Review Article

A Survey of Long-Tail Item Recommendation Methods

Jing Qin

Northeastern University, China

Correspondence should be addressed to Jing Qin; annyproj@126.com

Received 9 May 2021; Revised 30 October 2021; Accepted 9 November 2021; Published 29 November 2021

Academic Editor: Danfeng Hong

Copyright © 2021 Jing Qin. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Recommender systems represent a critical field of AI technology applications. The core function of a recommender system is to recommend items of interest to users, but if it is only user history-based (purchasing or browsing data), it can only recommend similar products to a user, which makes the user feel fatigued (creating so-called “Information Cocoons”). Besides, transaction data (purchasing or browsing data) in various fields usually follow Pareto distributions. Accordingly, 20% of products are purchased or viewed a greater number of times (short-head items), while the remaining 80% of products are purchased or viewed less frequently (long-tail items). Using the traditional recommendation method, considering only the accuracy of recommendations, the coverage rate is relatively low, and most of the recommended items are short-head items. The long-tail item recommendation method not only considers the recommendation of short-head items but also considers recommending more long-tail items to users, thus improving the coverage and diversity of the recommendation results. Long-tail item recommendation research has become a frontier issue in recommendation systems in recent years. While the current research paper is still scarce, there have been related research achievements in top-level conferences in the field of computers, such as VLDB and IJCAI. Due to the fact that there is no review literature in this field, to allow readers to better understand the research status of the long-tail item recommendation method, this paper summarizes the progress of the research on long-tail item recommendation methods (from clustering-based, which began in 2008, to deep learning-based methods, which began in 2020) and the future directions associated with this research.

1. Introduction

In 1992, the concept of a recommender system was first proposed by Goldberg et al. [1]. Recommender systems have since become a significant way for people to obtain information. Different from search engines, recommender systems actively recommend items that users may be interested in according to their preferences. The content-based recommendation method and collaborative filtering recommendation method [2, 3] are classic methods in the recommender system. Machine learning and deep learning have great advantages in learning the inherent laws and representation levels of sample data and have made many research achievements in image classification [4–7], object detection [8–11], speech recognition [12, 13], and emotion recognition [14]. Therefore, researchers combine machine learning, deep learning, knowledge graph, and other technologies in these basic methods, allowing recommender systems to be widely used in news, tourism, e-commerce, and other fields.

Due to the sparse rating dataset, products with few rates or browses are excluded from the recommendation results. Taking the ratings in the MovieLens 1M dataset [15] as an example, the distribution of movie ratings is shown in Figure 1.

As can be seen from Figure 1, most movies have less than 100 ratings, and many movies have only 1 rating. Anderson [16] conducted the first analysis of the long-tail item recommendation research in commerce. Due to the large number of long-tail items, the coverage and diversity of the recommended results are low [17]. A long-tail item recommendation recommends more long-tail items to users and improves the recommendation results' coverage and diversity rate [17]. Hervas-Drane [18] suggests that improving the recommendation of long-tail items in the field of e-commerce can improve the profits of enterprises. As in the field of e-commerce, it will be a win-win result to increase the recommendation rate of long-tail items. Most researches on long-tail item recommendation focus on improving the

FIGURE 1: The distribution of movie ratings.

diversity of recommendation results. The long-tail recommendation method proposed in [19] also considers the credibility of recommendation results. This paper summarizes the research papers on the long-tail recommendation method since 2008. The research papers were mainly retrieved from DBLP, IEEE, and ACM. We divide the long-tail item recommendation methods proposed in recent years into clustering-based, graph-based, multiobjective optimization-based, deep learning-based, and other long-tail recommendation methods.

The main contributions of this paper are as follows:

- (1) To the best of our knowledge, this paper is the first review of long-tail item recommendation methods
- (2) We classify the long-tail item recommendation method into 5 categories (clustering-based, graph-based, multiobjective optimization-based, deep learning-based, and other long-tail item recommendation methods)
- (3) We propose future research directions on the long-tail item recommendation problem

The organization of the rest of this paper is as follows: Section 2 introduces the problem description and general framework of the long-tail item recommendation method. Section 3 reports the research progress on the long-tail item recommendation method, according to the timelines of recommendation methods. Section 4 analyzes the future directions in the long-tail item recommendation research, and Section 5 summarizes the paper.

2. Problem Description and General Framework

This section will introduce the problem description and general framework of long-tail item recommendation methods.

2.1. Problem Description. The long-tail items refer to items with fewer ratings in the user rating dataset of the long-tail recommendation method. The general problem definition for long-tail item recommendation is as follows:

Given a target user *targetUser*, recommend N items to meet the *targetUser* preference, and the items are more likely to be long-tail items [17].

In addition, in our previous work [20], to allow products to flexibly adjust the recommended quantity of long-tail items, the recommended target is as follows: the quantity of long-tail items in the user's recommendation list meets the specified probability p . For example, $p = 0.2$, so the quantity of long-tail items accounts for 20% of the recommendation list.

In short, unlike other recommendation methods, the long-tail item recommendation method takes into account how long-tail items appear in the user's recommendation list.

2.2. The General Framework. In long-tail item recommendation research, the general framework usually consists of three modules: a data processing module, recommendation module, and evaluation module. The general framework is shown in Figure 2. The following describes the functions of each module in the framework.

(1) Data processing module

In the research on the long-tail item recommendation method, the dataset should conform to the long-tail distribution characteristics (e.g., MovieLens [17, 21, 22], Book-Crossing [20, 23], Amazon [24], and Douban [17, 19]). The data processing module is mainly used to represent and analyze the data in the dataset. To facilitate the use of data in the recommendation method, the user rating data are expressed as a matrix and graph. Feature data used embedding expression in the data processing module. In addition, the long-tail items are related to the rating times of the items. Thus, the item's rating times and the distribution of the long-tail items are also counted in the data processing module.

(2) Recommendation module

The recommendation method module includes clustering-based methods, graph-based methods, multiobjective optimization-based methods, deep learning-based methods, and other long-tail item recommendation methods. The recommendation method using clustering or multiobjective optimization methods needs matrix representation and statistical data of long-tail items from the data processing module. The recommendation method using the graph-based method needs graph representation in the data processing module and statistical data on long-tail items from the data processing module. The recommendation method using deep learning-based methods may need the data represented by being embedded in the data processing module. In the ranking module, according to the results of calculation or training of the recommended methods, the main ranking methods include similarity sorting, the concentration index [25], the popularity and the number of recommended items [20], and a mixed ranking of various methods [26].

FIGURE 2: The general framework for long-tail item recommendation.

