Information Sciences 582 (2022) 22–37

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

A two-stage embedding model for recommendation with multimodal auxiliary information

Juan Ni [a], Zhenhua Huang [b,c,*,1], Yang Hu [a], Chen Lin [d]

[a] School of Philosophy and Social Development, South China Normal University, Guangzhou 510631, China
[b] School of Computer Science, South China Normal University, Guangzhou 510631, China
[c] Research and Development Department, DataGrand Inc., Shenzhen 518063, China
[d] School of Informatics, Xiamen University, Xiamen 361000, China

ARTICLE INFO

Article history:
Received 5 October 2020
Received in revised form 2 August 2021
Accepted 5 September 2021
Available online 08 September 2021

Keywords:
Recommendation model
Multimodal information
Embedding
Multi-task learning
Graph convolutional network

ABSTRACT

Recommender system has recently received a lot of attention in the information service community. In many application scenarios, such as Internet of Things (IoTs) environments, item multimodal auxiliary information (such as text and image) can be obtained to expand their feature representation and to increase user satisfaction with recommendations. Motivated by this fact, this paper introduces a novel two-stage embedding model (TSEM), which adequately leverage item multimodal auxiliary information to substantially improve recommendation performance. Specifically, it encompasses two sequential stages: graph convolutional embedding (GCE) and multimodal joint fuzzy embedding (MJFE). In the former, we first generate a bipartite graph for user-item interactions, and then utilize it to construct user and item backbone features via a spatial-based graph convolutional network (SGCN). While in the latter, by employing item multimodal auxiliary information, we integrate multi-task deep learning, deterministic Softmax, and fuzzy Softmax into a convolutional neural network (CNN)-based learning framework, which is optimized to obtain user backbone features and item semantic-enhanced fuzzy (SEF) features accurately. After TSEM converges, user backbone features and item SEF features can be utilized to calculate user preferences on items via Euclidean distance. Extensive experiments over two real-world datasets show that the proposed TSEM model significantly outperforms the state-of-the-art baselines in terms of various evaluation metrics.

1. Introduction

With the rapid development of mobile communication and Internet of Things (IoTs) technology, the amount of information on the Internet has grown exponentially, which makes it difficult for users to find useful items such as products or services by themselves [1]. Recommender system can effectively solve this issue, which predicts users' potential interested items based on their preference history and recommends top-*k* items to them. According to the current practice, recommendation model is the core of recommender system, and it determines the performance of recommender system [2].

* Corresponding author at: School of Computer Science, South China Normal University, Guangzhou 510631, China.
E-mail address: huangzhenhua@m.scnu.edu.cn (Z. Huang).
[1] Juan Ni and Zhenhua Huang contributed equally to this work.

https://doi.org/10.1016/j.ins.2021.09.006
0020-0255/

Recently, both industry and academia have considered using deep learning to improve the effectiveness of traditional recommendation models [3,4]. Their core task is to accurately extract user and item features by applying various deep neural networks such as multilayer perceptron (MLP), convolutional neural network (CNN), and recurrent neural network (RNN), which is known as embedding process [5,6]. The adequacy and accuracy of feature embedding is a pivotal factor to influence recommendation effectiveness. According to our investigation, user feature embedding is relatively simple since recommender systems usually only involve user demographic information such as user-ID, age, gender, etc. A popular practice is to first map each attribute of an input user into a one-hot vector and then feed the concatenation of all one-hot vectors into a deep neural network for learning her feature embedding [7,8].

Compared with user feature embedding, item feature embedding is more difficult and requires more elaborate design. This is mainly because in a recommender system, besides descriptive information (such as item-ID, genre, price, etc.), items may also have auxiliary information (e.g., instruction documents). Specifically, in IoT environments, recommender systems usually can collect abundant multimodal auxiliary information for items such as text, image, video, etc. For instance, in a smart shopping system, laptops on sale may have the following multimodal auxiliary information: user reviews, instruction documents, pictures from different perspectives, promotional videos, sound samples, and semantic contents in knowledge base (or knowledge graph). Clearly, these multimodal auxiliary information of items can be leveraged to optimize their feature embedding and to improve user satisfaction with recommendations.

However, most existing works do not adequately consider the use of item multimodal auxiliary information. They usually only utilize item descriptive information [9–12] or text auxiliary information [13–17] for learning item feature embedding. And according to our investigation, there are currently a few works have made additional use of image [18–21,28], audio [22–24,26,27], video [25–27], and semantic contents in knowledge base [28,29]. They generally exploit a straightforward way to incorporate different modal auxiliary information into the recommendation process. Concretely, for each item, the features from its different modal information are separately extracted through deep neural networks, and all the extracted features are concatenated as its final multimodal feature. On this basis, user features and item multimodal features are combined for performing recommendations. Lately, the work [30] analyzes and points out that this straightforward way may lead to a poor recommendation performance. Then, it makes an attempt to adequately use item multimodal auxiliary information and introduces a multimodal representation learning-based model called MRLM, to improve recommendation effectiveness.

Although the MRLM model can effectively fuse item multimodal auxiliary information into recommendation, it still has two deficiencies on feature embedding. First, in a recommender system, each user and each item are not independent, and they are interrelated. Therefore, for a user or an item, its feature embedding will be affected by other related users and items. Yet, MRLM ignores this fact and mainly leverages user and item content information to perform feature embedding. Second, MRLM generates item multimodal features through jointly optimizing multiple deterministic modal classifiers. In particular, each modality corresponds to a classifier and each item's auxiliary feature on this modality is clustered into a deterministic class. However, this may be unreasonable because usually for an item, its auxiliary feature on each modality does not only correspond to a deterministic class and has the fuzzy-class characteristic. For example, for a fruit, its image auxiliary feature may have a 65.5% probability corresponding to apple-class and a 34.5% probability corresponding to orange-class. These two deficiencies are likely to influence accuracy of feature embedding and to ultimately affect recommendation effectiveness.

To address the above deficiencies, in this paper, a novel two-stage embedding model (TSEM) is proposed, which can fully leverage user-item interaction data and items' multimodal auxiliary information to improve recommendation effectiveness. Specifically, the proposed TSEM model mainly includes two sequential stages: graph convolutional embedding (GCE) and multimodal joint fuzzy embedding (MJFE). In the GCE stage, similar to the work [31], we first generate an undirected bipartite graph for user-item interactions by using user historical data. Then, we take the bipartite graph and user demographic information as input, and utilize a spatial-based graph convolutional network (SGCN) for constructing user and item backbone features. Inspired by previous works [31,32], the SGCN network is designed to contain two graph convolutional layers and two fully-connected layers. Via the GCE stage, the proposed TSEM model is able to employ both content information of users and items as well as graph topology structure adequately to construct effective user and item backbone features. In this way, the first deficiency of MRLM can be effectively alleviated.

While in the MJFE stage, we first take item backbone features and descriptive information as input, and use a three-layer CNN-based architecture for constructing item semantic-enhanced fuzzy (SEF) features. Then, under this architecture, we design multiple related task-components that are jointly optimized to get user backbone features and item SEF features accurately. These task-components involve one metric learner, one deterministic classifier, and m fuzzy classifiers. Here, m is the number of item's modalities. Specifically, the metric learner is employed to learn user preferences on items. The deterministic classifier is employed to identify items' different grades. Each fuzzy classifier corresponds to a modality, and its output classes are generated through performing fuzzy clustering algorithms [33–35] for item auxiliary features on this modality. Via the MJFE stage, the proposed TSEM model is able to fully consider: (*i*) the fuzzy-class characteristics of auxiliary features on each modality of items; (*ii*) the mutuality and the complementarity between different modalities of items; and (*iii*) the potential influences of different modalities of items on user preferences. In this way, the second deficiency of MRLM can be effectively alleviated.

After the TSEM model converges, user backbone features and item SEF features can be employed for calculating user preferences on items via Euclidean distance. Furthermore, the recommendation effectiveness of TSEM is investigated through

extensive experiments on two real-world datasets. And the results demonstrate that TSEM significantly outperforms the state-of-the-art baselines in terms of various evaluation metrics.

