/pyCollaborativeFiltering                        |
|                       | CNN-based collaborative filtering [37]                               |                                                                            |
|                       | Session-based recommendations with recurrent neural networks[46]     |                                                                            |
|                       | Predicting the next location: A recurrent model with spatial and temporal contexts[47] | https://github.com/yongyuan/SRNN                                         |
|                       | A neural network approach to jointly modelling social networks and mobile trajectories[48] | https://github.com/thumlp/JNMT                                             |
| Hybrid recommendation system | A content-collaborative recommender that exploits WordNet-based user profiles for neighbourhood formation [49] | https://github.com/groveco/content-engine                                  |
|                       | Collaborative metric learning[52]                                    |                                                                            |

TABLE 2. Recommended system experiment comparison.

| Classification | Ref | name | RMSE | Precision | recall | AUC | MRR | HR | coverage |
|---------------|-----|------|------|-----------|--------|-----|-----|----|----------|
| Graph-based recommendation system | [9] | Pinsage | 0.895 | 0.7701 | 0.96 | 0.591 | 0.67 |  |
| [10] | STACR-GCN | 0.9794 | 0.284 | 0.535 |
| [11] | NGNN | 0.905 | 0.149 |  |
| [12] | GraphRec | 0.905 | 0.149 |  |
| [13] | GC-SAN | 0.905 | 0.149 |  |
| [14] | KGAT | 0.905 | 0.149 |  |
| [15] | GC-MC | 0.905 | 0.149 |  |
| [16] | CR-CA | 0.905 | 0.149 |  |
| [17] | Node2vec | 0.905 | 0.149 |  |
| [18] | IDGCF | 0.905 | 0.149 |  |
| [19] | CO-Attention | 0.905 | 0.149 |  |
| [20] | LSTM-CAV | 0.905 | 0.149 |  |
| [21] | CNNNN | 0.905 | 0.149 |  |
| [22] | DL | 0.905 | 0.149 |  |
| [23] | SA-UserCF | 0.905 | 0.149 |  |
| [24] | DBNCF | 0.905 | 0.149 |  |

VIII. RESEARCH TREND FOR RECOMMENDATION SYSTEMS

With the deepening of the era of big data, the recommendation system, as an effective tool for filtering information, exists based on massive data mining. It can solve the problem of information overload by providing personalized recommendations. The content of user needs has become the focus of industry and academia, which will be extended further in future research. The following summarizes seven possible development trends in the future.

A. ATTENTION MECHANISM COMBINED WITH RECOMMENDATION SYSTEM

Attention mechanism has achieved great success in computer vision, natural language processing, and other fields. Attention mechanism has recently been applied in the hashtag, article, multimedia, and citation recommendations. However,
| Classification                      | Ref   | Advantages                                                                 | Disadvantages                                                                 |
|-------------------------------------|-------|----------------------------------------------------------------------------|-------------------------------------------------------------------------------|
| Graph-based recommendation system   | [9]   | It has low complexity, good scalability, robustness, and convergence.      | Its scope of adaptation is limited.                                            |
|                                     | [10]  | It enhances the recommended performance and effect under cold start conditions and reduces the space complexity of the model. |                                                                                |
|                                     | [11]  | It can jointly model outfit compatibility from multiple modalities.          | It cannot handle heterogeneous graphs with diverse node types.                 |
|                                     | [12]  | It can differentiate tie strengths by considering heterogeneous strengths of social relations. | It lacks research on model personalization compatibility.                      |
|                                     | [15]  | It can alleviate performance bottlenecks related to the cold start problem. | It is unsuitable for dynamic network recommendation.                          |
|                                     | [22]  | It alleviates data sparsity, cold start, and scalability problems to a certain extent. | It has poor scalability.                                                      |
|                                     |       | The advantage of the graph-based recommendation system is that it has good interpretability with the aid of graphical methods. |                                                                                |
| Text-based recommendation            | [27]  | It uses the personalized context-aware citation recommendation, which is more effective. | It cannot facilitate automatic parameter setting.                              |
|                                     |       |                                                                                |                                                                                |
|                                     | [29]  | It outperforms support vector machine (SVM) and random forest (RF) from the recommendation accuracy viewpoint. | The overall performance needs to be further enhanced.                         |
|                                     |       |                                                                                |                                                                                |
|                                     | [30]  | It enhances the problem of sparsity.                                         | Currently, the scope of application of this method is limited.                |
|                                     |       |                                                                                |                                                                                |
| Behavior-based recommendation system | [7]   | Mining movie reviews and calculating sentiment tendencies can achieve better recommendation quality. |                                                                                |
|                                     | [32]  | It has better scalability.                                                   |                                                                                |
|                                     | [39]  | It effectively reduces the MAE value and enhances recommendation accuracy.   |                                                                                |
|                                     |       | Simultaneously, it enhances the coverage and recommendation system’s ability to discover a product’s long tail. |                                                                                |
|                                     | [40]  | It enhances the diversity and coverage of doctor recommendations and reduces information loss or decreases recommendation times. |                                                                                |
|                                     | [42]  | It effectively enhances data scarcity and real-time problem and significantly improves accuracy. |                                                                                |
|                                     |       |                                                                                |                                                                                |
| Spatiotemporal-based recommendation system | [8]  | It couples time and distance gates to enhance ST-LSTM efficiency.          | The model lacks textual description content.                                   |
|                                     |       | It is not very sensitive to the window width and dimensionality and can converge in a satisfactory number of iterations. | It is unsuitable for handling long-time series problems.                       |
|                                     | [48]  | It provides a promising approach to characterize heterogeneous data types in LBSNs and can enhance the sparsity problem to some extent. | It does not consider the GPS information and semantic information.            |
| Spatiotemporal-based recommendation system |       |                                                                                |                                                                                |
|                                     | [49]  | It enhances sparsity, scalability, and the lack of transparency problems.    |                                                                                |
|                                     | [50]  | It produces more diverse and effective recommendation results.              | It fails in identifying domain entities and technical terms.                  |
|                                     |       | This method has a better recommendation effect in the case of sparse data and cold start, and the efficiency of online recommendation is also good. | It is prone to data sparsity problems.                                        |
|                                     |       |                                                                                | This method lacks a feedback mechanism and does not consider the issue of parallel processing. |
|                                     | [52]  | It achieves significant speedup for Top-k recommendation tasks using off-the-shelf. | It does not consider user features. The model’s representational capacity and interpretability were poor. |