(3) Evaluation module

Unlike other recommendation methods, the evaluation metrics of the long-tail item recommendation method include the coverage rate, diversity, and recommendation effect of long-tail items, in addition to accuracy evaluation metrics (e.g., precision, recall, MAP, and MAE). The coverage and diversity metrics both evaluate the mining ability of long-tail items in recommendation methods, but they are not intuitive enough. The evaluation metric of the long-tail item recommendation effect also includes the popularity distribution of the recommended item [27] metric.

2.3. Long-Tail Item Recommendation and Cold-Start Problem. The cold-start problem refers to the recommendation problem of users or items, without any interactive information. It includes the user cold-start problem and the new item cold-start problem [28]. The long-tail item recommendation method solves the problem of recommending more long-tail items that meet the user's preferences to users, and the popularity of items is more evenly distributed in the recommendation results. Long-tail item recommendation can alleviate the cold-start problem of new users and can produce diversified results. For example, Li et al. [27] proposed a long-tail item recommendation method and solved the cold-start problem of new users. However, when recommendation methods solve the cold-start problem, it may not be possible to solve the problem of long-tail item recommendation, for example, recommending hot items to new users. Therefore, solving the cold-start problem cannot solve the long-tail item recommendation problem.

3. Research Progress of Long-Tail Item Recommendation Methods

According to the timeline, the main methods and related references used in long-tail item recommendation researches are shown in Table 1.

3.1. Clustering-Based Long-Tail Item Recommendation Methods. In the research of recommendation methods, clustering refers to the classification of similar items or similar users. Clustering methods are used to alleviate the problem of data sparsity and improve the accuracy of recommendation [29]. Park and Tuzhilin [29] first proposed the clustering-based long-tail item recommendation method. Park and Tuzhilin [29] proposed EI (Each Item), TC (Total Cluster), and CT (Clustered Tail) long-tail item recommendation methods. Park [23] proposed the AC (Adaptive Clustering) method based on the clustering method proposed in 2008 [29]. Grozin and Levina [30] applied the idea of AC and TC methods to cross-selling recommendations. Huang and Wu [24] proposed a long-tail item recommendation method that combines topic model-based methods with clustering methods.

Park and Tuzhilin [29] used the Expectation-Maximization (EM) [40] clustering method. TC refers to the method of dividing all items into different groups and building a rating prediction model for the items in each group. CT clusters long-tail items, divides them into different groups, and then builds a rating prediction model for each group. The division of long-tailed items is based on the rating times of items. For example, if the number of

TABLE 1: Classification of long-tail item recommendation methods.

Methods	Ref.
Clustering methods	[23, 24, 29, 30]
Graph methods	[17, 21, 22]
Multiobjective optimization methods	[19, 26, 31–33]
Deep learning methods	[20, 34–36]
Other methods	[25, 27, 37–39]

rating times of long-tail items is 30, items with more than 30 rating times are divided into short-head items, and others are considered long-tail items. Besides, a situation in which there are no short-head items in the experiment, that is, where there only long-tail items are included, is equivalent to using the TC clustering method. It can be seen from the experimental results that the CT method is superior to the EI method in most cases. While the accuracy of item rating prediction is improved by the clustering method in the experiment, the rating distribution is different in different datasets, and how to better divide short-head items and long-tail items is still a challenge.

The AC [23] method refers to clustering items according to their rating times. For items with fewer rating times, the AC method finds similar items and clusters them. For items with more rating times, the AC method uses the EI method [29]. The AC clustering method divides the clustering according to the total number of ratings used. For example, the total number of ratings for clustering is 100. If the number of ratings for items is higher than 100, it only uses the EI method, and clustering is considered when it is lower than 100. If the number of ratings for item A is 30, the clustering is completed until the total number of ratings for all items is higher than 100. From the experimental results in this paper, compared with the EI method, the AC method has a lower RMSE (Root Mean Square Error) metric, and compared with the TC method, the AC method has a lower popularity to recommend items (i.e., a stronger ability to recommend long-tail items). However, the challenge of the AC method is still how to find a suitable clustering standard.

Cross-sell recommendation is performed by using ARM (Association Rule Mining) [41]. The ARM method is effective in cross-selling recommendation [42], but the ARM method still cannot satisfy the mining of long-tail items. Therefore, based on the AC and TC clustering methods, Grozin and Levina [30] proposed three methods to measure the distance between objects, including category-based distance, session-based distance, and Prod2Vec-based distance [43]. The experimental results in this paper show that the NDCG (Normalized Discounted Cumulative Gain) metric of the method based on session distance is the best in AC and TC methods, as shown in Figure 3.

Additionally, Huang and Wu [24] proposed the long-tail product recommendation framework using the improved k-medoids [44] clustering method. The proposed framework, based on the topic model, mainly includes two components, one for extracting product profiles and the other for extracting similar products. Topic model-based methods are used in the components of product profile extraction. A similar

FIGURE 3: The NDCG on TC and AC in different distance measure methods [30].

TABLE 2: PMI score of four methods on two datasets [24].

Datasets	N	MaxEnt-LDA	DLDA	FSenExt	MaxEnt-BTM
Mobile electronics	5	1.15	1.2	1.7	1.84
	10	0.67	0.79	1.09	1.28
	20	0.44	0.4	0.68	0.83
Seattle	5	1.55	1.61	1.72	1.85
	10	0.88	0.9	1.19	1.37
	20	0.68	0.69	0.81	0.9

product extraction component is used the Ek-medoids (enhanced k-medoids) method. In terms of runtime, Ek-medoids have shorter runtime than k-prototype [45], MK-mean [46], Fk-prototype [47], and k-partitioning [44] methods. In terms of F1-score, the Ek-medoids method is also superior to the four methods compared. The product profile extraction model uses the MaxEnt-BTM model. The MaxEnt-BTM model is implemented in the LDA-based extended BTM model [48] and MaxEnt (Maximum Entropy) model [49]. Huang and Wu [24] used PMI (Point-Wise Mutual Information) score on two datasets to validate the effect of the topic model; the results are shown in Table 2. In Table 2, N denotes probable words number, MaxEnt-BTM model is proposed in this paper, and the three models compared with it are MaxEnt-LDA [49], DLDA [50], and FSenExt [51].

It can be seen from Table 2 that the PMI score of the MaxEnt-BTM model works best.

Huang and Wu [24] use the precision, diversity, and novelty metrics in [31] to compare LTC [29], AC [23], and MORS [31] methods. From the perspective of precision, the TopicRec method is only slightly better than other comparison methods, and the TopicRec method has an obvious improvement effect in diversity and novelty metrics.

The summary information of the above clustering-based long-tail item recommendation methods is shown in Table 3.

TABLE 3: The summary on clustering-based long-tail item recommendation methods.