In summary, our main contributions are:

- We introduce a novel two-stage embedding model that can fully leverage user-item interaction data and items' multimodal auxiliary information to enhance recommendation performance.
- User and item backbone features can be effectively constructed via a SGCN network in the GCE stage. Compared with purely content-based deep neural networks, SGCN can adequately utilize both content information of users and items as well as graph topology structure.
- User backbone features and item semantic-enhanced fuzzy features can be accurately obtained through jointly learning multiple fuzzy modal classifiers in the MJFE stage. In particular, this joint learning method can exploit the mutuality and the complementarity among items' different modalities adequately and capture the potential influences of items' different modalities on user preferences accurately.
- We comprehensively investigate the effectiveness of the proposed model through the extensive experiments over two real-world datasets. The experimental results demonstrates that the proposed model significantly outperforms existing state-of-the-art models in terms of various evaluation metrics.

The rest of the paper is organized as follows: Section 2 introduces related works to this paper. Section 3 presents the details of the TSEM model. Experimental results are presented in Section 4. Finally, Section 5 concludes this paper and outlines future work.

2. Related work

This section introduces related works on item feature embedding in the recommender system domain, which include two categories.

The first category encompasses the works that perform item feature embedding by only utilizing item descriptive information. He et al. [9] introduce a deep learning-based framework, namely neural collaborative filtering (NCF), which uses a MLP to learn the user-item interaction function. Covington et al. [10] design a video recommendation model, which learns a score for each video through extracting user and item features from user demographic information and item descriptive information. Huang et al. [11] introduce a deep hybrid recommendation model called DMFL (Deep Metric Factorization Learning), which combines deep learning with improved machine learning framework for learning user-item interactions from multiple perspectives. Specifically, the DMFL model is able to overcome the deficiencies of individual methods and improve the overall recommendation performance. Yin et al. [12] present a generative adversarial network (GAN)-based model, which utilizes a generative network to perform recommendations and a discriminative network for guiding the training process. Specifically, the generative network can converge to an optimal solution under guidance of the discriminative network.

At present, there are two mainstream GCN-based recommendation models [31,32]. Ying et al. [31] introduce PinSage, a data efficient GCN-based framework. It is based on a bipartite graph as well as content information of users and items, and combines random walks and graph convolutions to perform feature embedding. Then, Wang et al. [32] point out that the PinSage model cannot accurately generate user and item embeddings because they do not sufficiently consider the collaborative signal between users and items. Based on this, the authors present a novel recommendation model called neural graph collaborative filtering (NGCF), which models the high-order connectivity in a bipartite user-item graph and integrates the user-item collaborative signal into the embedding process in an explicit fashion.

The second category contains the works in which item auxiliary information is used. However, to the best of our knowledge, most of them employ item text contents, i.e., single modality (e.g., the studies [13–17]), and a few works use auxiliary information on other modalities such as image [18–21,28], audio [22–24,26,27], video [24–27], and semantic contents in knowledge base [28,29].

Zhao et al. [13] introduce a novel predictive collaborative filtering model that leverages both the partially observed user-item interaction matrix and item reviews for performing recommendations. Kim et al. [14] propose a convolutional matrix factorization (ConvMF) model that uses item texts as auxiliary information. Specifically, it first utilizes CNN to extract item features from their text contents, and then employs a matrix factorization method [15] for calculating the scores of users on items. Chen et al. [16] design an effective recommendation model based on neural attentive regression called NARRE, which learns the importance of reviews for each item and performs the prediction of ratings via review-level explanations. Xing et al. [17] present HAUP, a hierarchical attention model by using product reviews. Specifically, it makes recommendations by jointly learning a user-product rating matrix and product review texts.

Zhang et al. [18] introduce a co-attention network that leverages both texts and images for performing hashtag recommendation. Inspired by the study of [18], Ma et al. [19] develop an effective cross-attention memory network (CAMN) that also utilizes texts and images to implement mention recommendations for tweets. Based on the NCF framework [9], Lin et al. [20] present an effective model called MF-VMLP, to fuse visual factors into user preference prediction. Specifically, it first utilizes a pre-trained CNN to obtain item visual features, and then incorporates item basic and visual features for learning

user preferences via a MLP. Moreover, Yang et al. [21] design an attention-based multimodal neural network model (AMNN) to learn the representations of multimodal microblogs and recommend relevant hashtags. It extracts the features of both texts and images and fuses them into the sequence-to-sequence framework for hashtag recommendation.

By the aid of deep neural networks, Oramas et al. [22] propose to combine item texts and audios with user feedback data to address the cold-start issue in the process of music recommendation. Bougiatiotis et al. [23] present an effective model called MRTA, which performs movie feature embedding via using movie subtitles (i.e., texts) and audios. Li et al. [24] introduce a content-based video recommendation model by utilizing deep CNNs to solve the cold-start issue. In particular, besides video features, the proposed model also utilizes video *meta*-data and audio features. Kumar et al. [25] present a new dataset called content based video relevance prediction (CBVRP), and propose different frameworks on the CBVRP dataset that use frame (i.e., image) and video level features to make video recommendations. Xu et al. [26] design a course recommendation model that extracts multimodal course features via a deep learning method. In this model, different kinds of information of course, such as course title, course audio and course comments, are utilized to make recommendation in online learning platforms. Furthermore, Tao et al. [27] propose a graph-based method called multimodal graph Attention network (MGAT), which models user preferences with high-order neighboring information and implements an attention mechanism across three modalities (i.e., text, audio, and video).

Zhang et al. [28] introduce a novel model CKE (Collaborative Knowledge Base Embedding) for enhancing recommendation performance. In CKE, three related components are presented to perform item feature embedding via using texts, images and semantic contents in knowledge base, respectively. Further, Sun et al. [29] present a multimodal knowledge graph attention network, named MKGAT, to improve the recommendation performance via multimodal knowledge. It utilizes a multimodal graph attention mechanism to perform information propagation, and then leverages the resulting aggregated embedding for recommendation.

The above-mentioned works usually use a straightforward way to incorporate different modal auxiliary information into the recommendation process. Lately, Huang et al. [30] make an attempt to adequately employ item multimodal auxiliary information to improve recommendation effectiveness. In this study, a multimodal representation learning-based model called MRLM, is proposed, which contains two closely related modules, i.e., global feature representation learning and multimodal feature representation learning. Specifically, the former is introduced to learn global features of items and users by jointly training three tasks: triplet metric learning, Softmax classification, and microscopic verification. While the latter leverages item multimodal auxiliary information to produce item multimodal features through jointly optimizing multiple deterministic modal classifiers. MRLM shows the state-of-the-art performance to our best knowledge.

However, as discussed in Section 1, the MRLM model still has two deficiencies, which may influence accuracy of feature embedding and ultimately affect recommendation effectiveness.

3. Details of the proposed TESM model

Subsection 3.1 presents the problem definition of recommendation with multimodal auxiliary information. Then the two sequential stages GCE and MJFE of TSEM are proposed in Subsections 3.2 and 3.3, respectively. Finally, the training details of TSEM is introduced in Subsection 3.4.

3.1. Problem definition

We assume that there are w users and n items in a target recommendation scenario. Denote the user and item sets as $U = \{u_1, u_2, \cdots, u_w\}$ and $V = \{v_1, v_2, \cdots, v_n\}$, respectively. We employ D to denote the demographic information of U. We employ D_i to denote the basic information of $u_i \in U$and $D = \bigcup_{i=1}^{w} D_i$. D_i includes user u_i's ID, name, age and other basic information. We employ I to denote the descriptive information of V. We employ I_i to denote the descriptive information of $v_i \in V$and $I = \bigcup_{i=1}^{n} I_i$. I_i includes item v_i's ID, name, category and other basic information. Similarly, we employ M to denote the modal auxiliary information of V. We employ M_i to denote the modal auxiliary information of $v_i \in V$and $M = \bigcup_{i=1}^{n} M_i$. M_i includes item v_i's user reviews, pictures from different perspectives, promotional videos, etc. In addition, a user-item interaction matrix is denoted as $\mathrm{R} \in \mathbb{R}^{w \times n}$. If user u_i has interacted with item v_j, then $\mathrm{R}[i, j] = 1$, otherwise $\mathrm{R}[i, j] = 0$.