The advantages of the hybrid recommendation system include no cold start problem, not restricted by the data sparsity problem, and the effective use of auxiliary data to enhance the recommendation performance. The disadvantage is the inability to adequately express the features of heterogeneous data. Although there are many combinations of methods, they are not always effective in a specific problem.
research in this area is still relatively sparse, and there is a need for further research in the future.

B. INTERPRETABILITY OF RECOMMENDATION SYSTEMS

Studies on recommendation systems should consider the interpretability of the recommendation results to enhance the user’s understanding of the recommendation results.

C. CROSS-DOMAIN INFORMATION RECOMMENDATION FUSION

With the continuous deepening of big data technology, cross-domain data can enhance the discovery of users’ personalized preferences. Building cross-domain information fusion recommendations will be one of the important directions for future academic research.

D. GROUP RECOMMENDATION

The application of recommendation systems is becoming more extensive, and the object of recommendation has evolved from single-user and personalized recommendations to group recommendation for multiple users. Existing recommendation systems rarely consider group recommendation. Therefore, the implementation of group recommendations should be considered in future studies.

E. COMBINATION OF SENTIMENT ANALYSIS AND RECOMMENDATION SYSTEM

Most of the existing recommendation systems extract sentiment information of users from textual comments. With the development of big data and internet technology, there is an exponential increase in the volume of texts, videos, voices, and images containing users’ opinions. Future research can potentially explore how to effectively extract opinions and identify emotional orientation from the massive unstructured or semi-structured user comment data and apply it to information recommendation.

F. COMBINATION OF COGNITIVE MODELS AND RECOMMENDATION SYSTEMS

Although some articles apply cognitive reasoning to recommendation systems, there are few studies in this area. In 2020, Angulo et al. [53] proposed bridging cognitive models and recommender systems and reviewed the application of four cognitive models. Combining causal reasoning and context-aware technology from the comment information feedback from users. Integrating the cognitive model into information recommendation is an interesting and difficult area to be explored in future research.

G. COMBINATION GRAPH CLUSTERING AND RECOMMENDATION

Although clustering methods have been studied in recommendation systems research for many years, there are relatively few studies on graph clustering methods. In 2019, Peng et al. [54]–[56] proposed some novel clustering methods. Applying graph clustering methods to recommendation systems is the future development trend in this field.

IX. CONCLUSION

With the rapid development of the Internet, there is a steady increase in “information overload” is heating up year by year. Recommendation systems are increasingly playing an important role in the digitization of various fields. In this paper, we applied a hierarchical design based on heterogeneous input features to recommendation systems to learn text, behavior, graph structure, and spatiotemporal features from massive data. Users generate an effective recommendation list, which enhances the performance and accuracy of the recommendation system. Based on the analysis of existing research and progress in the area of recommendation systems, this paper systematically classifies existing recommendation systems and proposed a new classification method. Furthermore, we introduced the classification model design of recommendation systems, which is divided into three layers: feature input, feature learning, and output layers. The feature learning layer includes commonly used recommendation methods. We divided the recommendation methods into graph-based, text-based, behavior-based, spatiotemporal-based, and hybrid recommendation systems based on the type of input features. The evaluation index, open-source implementation, relative merits, experimental comparison, and future development direction of recommendation systems were discussed.