Ref.	Datasets	Evaluation metrics	Proposed methods
[23]	MovieLens, BookCrossing	RMSE, MAE	AC (Adaptive Clustering)
[24]	Amazon, Airbnb	PMI score (Point-Wise Mutual Information), precision, F1-score, diversity, novelty	EK-medoids (enhanced k-medoids algorithm), MaxEnt-BTM, TopicRec
[29]	MovieLens, BookCrossing	RMSE, MAE	EI (Each Item), TC (Total Cluster), CT (Clustered Tail)
[30]	e-commerce data (not public dataset)	NDCG	Based on AC and TC methods, the effects of three distance measurement methods are compared

To sum up, the clustering method can effectively improve long-tail item recommendation. However, there are still some shortcomings associated with the application of clustering:

- (1) Only the rating times of items are considered in clustering [23], and the items' attributes and users' preferences are not considered
- (2) No consideration is given to the collocation between items

3.2. Graph-Based Long-Tail Item Recommendation Methods.

The graph-based method means that the interactive data between users and items is represented in the form of graphs, which enhances the availability of data [17]. The long-tail item recommendation method mainly involves a bipartite graph [17], tripartite graph [21], and extended tripartite graph [22]. Yin et al. [17] first proposed a long-tail item recommendation method based on the bipartite graph method. Then, Johnson and Ng [21] proposed a tripartite graph method based on the bipartite graph method. Luke et al. [22] proposed an extended tripartite graph method based on the first two methods.

Yin et al. [17] used a bipartite graph to express the relationship between users and items. The representation of the bipartite graph is shown in Figure 4. Yin et al. proposed four methods to realize long-tail item recommendation: the HT (Hitting Time), AT (Absorbing Time), AC1 (Absorbing Cost), and AC2 (topic-based Absorbing Cost) methods. The experimental results in this paper show that AC2 has the best effect on the recall metric ($AC2 > AC1 > AT > HT$). In the diversity metric, AC1 has the best performance ($AC1 > (AC2, AT) > HT$), and the values of AC2 and AT are the same. In addition, the popularity metric is also used to measure the recommendation effect of long-tail items in the experiment. AC1 has the lowest popularity in the MovieLens 1M dataset, while HT has the lowest popularity in the Douban dataset. In addition, the running time of the AC2 method in online case is calculated to be 0.52 seconds in this experiment.

Johnson and Ng [21] used the tripartite graph method [52] to represent the relationship between users and items based on the bipartite graph proposed by Yin et al. [17]. Users do not interact with many long-tailed items using a bipartite graph method, and these long-tail items cannot be

FIGURE 4: The bipartite graph [17].

traversed only by the random walk method. The tripartite graph method adds item categories as node sets based on the bipartite graph to associate each item with the item categories. The representation method of the tripartite graph is shown in Figure 5.

When a user selects a certain category of item using the tripartite graph, the long-tail item in the collection of items of this category may become the item in the user recommendation list. Experiments show that the proposed method based on the tripartite graph and Markov process is superior to the HT and AC methods proposed by Yin et al. [17] in terms of diversity and recall rate.

Luke et al. [22] proposed an extended tripartite graph method based on the tripartite graph proposed by Johnson and Ng [21]. Luke et al. used an extended tripartite graph method to represent the relationships among users, items, and item category. The proposed method uses the LSI (Latent Semantic Indexing) method [53] and SVD (Singular Value Decomposition) method [54] to implement long-tail item recommendation. The extended tripartite graph method divides the item category into separate categories, thus increasing the edge between the user and the item

FIGURE 5: The tripartite graph [21].

FIGURE 6: The extended tripartite graph [22].

category, and uses the Bayesian average [55] method to calculate the weight value of the edge. Figure 6 is an extended tripartite graph converted from Figure 5.

As can be seen from Figure 6, the item's composite category (e.g., ("Action," "Comedy")) is split into a single basic category (e.g., "Action" and "Comedy"). The weight value of the edge between the split categories and items adopts the quotient of the average value of the items' ratings and the total number of basic categories. The experimental results

show that the proposed method is more effective in terms of recall rate and diversity than that proposed by Johnson and Ng [21].

The summary information of the above graph-based long-tail item recommendation methods is shown in Table 4.

In [22], the results of specific diversity value in HT [17], AC1 [17], AC2 [17], T3 [21], T5 [21], T3-extended [22], and T5-extended [22] methods are given, as shown in Figure 7.

TABLE 4: The summary on graph-based long-tail item recommendation methods.

Ref.	Datasets	Main evaluation metrics	Proposed methods
[17]	MovieLens, Douban	Recall, popularity, diversity, novelty	HT, AT, AC1, AC2
[21]	MovieLens	Recall, diversity	Method based on tripartite graph and Markov chain
[22]	MovieLens	Recall, diversity	Extended tripartite graph method (based on [21])

FIGURE 7: Diversity values on different methods [22].

As can be seen in Figure 7, T3-extend has the highest diversity value, increasing from 0.404 at T3 to 0.45. In addition, the T3-extend value is also the best in terms of recall value, and the order of the effects of the above methods of recall value is (T3-extend > (AC2, T5, T3, AC1) > T5-extend > HT), and the recall values of AC2, T5, T3, and AC1 are relatively close.

In sum, representing the user's rating data in the form of graphs (bipartite graph, tripartite graph, and extended tripartite graph) can more comprehensively represent the relationship between users and items. Thus, researchers applied the random walk, Markov chain, LDA model, LSI model, and other methods to realize the recommendation of long-tail items. However, the long-tail item recommendation method based on graphs still has some shortcomings as follows:

- (1) Using graph representation cannot represent the context information of users and items and the relationship between items. Therefore, the method of using a graph has certain limitations in mining long-tail items
- (2) It does not guarantee that each user is recommended by long-tail items
- (3) It does not take into account the recall rate and diversity differences between different users

3.3. Multiobjective Optimization-Based Long-Tail Item Recommendation Methods. Multiobjective optimization-based long-tail item recommendation methods [19, 31–33] have been widely used in recent years. The so-called multiobjective optimization method refers to the method used to train a recommendation model, where not only one recall

rate or accuracy rate is applied to train the model but also multiple objectives are taken into account at the same time. For example, in terms of recall rate and diversity, two functions are combined into one function and use differently weighted values for each function. Unlike multiobjective optimization methods, Ho et al. [26] proposed the 5-D score method to allocate items as resources.