Like previous works, in this paper, we treat the personalized recommendation problem as a metric learning problem. Hence, a problem in this paper is: Given a user $u_i$and two items $v_j{}^1$ and $v_j{}^2$, the recommendation model can predict the preferences of u_i for $v_j{}^1$ and $v_j{}^2$, respectively, and evaluate which item u_i prefers. Specifically, the training of recommendation model is based on U, V, D, I, M, and $\mathbf{R}$.

3.2. GCE: Graph convolutional embedding

The GCE stage is introduced for learning to construct user and item backbone features via a SGCN network. Fig. 1 shows the overall framework used in GCE.

In the GCE stage, we first generate an undirected bipartite graph$\mathcal{G} = (N, E)$ for user-item interactions based on H. Here, the node set $N = U \cup V$. And the edge set E={$(u, v)|u \in U$ and $v \in V$ and u has interacted with v}. In our study, $\mathcal{G}$ is utilized to depict

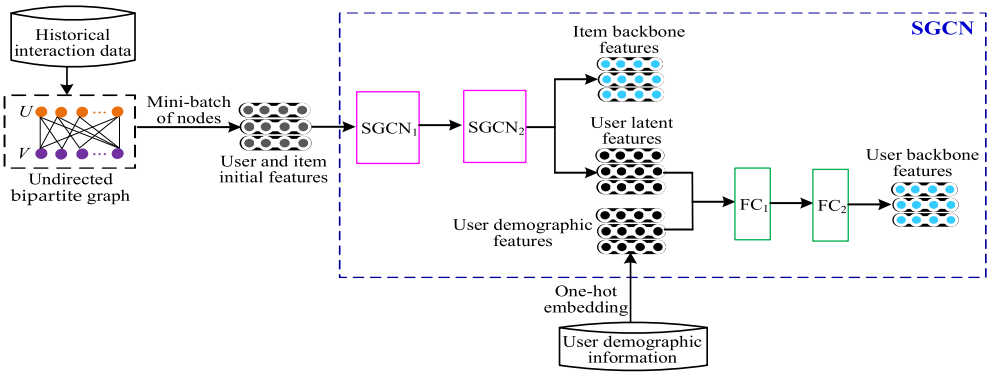


Fig. 1. The overall framework used in the GCE stage.

the correlations between users and items. On this basis, we use the Glorot strategy [36] to randomly initialize the features of all nodes (i.e., users and items) in $\mathcal{G}$:

$$\tilde{\boldsymbol{u}},\ \tilde{\boldsymbol{v}} \leftarrow \mathcal{T}_{\text{udf}}\left(-\sqrt{\frac{6}{\mathrm{d}}}, \sqrt{\frac{6}{\mathrm{d}}}\right) \tag{1}$$

Here, $\mathcal{T}_{udf}()$ is a uniform distribution function and d is the dimensionality of user and item initial features.

As shown in Fig. 1, the SGCN network contains two graph convolutional layers (i.e., $\text{SCGN}_1 \sim \text{SCGN}_2$) and two fully-connected layers (i.e., $\text{FC}_1 \sim \text{FC}_2$). To obtain the input of SGCN, we sample a mini-batch U_b of user nodes from $\mathcal{G}$ at a time. Then based on U_b, we generate a node set S_b and a preference set P_b, which is shown in Algorithm 1. Following previous works, a negative sampling ratio r is set to 4 in the algorithm. Please note that P_b is used for metric learner in the next stage (Subsection 3.3). SGCN takes S_b as input samples and utilizes SCGN_1 and SCGN_2 to produce backbone features of items and latent features of users. Then, users' latent features and demographic features are concatenated and fed to FC_1 and FC_2 for constructing their backbone features.

Algorithm 1: Generating a node set and a preference set

Input: U_b, V and r (negative sampling ratio).
Output: S_b, P_b.
1: $S_b \leftarrow U_b$; $P_b \leftarrow \varnothing$;
2: **for** each $u \in U_b$ **do**
Randomly choose one positive item v^+ from V;
3: $S_b \leftarrow S_b \cup \{v^+\}$; $P_b \leftarrow P_b \cup \{(u\ v^+)\}$;
4: $V_n \leftarrow$ randomly choose r negative items from V;
5: **for** each $v^- \in V_n$ **do**
6: $S_b \leftarrow S_b \cup \{v^-\}$; $P_b \leftarrow P_b \cup \{(u\ v^-)\}$;
7: Return S_b, P_b.

The process of SCGN_1 and SCGN_2 handling each node $s \in S_b$ is shown in Fig. 2. Concretely, for each $s \in S_b$, we obtain its first-hop neighborhood $N_s = \{g_1, g_2, \ldots, g_t\}$, and further obtain the first-hop neighborhood of each $g_i \in N_s$, i.e., $N_{g_i} = \{l_1^{g_i}, l_2^{g_i}, \ldots, l_{m_{g_i}}^{g_i}\}$. Here, t and n_{g_i} are the cardinalities of N_s and N_{g_i}, respectively. Then, inspired by the study of [31,32], in SCGN_1, we produce an aggregation feature $\mathbf{a}_{g_i}$ for each g_i by employing the following aggregation function ($1 \le x \le m_{g_i}$):

$$\mathbf{a}_{g_i} = \mathcal{A}(N_{g_i}) = \sum\nolimits_{l_x^{g_i} \in N_{g_i}} \left(\frac{\text{ReLU}\left(\mathbf{W}_1 \mathbf{l}_x^{g_i} + \mathbf{b}_1\right)}{\mathcal{F}\left(l_x^{g_i}, g_i\right)} \right) \tag{2}$$

where $\mathbf{l}_x^{g_i}$ is the initial feature of $l_x^{g_i}$, ReLU (Rectified Linear Unit) [37] is a nonlinear activation function, and $\mathbf{W}_1$ and $\mathbf{b}_1$ are the two trained parameters. While $\mathcal{F}\left(l_x^{g_i}, g_i\right)$ is the graph Laplacian norm [38]:

$$\mathcal{F}\left(l_x^{g_i}, g_i\right) = \left(deg\left(l_x^{g_i}\right) \cdot deg(g_i)\right)^{1/2} \tag{3}$$

where $deg(g_i)$ is the degree of node g_i. On this basis, we take $\mathbf{a}_{g_i}$ and $\mathbf{g}_i$ (the initial feature of g_i) as an input, and leverage a fully-connected layer FC_1^s to generate g_i's derivation feature $\mathbf{g}_i^d$:

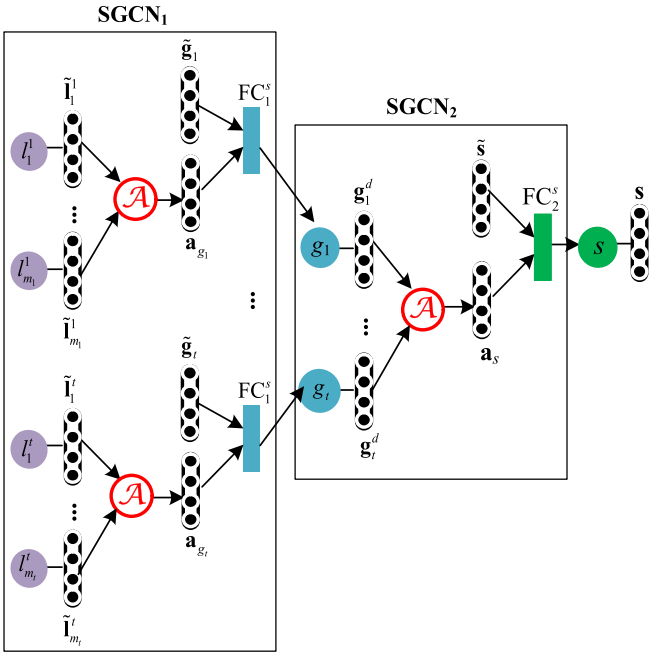


Fig. 2. The process performed by $\text{SCGN}_1 \sim \text{SCGN}_2$ for each $s \in S_u$.

$$\boldsymbol{g}_i^d = \text{ReLU}\left(\boldsymbol{W}_2\left(\boldsymbol{a}_{g_i} \oplus \tilde{\boldsymbol{g}}_i\right) + \boldsymbol{b}_2\right) \tag{4}$$

where '$\oplus$' is a concatenation operation, and $\mathbf{W}_2$ and $\mathbf{b}_2$ are the two trained parameters.