REFERENCES

[1] A. Da’u and N. Salim, “Recommendation system based on deep learning methods: A systematic review and new directions,” Artif. Intell. Rev., vol. 53, no. 4, pp. 2709–2748, Apr. 2020.
[2] W. Z. Zhou, D. Cao, and Y. F. Xu, “A survey of recommendation systems,” J. Hebei Univ. Sci. Technol., vol. 41, no. 2, pp. 76–87, Sep. 2020.
[3] P. Resnick and H. R. Varian, “Recommender systems,” Commun. ACM, vol. 40, no. 3, pp. 56–58, 1997.
[4] R. Burke, “Hybrid recommender systems: Survey and experiments,” User Model. User-Adapted Interact., vol. 12, no. 4, pp. 331–370, Nov. 2002.
[5] J. B. Schafer, J. A. Konstan, and J. Kied, “E-commerce recommendation applications,” Data Mining Knowl. Discovery, vol. 28, no. 6, pp. 107–121, Jan. 2001.
[6] J. Han, Z. Zhang, N. Cunmins, and B. Schuller, “Adversarial training in affective computing and sentiment analysis: Recent advances and perspectives [review article],” IEEE Comput. Intell. Mag., vol. 14, no. 2, pp. 68–81, May 2019.
[7] M. Lei and M. Zhu, “Applications of sentiment analysis in movie recommendation system,” Comput. Eng. Appl., vol. 52, no. 10, pp. 59–63, Jun. 2016.
[8] P. P. Zhao, H. F. Zhu, and Y. C. Li, “‘Where to go next: A spatio-temporal LSTM model for next POI recommendation,” in Proc. AAAI Conf. Artif. Intell., USA, 2018, pp. 1–8.
[9] R. Ying, R. He, K. Chen, P. Eckambatchai, W. L. Hamilton, and J. Leskovec, “Graph convolutional neural networks for Web-scale recommender systems,” in Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining, London, U.K., Jul. 2018, pp. 974–983.
[10] J. N. Zhang, X. J. Shi, and S. L. Zhao, “STAR-GCN: Stacked and reconstructed graph convolutional networks for recommender systems,” Inf. Retr., vol. 32, no. 21, pp. 28–35, May 2019.
[11] Z. Cui, Z. Li, S. Wu, X.-Y. Zhang, and L. Wang, “Dressing as a whole: Outfit compatibility learning based on node-wise graph neural networks,” in Proc. World Wide Web Conf. (WWW), Feb. 2019, pp. 307–317.
[12] W. Fan, Y. Ma, Q. Li, Y. He, E. Zhao, J. Tang, and D. Yin, “Graph neural networks for social recommendation,” in Proc. World Wide Web Conf. (WWW), Feb. 2019, pp. 417–426.

[13] C. Xu, P. Zhao, Y. Liu, V. S. Sheng, J. Xu, F. Zhuang, J. Fang, and X. Zhou, “Graph contextualized self-attention network for session-based recommendation,” in Proc. 20th Int. Joint Conf. Artif. Intell., Aug. 2019, pp. 1–7.

[14] X. Wang, X. He, Y. Cao, M. Liu, and T.-S. Chua, “KGA: Knowledge-edge graph attention network for recommendation,” in Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, Jul. 2019, pp. 950–958.

[15] R. D. Van Berg, N. Kipf, and M. Welling, “Graph convolutional network completion,” Stat., vol. 1050, no. 7, pp. 79–86, Jun. 2017.

[16] J. You, R. Ying, X. Ren, W. L. Hamilton, and J. Leskovec, “GraphRNN: Generating realistic graphs with deep auto-regressive models,” in Proc. 35th Int. Conf. Mach. Learn., Jan. 2018, pp. 462–463.

[17] L. Guo, X. Cai, H. Qin, Y. Guo, F. Li, and G. Tian, “Citation recommendation with a content-sensitive DeepWalk based approach,” in Proc. Int. Conf. Data Mining Workshops (ICDMW), Nov. 2019, pp. 1–6.

[18] P. Enríquez, G. Rizzo, R. Troncy, E. Baralis, M. Osella, and E. Ferro, “Knowledge graph embedding with node2vec for item recommendation,” in Proc. 15th Eur. Semantic Web Conf., Feb. 2018, pp. 117–120.