Wang et al. [31] proposed a long-tail item recommendation framework, called MORS (multiobjective long-tail recommendation framework). The basic idea of this method is (1) to convert the user's rating data of items into the form of a rating matrix, (2) to use the collaborative filtering method to rate and predict the items that the user has not rated, obtain top- k items for each user, and calculate the popularity of each item, (3) to use the MOEA (multiobjective evolutionary algorithm) method to evaluate the recommended results, and (4) to return the evaluated recommendation results to the target users. The recommendation accuracy rate and the recommendation of nonpopular items are taken as objective functions. The recommendation accuracy rate adopts the rating sum of the items in the recommendation list, and the formula is as follows:

$$\text{Accuracy} = \sum_{i=1}^L r_{u,i}, \quad (1)$$

where L denotes the recommendation list's length and $r_{u,i}$ denotes the user u rating of item i . The higher the accuracy value is, the more popular the recommended items are. The recommendation of nonpopular items uses the function proposed by Jambor and Wang [56]. The formula is as follows:

$$m_i = \frac{1}{\mu_i(\sigma_i + 1)^2}, \quad (2)$$

where μ_i represents the reciprocal of the average rating value of item i and σ_i represents the variance of the total score value of item i . The higher the value of m_i , the lower the popularity of the item. Therefore, taking this function as the objective function, the optimization objective of this function is still that the larger the value of m_i the better. The maximum of both objective functions is contained in the MORS framework. The proposed framework uses the collaborative filtering method to predict the user's rating. However, the user's rating of long-tail items is sparse. Thus, the prediction accuracy rate is relatively low.

Hu et al. [19] proposed a method to improve the recommendation quality of long-tail users and long-tail items. Unlike improving the diversity and surprise of recommendations, the proposed method improves the credibility and

specificity of the recommendation results. The framework of multiobjective function optimization proposed in this paper includes two latent factor models, C-HMF (Creativity-Specific Heteroscopic Matrix Factorization) and S-HMF (Specific-Specific Heteroscopic Matrix Factorization). The C-HMF model is used to find the long-tail items that best meet the user's preferences, while the S-HMF model obtains the user's feedback on the recommended items.

Pang et al. [32] proposed a multiobjective optimization method based on the NSGA-II (Nondominated Sorting Genetic Algorithm II) [57] algorithm. The proposed method sets the objective function into two objective functions (MAE (Mean Absolute Error) and coverage). The formulas are as follows (equations (3) and (4)):

$$\text{MAE}(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^n |y - \hat{y}|, \quad (3)$$

where n represents the number of ratings of the predicted results, y represents the real rating value of the item, and $\hat{y}$ represents the predicted rating value of the item.

$$\text{Coverage} = \frac{1}{I} (|\cup_{u \in U} R(u)|), \quad (4)$$

where $R(u)$ represents the set of nonrepeating items recommended to the user and I represents the total amount of items in the dataset. MAE metric is used to calculate the error value of the user's rating prediction of the item. Coverage is used to calculate the proportion of items recommended for users to all items. When the result is 1, this means that all items are recommended. It is also a measurement method to measure long-tail items. The objective of the optimization method proposed in this paper is to minimize MAE metric and maximize coverage metric.

In the long-tail item recommendation method proposed by Hamedani and Kaedi [33], the objective function consists of three functions: diversity, long-tailed distribution measurement, and accuracy functions. The diversity functions used attribute-based diversity methods, such as genre diversity. The formulas are as follows (equations (5), (6), and (7)):

$$\text{PersonalizedDiversity}_a = |E_a - \text{Diversity}_a|, \quad (5)$$

where E_a denotes the user's entropy of the items' attribute a and Diversity_a denotes item diversity based on attribute a .

$$\text{LongTail} = \sum_{i=1}^L \text{Popularity}(i), \quad (6)$$

where L denotes the recommendation list's length and $\text{Popularity}(i)$ denotes the rating times of item i .

$$\text{Accuracy} = \frac{1}{\sum_{i=1}^L \text{predictedRate}(i)}, \quad (7)$$

where L denotes the recommendation list's length and $\text{predictedRate}(i)$ denotes a rating prediction result for item i .

The function to measure long-tail distribution refers to the calculation of the popularity of items. The accuracy rate function calculates the reciprocal of all rating values of the item. Therefore, the smaller the accuracy value, the more accurate the predicted result. The optimization goal of these three functions is to achieve as low a value for each function as possible. The collaborative filtering method is also used to predict the user's rating value for unrated items.

Besides, Ho et al. [26] proposed a strategy (5D-score) to score items in terms of five dimensions (accuracy), distribution of items in the user's recommendation list (balance), coverage (coverage), quality of long-tailed items, and quantity of long-tailed items as resources for users. Recommendation quality judgment refers to the average value of item scores in the user's recommendation list. The recommendation method proposed includes two processes: the resource allocation stage and the recommendation stage. The resource allocation phase has two goals. One goal is to allow items with high scores to receive more resource allocation to ensure the quality of the recommendation results, and the other goal is to allow nonpopular items with low scores to receive a reasonable amount of resource allocation. The resource allocation function is shown in the following:

$$\text{argmax}_{X=\{x_1, x_2, \dots, x_n\}} \prod_{i \in I} x_i^{\text{allocation}(i)}. \quad (8)$$

Through this objective function, the method obtains a vector X of n dimensions; I represents a set of items. The calculation formula of the allocation function $\text{allocation}(i)$ is as follows:

$$\text{allocation}(i) = \sum_{u \in U} \frac{1}{\sigma_u} (r_{u,i} - \bar{R}_u), \quad (9)$$

where $i \in I$, $r_{u,i}$ represents user u 's rating of item i , $\bar{R}_u$ represents the average value of user u 's score of all items, and σ_u represents the variance of user u 's rating of all items. Through the resource allocation function, the preference of all users for the item i can be calculated.

In the recommendation stage, there are also two goals to be achieved. The first goal is to attach importance to the diversity of items selected by each user, and the second goal is to give long-tail items more opportunities to be recommended to users. It can be seen from the experimental results that the method proposed in this paper can effectively provide users with recommendations for long-tail items. In the recommendation process, only the items in the user's recommendation list are rated, and the relationship between the items is not considered.

The summary information of the above multiobjective optimization-based long-tail item recommendation methods is shown in Table 5.

From the objective functions listed in Table 5, it can be seen that the objective functions concerned by the long-tail item recommendation method are mainly accuracy and

TABLE 5: The summary on multiobjective optimization-based long-tail item recommendation methods.

Ref.	Datasets	Main evaluation metrics	Objective function
[19]	Amazon [58] Red dataset [59]	Recall, AP, MAE, NDCG	C-HMF, S-HMF
[26]	MovieLens	Precision, coverage, Gini coefficient	5D-score
[31]	MovieLens, Jester, Netflix	Precision, diversity, novelty	Accuracy, nonpopular items metric (novelty)
[32]	MovieLens, Netflix	Precision, recall, coverage, MAE	MAE, coverage
[33]	MovieLens, Netflix	Precision, RMSE, personalized diversity metric, novelty	Diversity, long-tailed distribution, accuracy

diversity, but the functions of calculating accuracy and diversity are slightly different.