SCGN_1 finally outputs t derivation features for N_z, i.e., $\mathbf{g}_1^d, \mathbf{g}_2^d, \ldots, \mathbf{g}_m^d$. And in SCGN_2, similar to SCGN_1, we apply the function $\mathcal{A}$ on these t derivation features to obtain an aggregation feature $\mathbf{a}_s$ for s. Then, it is combined with $\mathbf{s}$ to produce a semantic feature $\mathbf{s}$ via another fully-connected layer FC_2^s:

$$\mathbf{a}_s = \mathcal{A}(N_s) = \sum\nolimits_{g_i \in N_s} \left(\frac{\text{ReLU}(\mathbf{W}_3 \mathbf{y}_i^d + \mathbf{b}_3)}{\mathcal{F}(g_i, s)} \right) \tag{5}$$

$$\boldsymbol{s} = \text{ReLU}\left(\boldsymbol{W}_4\left(\boldsymbol{a}_s \oplus \tilde{\boldsymbol{s}}\right) + \boldsymbol{b}_4 \right) \tag{6}$$

where $\mathbf{W}_3$, $\mathbf{b}_3$, $\mathbf{W}_4$, and $\mathbf{b}_4$ are the four trained parameters, and $\mathcal{F}(g_i, s)$ is calculated via (3). Note that $\mathbf{s}$ is either an item backbone feature $\mathbf{v}_b$ or a user latent feature $\mathbf{u}_l$.

If $\mathbf{s}$ is a user latent feature (i.e., s is a user node), then we combine $\mathbf{s}$ with the demographic feature $\mathbf{u}_d$ of s, and use $\text{FC}_1 \sim \text{FC}_2$ to construct the backbone feature $\mathbf{u}_b$ of s:

$$\boldsymbol{F}_1 = \text{Sigmoid}(\boldsymbol{W}_5(\boldsymbol{s} \oplus \boldsymbol{u}_d) + \boldsymbol{b}_5) \tag{7}$$

$$\mathbf{u}_b = \text{Sigmoid}(\mathbf{W}_6 \mathbf{F}_1 + \mathbf{b}_6) \tag{8}$$

Here, Sigmoid is a nonlinear activation function [39], and $\mathbf{W}_5$, $\mathbf{b}_5$, $\mathbf{W}_6$, and $\mathbf{b}_6$ are the four trained parameters. Note that $\mathbf{u}_d$ can be obtained by concatenating one-hot features of demographic information of s.

3.3. MJFE: Multimodal joint fuzzy embedding

By utilizing multimodal auxiliary information, the MJFE stage is introduced to construct item semantic-enhanced fuzzy (SEF) features based on item backbone features. Fig. 3 shows the framework used in MJFE.

In this stage, we first obtain a subset V_b of S_b that only contains items. On this basis, for each $v \in V_b$, we obtain its backbone feature $\mathbf{v}_b$ and descriptive feature $\mathbf{v}_d$. Specifically, $\mathbf{v}_d$ can be produced by concatenating one-hot features of descriptive information of v. Then, we use a three-layer CNN-based architecture to construct v's semantic-enhanced fuzzy (SEF) feature $\mathbf{v}_s$, which contains two convolutional layers (i.e., $\text{C}_1 \sim \text{C}_2$) and a fully-connected layer FC. Following the general practice [39], C_1 and

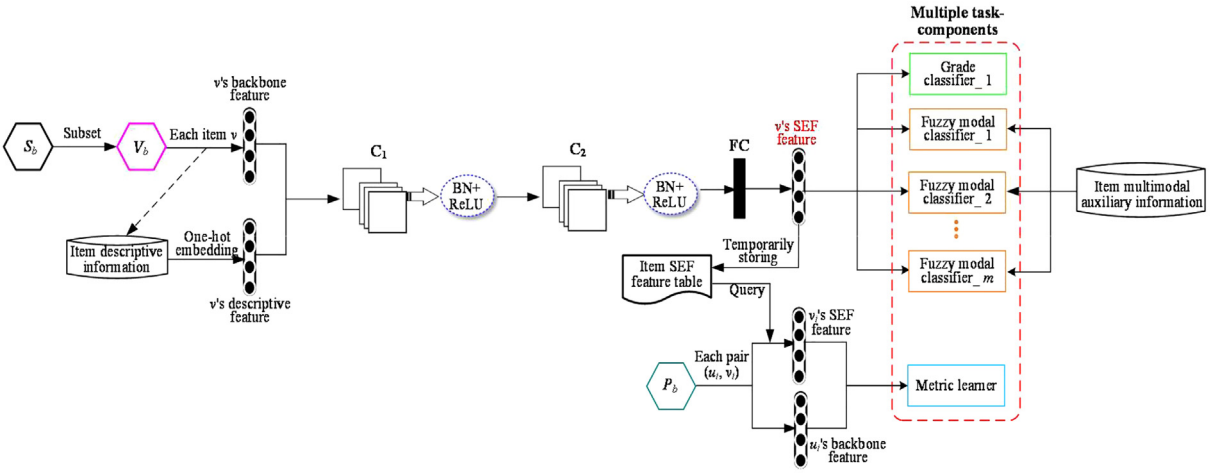


Fig. 3. The framework used in the MJFE stage.

C_2 are followed by a batch-normalization (BN) operation [39] and a ReLU nonlinear activation function. These three layers are defined as:

$$\boldsymbol{C}_1 = \mathrm{ReLU}(\boldsymbol{W}_7 \oplus (\boldsymbol{v}_b \oplus \boldsymbol{v}_d) + \boldsymbol{b}_7) \tag{9}$$

$$\boldsymbol{C}_2 = ReLU(\boldsymbol{W}_8 \oplus \boldsymbol{C}_1 + \boldsymbol{b}_8) \tag{10}$$

$$\mathbf{v}_s = \mathrm{Sigmoid}(\mathbf{W}_9\mathbf{C}_2 + \mathbf{b}_9) \tag{11}$$

Here, '$\otimes$' is a convolution operation, and $\mathbf{W}_7$, $\mathbf{b}_7$, $\mathbf{W}_8$, $\mathbf{b}_8$, $\mathbf{W}_9$, and $\mathbf{b}_9$ are the six trained parameters. After the SEF features of all the items in V_b are constructed, we temporarily store them in a SEF feature table $\mathcal{T}_b$.

Under this architecture, we adopt the idea of multi-task learning [40], and present multiple related task-components that are jointly learned for effectively optimizing user backbone features and item SEF features. In particular, these task-components encompass one metric learner, one deterministic classifier, and m fuzzy modal classifiers. It is worth pointing out that in multi-task learning, multiple tasks are solved jointly, sharing inductive bias between them. Specifically, multi-task learning can employ useful information encompassed in multiple related tasks to help improve the generalization performance of all the tasks.

• *Metric learner*. It is designed for learning user preferences on items. That is, it minimizes Euclidean distances between users and positive items, and maximizes Euclidean distances between users and negative items. Thus, the loss function used in metric learner is defined as:

$$\mathcal{L}_{lr} = \frac{1}{|P_b|}\sum_{(u,v)\in P_b}\left(\begin{cases} ||\mathbf{u}_b - \mathbf{v}_s||_2, v \text{ is a positive item} \\ -||\mathbf{u}_b - \mathbf{v}_s||_2, v \text{ is a negative item} \end{cases}\right) \tag{12}$$

where $|P_b|$is the size of P_b, and $||\mathbf{u}_b - \mathbf{v}_s||_2$ is the Euclidean distance between u and v. Note that $\mathbf{v}_s$ is obtained from the SEF feature table $\mathcal{T}_b$.