[19] T. H. Haveliwala, “Topic-sensitive pagerank: A context-sensitive ranking algorithm for Web search,” IEEE Trans. Knowl. Data Eng., vol. 15, no. 4, pp. 784–796, Jul. 2003.

[20] O. Nzeko, A. J. Nzekon, and T. Maurice, “A general graph-based for top-N recommendation using content,” Temporal Trust Inf., vol. 27, no. 5, pp. 6–34, Jun. 2019.

[21] H. Li, D. Y. Wu, and Q. Gao, “Construction of user clustering movie recommendation system based on bipartite graph networks,” Netw. Inf. Eng., vol. 20, no. 18, pp. 86–87, May 2016.

[22] Y. Cao and W. Li, “Collaborative filtering recommendation algorithm based on trust and graph clustering,” Inf. Technol., vol. 10, no. 2, pp. 76–82, Nov. 2019.

[23] L. Li and L. J. Hong, “Weighted bipartite graph recommendation algorithm based on random forest modification,” Comput. Eng. Softw., vol. 12, no. 1, pp. 110–115, Jan. 2018.

[24] B. You and R. Y. Liu, “Graph convolutional policy network for goal-directed molecular graph generation,” in Proc. 32nd Conf. Neural Inf. Proc. Syst. (NIPS), Montreal, QC, Canada, Sep. 2018, pp. 11–35.

[25] K. Do, T. Tran, and S. Venkatesh, “Graph transformation policy network for chemical reaction prediction,” in Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, Jul. 2019, pp. 750–760.

[26] Q. Zhang, J. Wang, H. Huang, X. Huang, and Y. Gong, “Hashtag recommendation for multimodal microblog using co-attention network,” in Proc. 26th Int. Joint Conf. Artif. Intell., Melbourne, VIC, Australia, Aug. 2017, pp. 3420–3426.

[27] L. Yang, Y. Zheng, X. Cai, H. Dai, D. Mu, L. Guo, and T. Dai, “A LSTM based model for personalized context-aware citation recommendation,” IEEE Access, vol. 6, pp. 59618–59627, 2018.

[28] H. Lee, Y. Ahn, H. Lee, S. Ha, and S.-G. Lee, “Quote recommendation in dialogue using deep neural network,” in Proc. 39th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR), 2016, pp. 957–960.

[29] W. Tan, K. Oku, and K. Kawagoe, “Graph-based recommendation using content,” Int. J. Knowl. Eng., vol. 2, no. 3, pp. 128–131, 2016.

[30] T. Bansal, D. Belanger, and A. McCallum, “Ask the GRU: Multi-task learning for deep text recommendations,” in Proc. 10th ACM Conf. Recommender Syst., Boston, MA, USA, Sep. 2016, pp. 107–114.

[31] J. Wang, “A collaborative filtering systems based on personality information,” in Proc. Int. Conf. Inform. Comput. Eng. Conf. (ICICEC), Jan. 2015, pp. 2352–2356.

[32] Z.-D. Zhao and M.-S. Shang, “User-based collaborative-filtering recommendation algorithms on Hadoop,” in Proc. 3rd Int. Conf. Knowl. Discovery Data Mining, Jan. 2010, pp. 478–481.

[33] E. Baralis and P. Garza, “I-prune: Item selection for associative classification,” Int. J. Intell. Syst., vol. 27, no. 3, pp. 279–299, Mar. 2012.
HUI WANG was born in China. She is currently pursuing the Ph.D. degree in control science and engineering with the Zhejiang University of Technology. She is also a Lecturer with the Jiangxi University of Science and Technology. Her main research interests include link prediction, recommendation systems, deep learning, AI, and big data. She is a member of the China Computer Federation (CCF).

ZICHUN LE was born in China. She received the bachelor’s degree in optical instrumentation from Zhejiang University, in 1987, and the Ph.D. degree in optical engineering from the Changchun Institute of Optics and Fine Mechanics, Chinese Academy of Sciences, in 1997. She has rich international cooperative research experience. She was also with the University of Oxford, U.K., Bielefeld University, Germany, The University of Iowa, USA, and Chonnam National University. She is currently a Professor with the Zhejiang University of Technology, the Director of the Joint International Research Laboratory of Optoelectronic Information Technology of Zhejiang Province, and the Head of the Department of Optical Engineering. She is the author of more than 100 scientific works, including six monographs. She holds more than 90 patents. Her research interests include optical communications, optics and photonics, and signal processing.

XUAN GONG was born in China. He received the bachelor’s degree from Nanhua University, in 2001, and the master’s degree from Hangzhou Dianzi University, in 2010. He is currently pursuing the Ph.D. degree in control science and engineering with the Zhejiang University of Technology. His main research interests include AI, machine learning, deep learning, and computational vision. He is a member of the China Computer Federation (CCF).

* * *