In [31], the running time of the MORS recommended method and its comparison method (CF (item-based collaborative filtering) [60], MF [61], CT [29], SOO [31], NNCosNbr (Nonnormalized Cosine Neighborhood) [62], PureSVD [62]) are given. Taking MovieLens dataset as an example, the results are shown in Figure 8.

As can be seen from Figure 8, the MORS method runs in 44.1 seconds on the MovieLens dataset. Although it is not the fastest, its running speed is much lower than the recommended method based on matrix decomposition (MF, PureSVD).

To sum up, multiobjective optimization-based long-tail item recommendation methods not only improve the recommendation rate of long-tail items but also take into account the accuracy of the recommendation results. However, there are three shortcomings as follows:

- (1) These methods do not consider different objective optimization methods for different users
- (2) These methods do not consider user's long-term preferences and short-term preferences
- (3) These methods do not consider the influence of multiple objective functions on the recommendation efficiency

3.4. Deep Learning-Based Long-Tail Item Recommendation Methods.

The deep learning method has stronger representation ability in text and image. Hong et al. proposed miniGCN (Minibatch Graph Convolutional Network) model for image classification based on CNN (Convolutional Neural Networks) and GCN (Graph Convolutional Networks). Experiments show that the miniGCN method is better than CNN or GCN models [4]. Hong et al. proposed an MDL (multimodal deep learning) framework for image classification. The advantage of this framework is to effectively use multimodal data and improve the diversity of information [63]. At the same time, the deep learning method has also become the important method of user profile in the recommendation system. Compared with traditional recommendation methods, it improves the accuracy of recommendation results [64]. In 2020, the long-tail item recommendation method began to the deep learning method. For example, Bai et al. [36] use stacked denoising autoencoders (SDAE) to realize online long-tail item recom-

FIGURE 8: The running time on different methods [31].

mendation, and Kumar et al. [12] proposed to realize long-tail item recommendation using few shot learning.

Bai et al. [36] proposed a deep learning framework for long-tail item recommendation (DLTSR). The DLTSR mainly used stacked denoising autoencoder (SDAE) technology [65]. The DLTSR is used for web service recommendations. The experimental data in this paper were obtained on "http://ProgrammableWeb.com." The idea of the DLTSR has two aspects: (1) the knowledge obtained from a hot service is used in SDAE, and (2) a special mechanism is designed to establish developers' preferences using the rating of long-tail items.

Sreepada and Patra [34] used a few shot learning techniques to solve the recommendation problem of long-tail items. The user preference matrix uses OCCF (One Class Collaborative Filtering Technology) [66]. Before calculating the user preference, the rating matrix of the original data needs to be converted into a binary rating matrix (values with scoring values set to 1). When extracting item sets, the items selected by each user are divided into liked item list, long-tail item list, and disliked item list in each cluster. A Siamese network [67] is used to calculate the similarity between the clustering results and user preference matrix and the top- k items recommended to users. In addition, this paper also proposes three methods (LTC (Long-Tail Coverage), WLTC (Weighted Long-Tail Coverage), and TTC (Tail of the Tail Coverage)) to verify the effectiveness of long-tail recommendation.

Aiming at the long-tail problem caused by the high repetition rate of recommended items in the nearest neighbor

TABLE 6: The summary on deep learning-based long-tail item recommendation methods.

Ref.	Datasets	Main evaluation metrics	Deep learning methods
[20]	MovieLens, BookCrossing	Recall, coverage, diversity	CNN
[34]	MovieLens, Netflix	Precision, recall, F1-score, Bpref, LTC, WLTC, TTC	Siamese network
[35]	YOOCHOOSE, 30MUSIC	Recall, MRR, coverage, tail_coverage, tail	GRU encoder, attention mechanism
[36]	http://ProgrammableWeb.com	Recall, diversity	Stacked denoising autoencoders

recommendation method, Qin et al. [20] proposed a CNN- (Convolutional Neural Network-) based recommendation framework to reduce the frequent recommendation items and recommend long-tail items. The proposed method reduced the frequent recommendation items and improved the coverage and diversity of recommendation results.

In addition, Liu and Zheng [35] proposed a framework for long-tail item recommendation from session data, and it is designed based on a neural network and attention mechanism. Three methods are used to verify coverage, namely, coverage, tail_coverage, and tail. Coverage is used to verify the number of items in the recommendation results, tail_coverage is used to verify the number of long-tail items in the recommendation results, and tail is used to verify the average number of long-tail items in each recommendation list. Experiments show that the proposed method (TailNet) is more accurate than the traditional recommendation method in the experiment and has better coverage than the recommendation method based on deep learning in the experiment.

The summary information of the above deep learning-based long-tail item recommendation methods is shown in Table 6.

There has not been a lot of research on the deep learning method in long-tail item recommendation. At present, there are some drawbacks associated with the long-tail item recommendation method based on deep learning: (1) it does not provide a more in-depth analysis of the correlation between items, (2) the user's context information is not added to the features of deep learning, and (3) the profile of the user is not targeted, and the long-tail items cannot be recommended to the user in a targeted way.

3.5. Other Long-Tail Item Recommendation Methods. In addition to the above-mentioned long-tail item recommendation methods, researchers have also adopted the ranking method [25], linear-model-based method [27], relevance model-based methods [37], user value-based method [38], and multilevel item similarity calculation method [39].

Zhang et al. [25] proposed a long-tail recommendation method based on the double ranking method. In this method, items are ranked twice, once in the training stage and again in the recommendation stage. In the recommendation phase, the values of c_{ui} obtained in the training phase are arranged in ascending order, and then, the first top- N items are recommended to the user. CI (concentration index) and GI [68] are used to evaluate the correlation between recommended items. The results show that the proposed method is more effective than traditional methods, such as cosine similarity and the Jaccard correlation coefficient.

The linear model is a simple and effective model in machine learning methods. Using the mixed linear model to model data can better solve the problems encountered in practical application. For example, the ALMM (augmented linear mixing model) (considering principal scaling factor and other variable spectral variability) is effectively used for spectral unmixing [69]. Li et al. [27] proposed a recommendation method based on a linear model, which solves the cold-start problem of new users and the long-tail item recommendation problem at the same time. The proposed method is also the first to solve the two problems simultaneously. In this paper, cold-start recommendation is defined as recommendations for new users (users have not rated any items). The proposed method divides the rating matrix (Y) into two parts: the rating data (SH) on the user and the short-headed items, and the interaction data (LT) on the user and long-tailed items, i.e., $Y = Y_{SH} + Y_{LT}$. Based on the idea of the linear model [70], Y_{SH} and Y_{LT} are expressed as XW and XH , respectively, and then, the final weight values W and H are obtained through learning. In addition, the matrix representation used for the cold-start problem is as follows:

$$Y_{\text{new}} = X_{\text{new}}W + X_{\text{new}}H, \quad (10)$$

where X_{new} represents the user's attribute matrix or social relationship matrix. X and H represent the weight value learned by the short-head matrix portion and the weight value learned by the long-tail matrix portion, respectively. The proposed method only recommends short-head items and long-tail items to the user, and it does not consider the diversity of recommended items.