• *Deterministic classifier*. It is a grade classifier, which is introduced to identify items' different grades and has two trained parameters $\mathbf{Q}_0$ and $\mathbf{q}_0$. In this study, items' grades have five classes, i.e., excellent, good, medium, general, and poor. Specifically, the grade classifier employs a deterministic Softmax layer that has five different output-classes, corresponding to five grades of items, respectively. Meanwhile, the probability value of each output-class is calculated by using a Softmax function. Thus, the loss function of grade classifier is defined as:

$$\mathcal{L}_{gc} = \frac{1}{|V_b|}\sum_{v\in V_b}\left(-p(v|g_v) + log\sum\nolimits_{i=1}^{5}\mathbf{e}^{p(v|g_i)}\right) \tag{13}$$

where g_v is the true corresponding grade of v, $p(v|g_v)$ is the probability value of v on g_v, and $p(v|g_i)$ is the probability value of v on each of five grades g_i.

• *Fuzzy modal classifiers*. Each fuzzy modal classifier corresponds to a modality of item. For the i-th modality ($1 \le i \le m$), we first use existing feature extraction methods to obtain the auxiliary feature of each item in V_b. Then, we use fuzzy c-means (FCM) clustering algorithms [33–35] to perform fuzzy clustering on all obtained items' auxiliary features. The implementation details is described as follow. Let ρ=$|V_b|$.

We first select two positive integers c_i^{min} and c_i^{max} satisfying$c_i^{min} < c_i^{max}$. And let c_i^{min} and c_i^{max} be the minimal and the maximal thresholds of fuzzy clusters, respectively. Then for each$c_i \in [c_i^{min}, c_i^{max}]$, we randomly initialize c_i fuzzy cluster centers c_i-*FC* =$\{\mathbf{v}_i^{(1)}, \mathbf{v}_i^{(2)}, \ldots, \mathbf{v}_i^{(c_i)}\}$, and carry out the following steps:

(*i*) Calculate the membership matrix c_i-M=$(m_{xy})_{\rho\times c}$:

$$m_{xy} = \begin{cases} \left(\sum_{z=1}^{c_i} \left(\frac{\| \boldsymbol{v}_{i.x} - \boldsymbol{v}_i^{(y)} \|_2}{\| \boldsymbol{v}_{i.x} - \boldsymbol{v}_i^{(z)} \|_2} \right)^{\frac{2}{\tau-1}} \right)^{-1} & if\ \forall z,\ \| \boldsymbol{v}_{i.x} - \boldsymbol{v}_i^{(z)} \|_2 > 0 \\ 1, & if\ \| \boldsymbol{v}_{i.x} - \boldsymbol{v}_i^{(y)} \|_2 = 0 \\ 0, & if\ \exists z \neq y, \| \boldsymbol{v} v_{i.x} - \boldsymbol{v}_i^{(z)} \|_2 = 0 \end{cases} \quad (14)$$

where $x \in [1,\ \rho]$, $y \in [1, c_i]$, $\mathbf{v}_{i.x}$ is the auxiliary feature of the x-th item in $\boldsymbol{V}_b$, and τ is a fuzzifier parameter.

On this basis, we normalize c_i-**M**. That is, for each $m_{xy} \in c_i$-**M**, its normalized value is: $n\_m_{xy} = m_{xy} / \sum_{t=1}^{\rho} \sum_{q=1}^{c_i} m_{tq}$. We can easily have: $\sum_{x=1}^{\rho} \sum_{y=1}^{c_i} n\_m_{xy}$=1. And for convenience, we still use m_{xy} to represent the normalized value in the following parts.

(*ii*) Update the centers c_i-*FC*'=$\{\mathbf{v}'^{(1)}_i, \mathbf{v}'^{(2)}_i, \ldots, \mathbf{v}'^{(c_i)}_i\}$. And for each $y \in [1,\ c_i]$, $\mathbf{v}'^{(y)}_i$ is expressed as:

$$\mathbf{v}'^{(y)}_i = \frac{\sum_{x=1}^{\rho} (m_{xy})^{\tau} \mathbf{v}_{i.x}}{\sum_{x=1}^{\rho} (m_{xy})^{\tau}} \quad (15)$$

(*iii*) Calculate the objective function values before and after updating centers, respectively:

$$f(c_i - \mathbf{M}, c_i - FC) = \sum_{x=1}^{\rho} \sum_{y=1}^{c_i} (m_{xy})^{\tau} \| \mathbf{v}_{i.x} - \mathbf{v}_i^{(y)} \|_2^2 \quad (16)$$

$$f(c_i - \mathbf{M}, c_i - FC') = \sum_{x=1}^{\rho} \sum_{y=1}^{c_i} (m_{xy})^{\tau} \| \mathbf{v}_{i.x} - \mathbf{v}'^{(y)}_i \|_2^2 \quad (17)$$

(*iv*) Given a convergence threshold ζ, we evaluate the value df=|f(c_i-**M**, c_i-*FC*')-f(c_i-**M**, c_i-*FC*)|. If $df \leq \zeta$, then return the fuzzy cluster result containing c_i-*FC*' and c_i-**M**; otherwise let c_i-*FC* = c_i-*FC*', and go to Step (*i*).

Thus, we can obtain (c_i^{max}-c_i^{min} + 1) fuzzy cluster results: <c_i^{min}-*FC*, c_i^{min}-**M**>, <c_i^{min+1}-*FC*, c_i^{min+1}-**M**>,. . ., and < c_i^{max}-*FC*, c_i^{max}-**M** > . Then, we utilize a validity index *VI*(<c_i-*FC*, c_i-**M** >) to evaluate the quality of fuzzy partition. And the smaller *VI*(<c_i-*FC*, c_i-**M** >), the better the fuzzy partition. Based on [35], *VI*(<c_i-*FC*, c_i-**M** >) is defined as:

VI(<c_i-*FC*, c_i-**M** >) = *C*(<c_i-*FC*, c_i-**M** >)-*S*(<c_i-*FC*, c_i-**M** >). (18)

In (18), *C*(<c_i-*FC*, c_i-**M** >) is a compactness measure that is defined as:

$$C(<c_i\text{-}FC,\ c_i\text{-}\mathbf{M}>) = \left(\frac{c_i+1}{c_i-1}\right)^2 \cdot \sum_{y=1}^{c_i} \sum_{x=1}^{\rho} (\mathcal{R}) \cdot \sum_{x=1}^{\rho} \frac{m_x}{\rho}. \quad (19)$$

Here, $\mathcal{R} = 1 - \exp\left(- \frac{\rho \| \mathbf{v}_{i.x} - \mathbf{v}_i^{(y)} \|_2^2}{\sum_{x=1}^{\rho} \| \mathbf{v}_{i.x} - \overline{\mathbf{v}}_i \|_2^2} \right)^{1/2}$ and $\overline{\mathbf{v}}_i = \frac{\sum_{t=1}^{\rho} \mathbf{v}_i}{\rho}$. And *S*(<$c_i$-*FC*, c_i-**M** >) is a separation measure that can be defined as:

$$S(< c_i - FC, c_i - \mathbf{M} >) = \frac{\sum_{x=1}^{\rho} \left(\sum_{t=1}^{c_i-1} \sum_{z=t+1}^{c_i-1} \mathcal{J}_{t,z,x}(c_i, c_i - \mathbf{M}) \right)}{\rho} \quad (20)$$

where $\mathcal{J}_{t,z,x}(c_i, c_i - \mathbf{M})$ is the degree of separation between two fuzzy clusters t and z for the x-th given auxiliary feature. Specifically, it is defined as:

$$\mathcal{J}_{t,z,x}(c_i, c_i - \mathbf{M}) = \begin{cases} 1 - |m_{tx} - m_{zx}|, if |m_{tx} - m_{zx}| > t_{sm}, t \neq z \\ 0, \text{otherwise} \end{cases} \quad (21)$$

where t_{sm} is a separation threshold.