Harper and Konstan [15] proposed a collaborative filtering recommendation relevance model for long-tail items. The relevance model [71] includes hypothetical independent and identical distribution sampling (RM1) and conditional sampling models (RM2). Valcarce et al. proposed IRM2 (Item Relevance Modelling 2) based on RM2. The basic idea of the IRM2 model is to establish a correlation model R_i between each user and the long-tailed item i . The rating probability of user u on long-tailed item i is expressed as $p(u | R_i)$. Based on the idea of random sampling and Bayesian conditional probability, the IRM2 model obtains the probability estimation of $p(u | R_i)$ and uses the maximum likelihood estimation to obtain the rating of user u for long-tail items.

Takama et al. [38] proposed a long-tail item recommendation method based on personal values. Personal values are an important factor affecting users' purchases. Hattori and Takama used the RMRate (Rating Matching Rate) model

[72] to model personal values and then used a user-based collaborative filtering method to find neighbors and calculate the scores of neighbors.

Alshammari et al. [39] proposed a long-tail item recommendation framework based on the multilevel similarity method [73] to calculate the similarity between items. The proposed framework includes two components (a collaborative filtering component and content-based component). The collaborative filtering component is used to predict users' ratings of items. The content-based component is used to calculate the predicted rating based on movie genres in user's profile. When predicting the score of an item, the collaborative filtering component is used if the item is scored more times; otherwise, the content-based component is used. Unlike traditional similarity calculation [74], the collaborative filtering method used the multilevel similarity calculation method. While the rating accuracy of long-tail items can be improved in the experimental results, when users predict long-tail items are not considered, the items selected by users in their history are very few, and the content-based recommendation method cannot accurately predict the user's rating of items.

4. Future Directions

With the continuous upgrading of the internet and mobile devices, long-tail item recommendation will directly affect the income of merchants and the experience of users. Therefore, the long-tail recommendation is the common demand of merchants and users. While the existing work has been able to provide users with more long-tail items, there are still many challenges in the research on long-tail item recommendations. The following will introduce the problems worth studying in the field of long-tail item recommendation in the future.

4.1. Evaluation Metric of Long-Tail Item Recommendation Results. The long-tail item recommendation method lacks unified evaluation metrics to measure the quality of long-tail item recommendations. The coverage rate and diversity metric in the existing evaluation metrics can only measure the overall effect of the recommendation results and cannot accurately determine whether the long-tail items meet the needs of users. Using the method of popularity distribution to evaluate the recommendation effect of long-tail items can only reflect the proportion of long-tail items recommended, but it cannot reflect whether these items meet the needs of users. Therefore, how to design a reasonable metric to meet the needs of individual users and measure the needs for long-tail items of users, in turn, is worthy of in-depth study.

4.2. Context Information in Long-Tail Item Recommendation. The existing long-tail item recommendation method only considers the rating information of items, the features of users, and the features of products. Little consideration is paid to the context information of users. For example, recommendation methods recommend different long-tail items for users at different locations and at different times. Therefore, how to relate

the long-tail items to the user's context information is also a problem worth studying.

4.3. Long-Tail Item Recommendation for Sequential Data. In the long-tail item recommendation method, the research results on sequential data recommendation are few, and only Liu and Zheng [35] proposed a long-tail recommendation framework based on session data. If the recommendation method only considered the data in each session to recommend long-tail items to users, the long-tail items will only be related to the short-term preferences of users, ignoring their long-term preferences. Therefore, how to balance users' long-term and short-term preferences and combine users' context information is a problem worth studying for sequential data.

5. Conclusions

This paper is the first review of the long-tail item recommendation problem. This paper first introduces the definition and general framework of long-tail item recommendation. Then, we summarize the recommendation method based on clustering, graphing, multiobjective optimization, deep learning, and other recommendation methods according to the timeline of recommendation method research. Finally, we discuss the future directions of long-tail item recommendation research. We hope this paper is helpful to researchers.

Data Availability

The data involved in the paper comes from the corresponding cited paper.

Conflicts of Interest

The author declares that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

The author would like to thank the authors of all the references.

References

- [1] D. Goldberg, D. A. Nichols, B. M. Oki, and D. B. Terry, "Using collaborative filtering to weave an information tapestry," *Communications of the ACM*, vol. 35, no. 12, pp. 61–70, 1992.
- [2] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. G. L. Riedl, "An open architecture for collaborative filtering of Netnews. CSCW '94," in *Proceedings of the Conference on Computer Supported Cooperative Work*, pp. 175–186, Chapel Hill, NC, USA, 1994.
- [3] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proceedings of the 10th International Conference on World Wide Web*, ACM, pp. 285–295, New York, NY, USA, 2001.
- [4] D. Hong, L. Gao, J. Yao, B. Zhang, A. Plaza, and J. Chanussot, "Graph convolutional networks for hyperspectral image