After the above processing, for items in V_b, we can obtain c_i fuzzy clusters for their auxiliary features on the i-th modality, denoted as FC_1, FC_2,. . ., FC_{c_i}. Based on this, the i-th fuzzy modal classifier utilizes a fuzzy Softmax layer, which involves two trained parameters $\mathbf{Q}_i$ and $\mathbf{q}_i$, and has c_i output-classes, corresponding to c_i fuzzy clusters, respectively. The probability value of each output-class can be calculated via a fuzzy Softmax function. Thus, the loss function of this classifier is defined as:

$$\mathcal{L}_{f.i} = \frac{c_i - 1}{|V_b|} \sum\nolimits_{v \in V_b} \left(\sum_{x=1}^{c_i} p(v|FC_x) m_{\varphi(v)x} \right) \quad (22)$$

Here, $p(v|FC_x)$ is the probability value of the current item v on the output-class (i.e., fuzzy cluster) FC_x, $\varphi(v)$ is the index position of v in V_b, and $m_{\varphi(v)x} \in c_i$-**M** is a degree of membership that equals the value of the $\varphi(v)$-th row and x-th column in c_i-**M**.

Thereby, the joint loss function of all the $(m + 2)$ task-components can be defined as:

$$\mathcal{L}_{TESM} = \lambda_1 \mathcal{L}_{lr} + \lambda_2 \mathcal{L}_{gc} + \lambda_3 \mathcal{L}_{f.1} + \cdots + \lambda_{m+2} \mathcal{L}_{f.m} \quad (23)$$

where $\lambda_1 \sim \lambda_{m+2} \in (0,\ 1)$ are the trained parameters that are used to control the importance of each task-component, and

satisfy $\sum_{t=1}^{m+2}\lambda_t = 1$.

3.4. Overall training of TESM

In the proposed model TESM, all the trained parameters form a parameter-set ψ_{all}. That is, ψ_{all}= {$\mathbf{W}_1 \sim \mathbf{W}_9$, $\mathbf{b}_1 \sim \mathbf{b}_9$, $\lambda_1 \sim \lambda_{m+2}$, $\mathbf{Q}_0 \sim \mathbf{Q}_m$, $\mathbf{q}_0 \sim \mathbf{q}_m$}. And we randomly initialized each parameter with a normal distribution of $\mathcal{N}(0,1)$.

We use a mini-batch stochastic gradient descent (SGD) method [39] to minimize the loss function $\mathcal{L}_{TESM}$. And an RMSprop optimizer [41] is utilized for updating each parameter. As an example, at the t-th iteration, RMSprop update $\mathbf{W}_1$ as follows:

$$\begin{cases} \varpi_t \leftarrow 0.9\varpi_{t-1} + 0.1\left(\frac{\partial \mathcal{L}_{TESM}}{\partial [\mathbf{W}_1]_t}\right)^2 \\ [\mathbf{W}_1]_{t+1} \leftarrow [\mathbf{W}_1]_t - \left(\frac{\sigma}{\sqrt{\varpi_t}+\epsilon}\right)\frac{\partial \mathcal{L}_{TESM}}{\partial [\mathbf{W}_1]_t} \end{cases} \tag{24}$$

where ϖ is a gradient cumulative variable, σ is an initial learning rate, and $\epsilon = 10^{-8}$ is a constant to ensure a non-zero denominator.

Then, based on the stages GCE and MJFE, the training procedure of TESM can be described in Algorithm 2. Line 1 generates an undirected bipartite graph $\mathcal{G}$ by using H, U and V. Line 2 initializes the features of all users and items, and Line 3 initializes each trained parameter in ψ_{all}. Via Lines 5–7, we can get four sets: a user-node set U_b, a node set S_b, a preference set P_b, and an item-node set V_b. Specifically, S_b and P_b are generated by employing Algorithm 1. Lines 8–10 construct backbone features of all the users in U_b, and Lines 11–13 construct SEF features of all the items in V_b. Lines 14–18 construct $m+2$ task-components, and Line 19 calculates the joint loss function $\mathcal{L}_{TESM}$ of these task-components. Line 20 then updates each parameter in ψ_{all} via RMSprop optimizer. Once the model converges, Algorithm 2 will return the optimal parameter-set ψ_{all}.

Algorithm 2: Overall training of TESM

Input: historical interaction dataset is H, user demographic information D, item descriptive information I, item multimodal auxiliary information M, user set U, item set V, min-batch size b, negative sampling ratio r.
Output: the parameter-set ψ_{all}.

1: Construct $\mathcal{G}$ based on H, U and V;
2: Initialize the features of all nodes in $\mathcal{G}$ according to (1);
3: Initialize ψ_{all}: $\psi_{all} \leftarrow \mathcal{N}(0,1)$;
4: **repeat**
5: Sample a mini-batch U_b of user nodes from $\mathcal{G}$;
6: Generate S_b and P_b via Algorithm 1(U_b, V, r);
7: $V_b \leftarrow S_b$-U_b;
8: **for** each $u \in U_b$ **do**
9: Generate $\mathbf{u}_l$ according to (2)-(6);
10: Construct $\mathbf{u}_b$ based on $\mathbf{u}_l$ and D according to (7)-(8);
11: **for** each $v \in V_b$ **do**
12: Generate $\mathbf{v}_b$ according to (2)-(6);
13: Construct $\mathbf{v}_s$ based on $\mathbf{v}_b$ and I according to (9)-(11);
14: Construct the metric learner based on P_b and $\{\mathbf{u}_b\} \cup \{\mathbf{v}_s\}$;
15: Construct the grade classifier based on V_b and $\{\mathbf{v}_s\}$;
16: **for** $i = 1 \sim m$ **do**
17: Generate c_i fuzzy clusters on the i-th modality based on M according to (14)-(21);
18: Construct the i-th fuzzy modal classifier based on V_b and $\{\mathbf{v}_s\}$;
19: Calculate $\mathcal{L}_{TESM}$ according to (12), (13), and (23);
20: Calculate the partial derivative of each parameter in ψ_{all} and update it via RMSprop optimizer;
21: **until** the model converges
22: Return ψ_{all}.

Complexity analysis. Assume that there are $\mathcal{N}_u$ users, $\mathcal{N}_v$ items, and $\mathcal{N}_{in}$ user-item interactions in a given recommender system. And on average, $\overline{\mathcal{N}}_u$ users have interacted with the same item, and $\overline{\mathcal{N}}_v$ items have interacted with the same user ($\overline{\mathcal{N}}_u \ll \mathcal{N}_u$, $\overline{\mathcal{N}}_v \ll \mathcal{N}_v$). From Algorithm 2, we can see that the computational complexity of training TESM mainly consists of five parts:

$O(\mathcal{N}_{in})$ for generating $\mathcal{G}$;
$O(\mathcal{N}_u + \mathcal{N}_v)$ for initializing the features of all nodes;
$O\left((\mathcal{N}_u + \mathcal{N}_v)\overline{\mathcal{N}}_u\overline{\mathcal{N}}_v\right) = O\left(\mathcal{N}_{in}\left(\overline{\mathcal{N}}_u + \overline{\mathcal{N}}_v\right)\right)$ for constructing users' backbone features and items' SEF features;

$O\left(\mathcal{N}_{in} + m\mathcal{N}_v{}^2\right)$ for performing the task-components and calculating the joint loss function;
$O(\mathcal{N}_{in})$ for updating the model parameters.

Based on the above analysis, we have the complexity of $O\left(\mathcal{N}_{in}\left(\overline{\mathcal{N}}_u + \overline{\mathcal{N}}_v\right) + \mathcal{N}_u + m\mathcal{N}_v{}^2\right)$ to train TESM.

Recommendation utilizing TESM. When the training of TESM is completed, for a given user $u \in U$ and an item set $A=\{v_1, v_2, \ldots, v_a\} \subseteq V$, we first obtain u's backbone feature $\mathbf{u}_b$ and the SEF feature set corresponding to A: $\overrightarrow{A}=\{\mathbf{v}_s^1, \mathbf{v}_s^2, \cdots, \mathbf{v}_s^a\}$. Then, for each $\mathbf{v}_s^i \in \overrightarrow{A}$, we calculate the predictive rating of $(\mathbf{u}_b, \mathbf{v}_s^i)$ via Euclidean distance. Finally, we select the k ($k < a$) items with the largest predictive ratings as recommendation.

4. Experiments

In this section, we perform an experimental evaluation of TESM with two real-world datasets.