- classification,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 7, pp. 5966–5978, 2021.
- [5] S. Jia, “A survey: deep learning for hyperspectral image classification with few labeled samples,” *Neurocomputing*, vol. 448, pp. 179–204, 2021.
 - [6] H. D. Nguyen, X. S. Vu, and D. T. Le, “Modular graph transformer networks for multi-label image classification,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 10, pp. 9092–9100, 2021.
 - [7] I. Kim, W. Baek, and S. Kim, “Spatially attentive output layer for image classification,” in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9530–9539, IEEE, 2020.
 - [8] W. Shi and R. Ragunathan, “Point-GNN: graph neural network for 3D object detection in a point cloud,” in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1708–1716, 2020.
 - [9] M. Ma, C. Xia, and J. Li, “Pyramidal feature shrinking for salient object detection,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 3, pp. 2311–2318, 2020.
 - [10] Q. Chen, Z. Liu, Y. Zhang, K. Fu, Q. Zhao, and H. Du, “RGB-D salient object detection via 3D convolutional neural networks,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 2, pp. 1063–1071, 2021.
 - [11] D. Yang, Y. Zou, C. Zhang, M. Cao, and J. Chen, “RR-Net: injecting interactive semantics in human-object interaction detection,” in *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, pp. 1224–1230, 2021.
 - [12] Y. Kumar, D. Sahrawat, S. Maheshwari et al., “Harnessing GANs for zero-shot learning of new classes in visual speech recognition,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 3, pp. 2645–2652, 2020.
 - [13] M. Doumbouya, L. Einstein, and C. Piech, “Using radio archives for low-resource speech recognition: towards an intelligent virtual assistant for illiterate users,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 17, pp. 14757–14765, 2021.
 - [14] T. Mittal, U. Bhattacharya, R. Chandra, A. Bera, and D. Manocha, “M3ER: multiplicative multimodal emotion recognition using facial, textual, and speech cues,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 2, pp. 1359–1367, 2020.
 - [15] F. M. Harper and J. A. Konstan, “The MovieLens datasets: history and context,” *ACM Trans. Interact. Intell. Syst.*, vol. 5, no. 4, pp. 1–19, 2015.
 - [16] C. Anderson, “The long tail: why the future of business is selling less of more,” *Hachette Books*, 2006.
 - [17] H. Yin, B. Cui, J. Li, J. Yao, and C. Chen, “Challenging the long tail recommendation,” *Proceedings of the VLDB Endowment*, vol. 5, no. 9, pp. 896–907, 2012.
 - [18] A. Hervas-Drane, *Word of Mouth and Taste Matching: A Theory of the Long Tail*, Working Papers, 2009.
 - [19] L. Hu, L. Cao, J. Cao, Z. Gu, G. Xu, and J. Wang, “Improving the quality of recommendations for users and items in the tail of distribution,” *ACM Transactions on Information Systems*, vol. 35, no. 3, pp. 1–37, 2017.
 - [20] J. Qin, Q. Zhang, and B. Wang, “Recommendation method with focus on long tail items,” *Journal of Computer Applications*, vol. 40, pp. 454–458, 2020.
 - [21] J. Johnson and Y. Ng, “Enhancing long tail item recommendations using tripartite graphs and Markov process,” in *Proceedings of the International Conference on Web Intelligence*, pp. 761–768, Leipzig, Germany, 2017.
 - [22] A. Luke, J. Johnson, and Y. Ng, “Recommending long-tail items using extended tripartite graphs,” in *2018 IEEE International Conference on Big Knowledge, ICBK 2018*, pp. 123–130, Singapore, 2018.
 - [23] Y. Park, “The adaptive clustering method for the long tail problem of recommender systems,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 8, pp. 1904–1915, 2013.
 - [24] X. Huang and F. Wu, “A novel topic-based framework for recommending long tail products,” *Computers and Industrial Engineering*, vol. 137, p. 106063, 2019.
 - [25] M. Zhang, N. Hurley, W. Li, and X. Xue, “A double-ranking strategy for long-tail product recommendation,” in *2012 IEEE/WIC/ACM International Conferences on Web Intelligence, WI 2012*, pp. 282–286, Macau, China, 2012.
 - [26] Y. Ho, Y. Chiang, and J. Y. Hsu, “Who likes it more?: mining worth-recommending items from long tails by modeling relative preference,” in *Seventh ACM International Conference on Web Search and Data Mining, WSDM 2014*, pp. 253–262, New York, NY, USA, 2014.
 - [27] J. Li, K. Lu, Z. Huang, and H. T. Shen, “Two birds one stone: on both cold-start and long-tail recommendation,” in *Proceedings of the 2017 ACM on multimedia conference, MM 2017, mountain view*, pp. 898–906, CA, USA, 2017.
 - [28] X. N. Lam, T. Vu, T. D. Le, and A. D. Duong, “Addressing cold-start problem in recommendation systems,” in *Proceedings of the 2nd International Conference on Ubiquitous Information Management and Communication, ICUIMC 2008*, pp. 208–211, Suwon, Korea, 2008.
 - [29] Y. Park and A. Tuzhilin, “The long tail of recommender systems and how to leverage it,” in *Proceedings of the 2008 ACM conference on recommender systems, RecSys 2008*, pp. 11–18, Lausanne, Switzerland, 2008.
 - [30] V. Grozin and A. Levina, “Similar product clustering for long-tail cross-sell recommendations,” in *Supplementary Proceedings of the Sixth International Conference on Analysis of Images, Social Networks and Texts (AIST 2017)*, pp. 273–280, Moscow, Russia, 2017.
 - [31] S. Wang, M. Gong, H. Li, and J. Yang, “Multi-objective optimization for long tail recommendation,” *Knowledge-Based Systems*, vol. 104, pp. 145–155, 2016.
 - [32] J. Pang, J. Guo, and W. Zhang, “Using multi-objective optimization to solve the long tail problem in recommender system,” in *Advances in Knowledge Discovery and Data Mining - 23rd Pacific-Asia Conference, PAKDD 2019*, pp. 302–313, Macau, China, 2019.
 - [33] E. M. Hamedani and M. Kaedi, “Recommending the long tail items through personalized diversification,” *Knowledge-Based Systems*, vol. 164, pp. 348–357, 2019.
 - [34] R. S. Sreepada and B. K. Patra, “Mitigating long tail effect in recommendations using few shot learning technique,” *Expert Systems with Applications*, vol. 140, article 112887, 2020.
 - [35] S. Liu and Y. Zheng, “Long-tail session-based recommendation,” in *In Fourteenth ACM Conference on Recommender Systems (RecSys '20)*. Association for Computing Machinery, pp. 509–514, New York, NY, USA, 2020.
 - [36] B. Bai, Y. Fan, W. Tan, and J. Zhang, “DLTSR: a deep learning framework for recommendations of long-tail web services,”

- IEEE Transactions on Services Computing*, vol. 13, no. 1, pp. 73–85, 2020.
- [37] D. Valcarce, J. Parapar, and Á. Barreiro, “Item-based relevance modelling of recommendations for getting rid of long tail products,” *Knowledge-Based Systems*, vol. 103, pp. 41–51, 2016.
 - [38] Y. Takama, Y. Chen, R. Misawa, and H. Ishikawa, “Analyzing potential of personal values-based user modeling for long tail item recommendation,” *Journal of Advanced Computational Intelligence and Intelligent Informatics*, vol. 22, no. 4, pp. 506–513, 2018.
 - [39] G. Alshammari, J. L. Jorro-Aragoneses, N. Polatidis, S. Kapetanakis, E. Pimenidis, and M. Petridis, “A switching multi-level method for the long tail recommendation problem,” *Journal of Intelligent Fuzzy Systems*, vol. 37, no. 6, pp. 7189–7198, 2019.
 - [40] K. Truong, F. Ishikawa, and S. Honiden, “Improving accuracy of recommender system by item clustering,” *IEICE Transactions on Information and Systems*, vol. E90-D, no. 9, pp. 1363–1373, 2007.
 - [41] R. Agrawal and R. Srikant, “Fast algorithms for mining association rules in large databases. VLDB’94,” in *Proceedings of 20th International Conference on Very Large Data Bases*, pp. 487–499, Santiago de Chile, 1994.
 - [42] B. Kitts, D. Freed, and M. Vrieze, “Cross-sell: a fast promotion-tunable customer-item recommendation method based on conditionally independent probabilities,” in *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 437–446, Boston, MA, USA, 2000.
 - [43] M. Grbovic, V. Radosavljevic, N. Djuric et al., “E-commerce in your inbox: product recommendations at scale,” in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1809–1818, Sydney, NSW, Australia, 2015.
 - [44] D. Choi and C. Chung, “A K-partitioning algorithm for clustering large-scale spatio-textual data,” *Information Systems*, vol. 64, pp. 1–11, 2017.
 - [45] Z. Huang, “Extensions to the k-means algorithm for clustering large data sets with categorical values,” *Data Mining and Knowledge Discovery*, vol. 2, no. 3, pp. 283–304, 1998.
 - [46] A. Ahmad and L. Dey, “A k -mean clustering algorithm for mixed numeric and categorical data,” *Data & Knowledge Engineering*, vol. 63, no. 2, pp. 503–527, 2007.
 - [47] J. Ji, W. Pang, C. Zhou, X. Han, and Z. Wang, “A fuzzy k-prototype clustering algorithm for mixed numeric and categorical data,” *Knowledge-Based Systems*, vol. 30, pp. 129–135, 2012.
 - [48] X. Cheng, X. Yan, Y. Lan, and J. Guo, “BTM: topic modeling over short texts,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 12, pp. 2928–2941, 2014.
 - [49] W. X. Zhao, J. Jiang, H. Yan, and X. Li, “Jointly modeling aspects and opinions with a MaxEnt-LDA hybrid,” in *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, EMNLP 2010*, pp. 56–65, ACL, 2010.
 - [50] X. Chen, W. Tang, H. Xu, and X. Hu, “Double LDA: a sentiment analysis model based on topic model,” in *10th international conference on semantics, knowledge and grids (SKG)*, pp. 49–56, 2014.
 - [51] R. K. Amplayo and M. Song, “An adaptable fine-grained sentiment analysis for summarization of multiple short online reviews,” *Data & Knowledge Engineering*, vol. 110, pp. 54–67, 2017.
 - [52] M. S. Shang, Z. K. Zhang, T. Zhou, and Y. C. Zhang, “Collaborative filtering with diffusion-based similarity on tripartite graphs,” *Physica A Statistical Mechanics and its Applications*, vol. 389, no. 6, pp. 1259–1264, 2010.
 - [53] T. Hofmann, “Probabilistic latent semantic indexing,” in *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 50–57, Berkeley, CA, USA; ACM, 1999, 1999.
 - [54] V. Klema and A. Laub, “The singular value decomposition: its computation and some applications,” *Automatic Control, IEEE Transactions on*, vol. 25, no. 2, pp. 164–176, 1980.
 - [55] X. Yang and Z. Zhang, “Combining prestige and relevance ranking for personalized recommendation,” in *22nd ACM International Conference on Information and Knowledge Management, CIKM’13*, pp. 1877–1880, San Francisco, CA, USA, 2013.
 - [56] T. Jambor and J. Wang, “Optimizing multiple objectives in collaborative filtering,” in *Proceedings of the 2010 ACM Conference on Recommender Systems, RecSys 2010*, pp. 55–62, Barcelona, Spain, 2010.
 - [57] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II,” *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
 - [58] J. McAuley, R. Pandey, and J. Leskovec, “Inferring networks of substitutable and complementary products,” in *In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794, ACM, New York, 2015.
 - [59] S. Meyffret, E. Guillot, L. M’Edini, and F. Laforest, “RED: a rich Epinions dataset for recommender systems,” *Research Report, LIRIS*, 2012.
 - [60] X. Su and T. M. Khoshgoftaar, “A survey of collaborative filtering techniques,” *Advances in Artificial Intelligence*, vol. 2009, Article ID 421425, 2009.
 - [61] Y. Koren, R. Bell, and C. Volinsky, “Matrix factorization techniques for recommender systems,” *Computer*, vol. 42, no. 8, pp. 30–37, 2009.
 - [62] P. Cremonesi, Y. Koren, and R. Turrin, “Performance of recommender algorithms on top-n recommendation tasks,” in *In Proceedings of the fourth ACM conference on Recommender systems (RecSys ’10)*. Association for Computing Machinery, pp. 39–46, New York, NY, USA, 2010.
 - [63] D. Hong, L. Gao, N. Yokoya, J. Yao, and B. Zhang, “More diverse means better: multimodal deep learning meets remote-sensing imagery classification,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 5, pp. 4340–4354, 2021.
 - [64] S. Zhang, L. Yao, A. Sun, and Y. Tay, “Deep learning based recommender system: a survey and new perspectives,” *ACM Computing Surveys*, vol. 52, pp. 1–38, 2019.
 - [65] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P. Manzagol, “Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion,” *Journal of Machine Learning Research*, vol. 11, pp. 3371–3408, 2010.
 - [66] R. Pan, Y. Zhou, B. Cao et al., “One-class collaborative filtering,” in *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM 2008)*, pp. 502–511, Pisa, Italy. IEEE Computer Society, 2008.

- [67] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a Siamese time delay neural network," in *Advances in Neural Information Processing Systems* 6, 7th NIPS Conference, pp. 737–744, Denver, Colorado, USA, 1993.
- [68] C. S. Moskowitz, V. E. Seshan, E. R. Riedel, and C. B. Begg, "Estimating the empirical Lorenz curve and Gini coefficient in the presence of error with nested data," *Statistics in Medicine*, vol. 27, no. 16, pp. 3191–3208, 2008.
- [69] D. Hong, N. Yokoya, J. Chanussot, and X. X. Zhu, "An augmented linear mixing model to address spectral variability for hyperspectral unmixing," *IEEE Transactions on Image Processing*, vol. 28, no. 4, pp. 1923–1938, 2019.
- [70] S. Sedhain, A. K. Menon, S. Sanner, L. Xie, and D. Braziunas, "Low-rank linear cold-start recommendation from social data," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pp. 1502–1508, San Francisco, California, USA; AAAI Press, 2017, 2017.
- [71] V. Lavrenko and W. B. Croft, "Relevance-based language models," in *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 120–127, New Orleans, Louisiana, USA, 2001, 2001.
- [72] S. Hattori and Y. Takama, "Recommender system employing personal-value-based user model," *Journal of Advanced Computational Intelligence & Intelligent Informatics*, vol. 18, no. 2, pp. 157–165, 2014.
- [73] N. Polatidis and C. K. Georgiadis, "A multi-level collaborative filtering method that improves recommendations," *Expert Systems with Applications*, vol. 48, pp. 100–110, 2016.
- [74] K. Shen, Y. Liu, and Z. Zhang, "Modified similarity algorithm for collaborative filtering," *Knowledge Management in Organizations KMO 2017 Communications in Computer and Information Science*, vol. 731, pp. 378–385, 2017.