4.1. Experimental settings

Two real-world datasets are used in experiments: Movielens-20 M[2] and BookCrossing[3]:

- *MovieLens-20 M*. It is one of the most widely utilizing datasets in recommender system domain, encompassing the information: users, movies, and users' rating on items. In experiments, according to previous works, the ratings greater than or equal to 4 are treated as positive feedback, and the ratings less than 4 are treated as negative feedback. Only the users with more than 20 ratings are considered in experiments.
- *BookCrossing*. It is a prevalent book dataset, encompassing the information: users, books, and users' rating on books. Following previous works, the ratings greater than or equal to 5 are treated as positive feedback, and the ratings less than 5 are treated as negative feedback.

In MovieLens-20 M, auxiliary information of four modalities is introduced for each movie: (a) For text modality, the text summary extracted from its plot is employed; (b) For image modality, its poster image is employed; (c) For video modality, its promotional video is utilized; and (d) For knowledge-base modality, its directly adjacent entities and relationships are used in KB4Rec [41]. In BookCrossing, auxiliary information of two modalities is introduced for every book: (a) For text modality, its brief introduction is used; (b) For image modality, its front cover image is used. Furthermore, BERT [42], ResNet-50 [43], ECNN [44], and TransG [45] are leveraged to implement feature extraction for the modalities of text, image, video, and knowledge-base, respectively. Please note that to improve training efficiency, for each modality, we perform fuzzy clustering on the auxiliary features of all items in advance, rather than on the auxiliary features of a mini-batch of items at a time.

Table 1 shows the statistical data of two real-world datasets used in the experiments.

In experiments, the datasets are randomly divided into training (70%), validation (20%), and test (10%) sets. We perform experimental evaluation on TensorFlow platform [39], and use RMSprop optimizer to update each parameter. We carry out hyper-parameter tuning on validation sets to choose the optimal value for each hyper-parameter via random search method [46].

For MovieLens-20 M, the dimensionality of features of all users and items $d = 200$, the mini-batch size $b = 128$, and the negative sampling ratio $r = 4$; FC_1 and FC_2 contain 300 and 200 neurons, respectively; C_1 has 9 kernels of size 1×3 with stride of 3, and C_2 has 5 kernels of size 1×3 with stride of 3; FC contains 200 neurons; and the initial learning rate$\sigma = 0.001$. While for BookCrossing, d, b, and r are set to 150, 128, and 4, respectively; FC_1 and FC_2 contain 250 and 150 neurons, respectively; C_1 has 7 kernels of size 1×3 with stride of 3, and C_2 has 5 kernels of size 1×3 with stride of 3; FC contains 150 neurons; and$\sigma = 0.001$.

As for evaluation metrics, following previous works, we utilize two well-known metrics Recall@n and AUC@n [47,48], which are widely applied for top-n recommendation evaluation. AUC represents the area under receiver operating characteristic (ROC) curve. Recall represents the percentage of correctly predicted true positive items in the samples:

$$\text{Recall} = TP/(TP + FN) \tag{25}$$

where TP is the number of positive items that are correctly predicted to be true, and FN is the number of positive items that are falsely predicted to be false.

To verify effectiveness of TESM, we compare it with eleven state-of-the-art models, i.e., DMFL [11], NGCF [32], HUAP [17], MF-VMLP [20], AMNN [21], ORAMAS [22], AMV [24], MGAT [27], CKE [28], MKGAT [29], and MRLM [30]. For comparison, the extended versions of the eight models are considered, which utilize all four modalities via a straightforward way proposed in

[2] https://grouplens.org/datasets/movielens/20m
[3] https://grouplens.org/datasets/book-crossing

Table 1
The statistics of two real-world datasets used in experiments.

	Movielens-20 M	BookCrossing
Users	138,493	278,858
Items	27,278	271,379
Ratings	20,000,263	1,149,780
Ratings per user	144.4	4.1
Ratings per item	733.2	4.2
Items having texts	26,012	259,385
Items having imagesItems having videosItems in KB	27,08424,61625,982	263,19400

Section 1. These extended models are denoted as DMFL*, NGCF*, HUAP*, MF-VMLP*, AMNN*, ORAMAS*, AMV*, MGAT*, CKE*, MKGAT*, and MRLM*, respectively. Since MRLM involves all four modalities, MRLM* is the same as MRLM. Here, we only present the experimental results of MRLM.

4.2. Performance comparison with baselines

We compare the effectiveness of TSEM with its peers (i.e., 21 baselines). Tables 2 and 3 show the values of Recall@n and AUC@n for the two datasets with n={5, 10, 15, 20}.

From Tables 2 and 3, we are able to observe that TSEM has superior recommendation accuracy than all the twenty-one baselines. For example, in Table 2, TSEM outperforms eleven base models DMFL, NGCF, HUAP, MF-VMLP, AMNN, ORAMAS, AMV, MGAT, CKE, MKGAT, and MRLM by 96.35%, 71.72%, 67.33%, 58.90%, 56.46%, 63.85%, 23.29%, 19.79%, 40.87%, 25.25%, and 9.10%, respectively, for Recall@20 on MovieLens-20 M. And it outperforms ten extended models DMFL*, NGCF*, HUAP*, MF-VMLP*, AMNN*, ORAMAS*, AMV*, MGAT*, CKE*, and MKGAT* by 83.05%, 47.81%, 39.94%, 35.82%, 32.58%, 4.96%, 13.59%, 32.30%, and 17.52%, respectively. The main reasons are two-folds: (*i*) First, by using the SGCN network in the GCE stage, TSEM is able to construct user and item backbone features adequately. Specifically, for each user and each item, TSEM can effectively capture the influence of other related users and items on its feature embedding. (*ii*) More important, by jointly learning six related tasks in the MJFE stage, TSEM is able to obtain user backbone features and item SEF features accurately. In particular, among these learning tasks, four fuzzy modal classifiers can effectively refine item backbone features to generate SEF features.

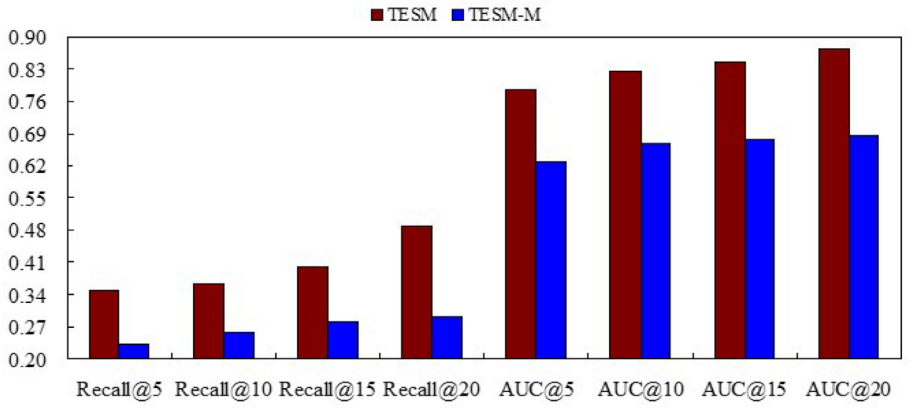
Meanwhile, we can observe that for every base model, its extended version slightly outperforms it in most cases. This observation is consistent with that in the work [30]. For example, in Table 3, DMFL* outperforms DMFL by 4.89% for AUC@20 on BookCrossing. The main reason is that for every base model, its extended version integrates different modal auxiliary information into recommendation process via a straightforward way, thus slightly increasing the accuracy of item

Table 2
The Recall@n values of TESM and 21 baselines (best results are bold-faced).

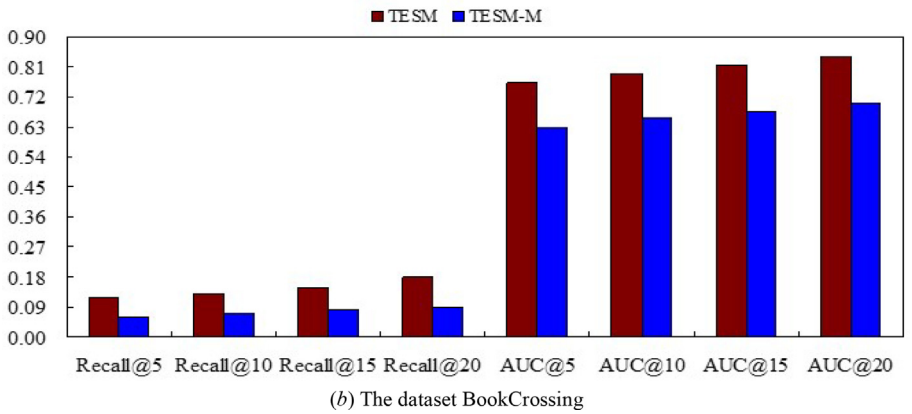
Model	Dataset							
	MovieLens-20 M				BookCrossing			
	n = 5	n = 10	n = 15	n = 20	n = 5	n = 10	n = 15	n = 20
DMFL	0.1742	0.2035	0.2250	0.2491	0.0367	0.0413	0.0481	0.0565
NGCF	0.2253	0.2514	0.2776	0.2848	0.0513	0.0605	0.0716	0.0853
HUAP	0.2321	0.2608	0.2815	0.2923	0.0590	0.0728	0.0811	0.0927
MF-VMLP	0.2359	0.2662	0.2891	0.3078	0.0654	0.0812	0.0993	0.1092
AMNN	0.2371	0.2704	0.2969	0.3126	0.0682	0.0859	0.1047	0.1161
ORAMAS	0.2340	0.2633	0.2854	0.2985	0.0639	0.0798	0.0931	0.1015
AMV	0.2784	0.3072	0.3419	0.3967	0.0765	0.0971	0.1086	0.1242
MGAT	0.2858	0.3165	0.3511	0.4083	0.0846	0.1028	0.1153	0.1336
CKE	0.2697	0.2929	0.3246	0.3472	0.0931	0.1026	0.1208	0.1431
MKGAT	0.2769	0.3041	0.3374	0.3905	0.0992	0.1071	0.1285	0.1527
MRLM	0.3327	0.3516	0.3952	0.4483	0.1108	0.1204	0.1390	0.1638
DMFL*	0.2015	0.2251	0.2495	0.2672	0.0430	0.0505	0.0593	0.0681
NGCF*	0.2501	0.2807	0.3006	0.3309	0.0616	0.0693	0.0879	0.1102
HUAP*	0.2526	0.2842	0.3063	0.3495	0.0854	0.1008	0.1102	0.1315
MF-VMLP*	0.2605	0.2936	0.3261	0.3601	0.0917	0.1021	0.1153	0.1398
AMNN*	0.2654	0.2998	0.3353	0.3689	0.0682	0.0859	0.1047	0.1161
ORAMAS*	0.2582	0.2913	0.3208	0.3594	0.0905	0.1019	0.1101	0.1390
AMV*	0.3119	0.3402	0.3795	0.4306	0.0948	0.1035	0.1243	0.1496
MGAT*	0.3194	0.3497	0.3881	0.4402	0.1005	0.1149	0.1321	0.1564
CKE*	0.2738	0.3005	0.3376	0.3697	0.0931	0.1026	0.1208	0.1431
MKGAT*	0.2821	0.3139	0.3495	0.4162	0.0992	0.1071	0.1285	0.1527
TESM	**0.3495**	**0.3638**	**0.4012**	**0.4891**	**0.1174**	**0.1292**	**0.1485**	**0.1801**

Table 3
The AUC@*n* values of TESM and 21 baselines (best results are bold-faced).

Model	Dataset							
	MovieLens-20 M				BookCrossing			
	n = 5	*n* = 10	*n* = 15	*n* = 20	*n* = 5	*n* = 10	*n* = 15	*n* = 20
DMFL	0.5562	0.5814	0.5928	0.6013	0.5631	0.5794	0.5912	0.6109
NGCF	0.5971	0.6307	0.6404	0.6510	0.5755	0.5936	0.6141	0.6337
HUAP	0.6308	0.6725	0.6801	0.6892	0.6309	0.6652	0.6805	0.7030
MF-VMLP	0.6423	0.6892	0.7026	0.7189	0.6392	0.6715	0.6943	0.7118
AMNN	0.6519	0.6993	0.7122	0.7285	0.6451	0.6783	0.7002	0.7146
ORAMAS	0.6346	0.6785	0.6891	0.7013	0.6308	0.6601	0.6818	0.7025
AMV	0.7093	0.7518	0.7675	0.7902	0.6471	0.6883	0.7047	0.7171
MGAT	0.7165	0.7592	0.7761	0.7998	0.6563	0.6949	0.7101	0.7218
CKE	0.6952	0.7321	0.7480	0.7705	0.6615	0.7028	0.7152	0.7263
MKGAT	0.7017	0.7392	0.7541	0.7782	0.6652	0.7067	0.7214	0.7326
MRLM	0.7405	0.7862	0.8049	0.8237	0.7259	0.7611	0.7729	0.7895
DMFL*	0.6004	0.6219	0.6335	0.6458	0.5804	0.5960	0.6215	0.6408
NGCF*	0.6367	0.6651	0.6760	0.6927	0.5950	0.6139	0.6390	0.6651
HUAP*	0.6912	0.7185	0.7298	0.7411	0.6592	0.7027	0.7124	0.7319
MF-VMLP*	0.7105	0.7342	0.7473	0.7622	0.6696	0.7101	0.7159	0.7194
AMNN*	0.7132	0.7361	0.7509	0.4625	0.6451	0.6783	0.7002	0.7146
ORAMAS*	0.7028	0.7286	0.7382	0.7617	0.6651	0.7074	0.7148	0.7169
AMV*	0.7309	0.7663	0.7827	0.8014	0.6805	0.7251	0.7426	0.7652
MGAT*	0.7392	0.7759	0.7926	0.8121	0.6864	0.7305	0.7481	0.7728
CKE*	0.7143	0.7394	0.7561	0.7798	0.6615	0.7028	0.7152	0.7263
MKGAT*	0.7215	0.7481	0.7656	0.7893	0.6652	0.7067	0.7214	0.7326
TESM	**0.7845**	**0.8247**	**0.8452**	**0.8731**	**0.7619**	**0.7896**	**0.8160**	**0.8413**



(*a*) The dataset MovieLens-20M



(*b*) The dataset BookCrossing

Fig. 4. Performance evaluation for TESM and TESM-M.

feature embedding. Please note that in Tables 2 and 3, for the three models AMNN, CKE, and MKGAT, their accuracy is equal to their corresponding extended models AMNN*, CKE*, and MKGAT* on BookCrossing. It is mainly since that AMNN, CKE, and MKGAT have used items' text and image modal auxiliary information in BookCrossing. Therefore, AMNN*, CKE*, and MKGAT* degenerates to AMNN, CKE, and MKGAT, respectively, on BookCrossing.

Moreover, we find that all twenty-two models have higher recommendation accuracy on MovieLens-20 M than on BookCrossing. For example, in Table 2, the average Recall@20 values of these twenty-two models on MovieLens-20 M and BookCrossing are 0.3599 and 0.1257, respectively. A possible explanation is that the average number of user-item interactions in MovieLens-20 M is greater than that in BookCrossing, which enables these models to learn user and item feature embedding more accurately.

4.3. Performance study for item semantic-enhanced fuzzy features

The overall performance comparison demonstrates that our TESM model has a higher recommendation effectiveness than twenty-one baselines. For further understanding the importance of item SEF features, we perform an "ablation" study in this subsection. First, we compare TESM with its simplified version TESM-M representing that four fuzzy modal classifiers are not used in the MJFE stage, i.e., only a metric learner and a deterministic classifier are used. Fig. 4 shows the values of Recall@n and AUC@n for the two datasets with n={5, 10, 15, 20}.

Fig. 4 clearly presents that the complete TESM model outperforms TESM-M on both two datasets. For example, in the figure, we are able to observe that TESM outperforms TESM-M by 67.79% and 101.68% for Recall@20 on MovieLens-20 M and BookCrossing, respectively. It shows that the recommendation results are not satisfactory if we only employ a metric learner and a deterministic classifier, and do not leverage four fuzzy modal classifiers. And we can substantively improve the final recommendation effectiveness through incorporating item multimodal auxiliary information.

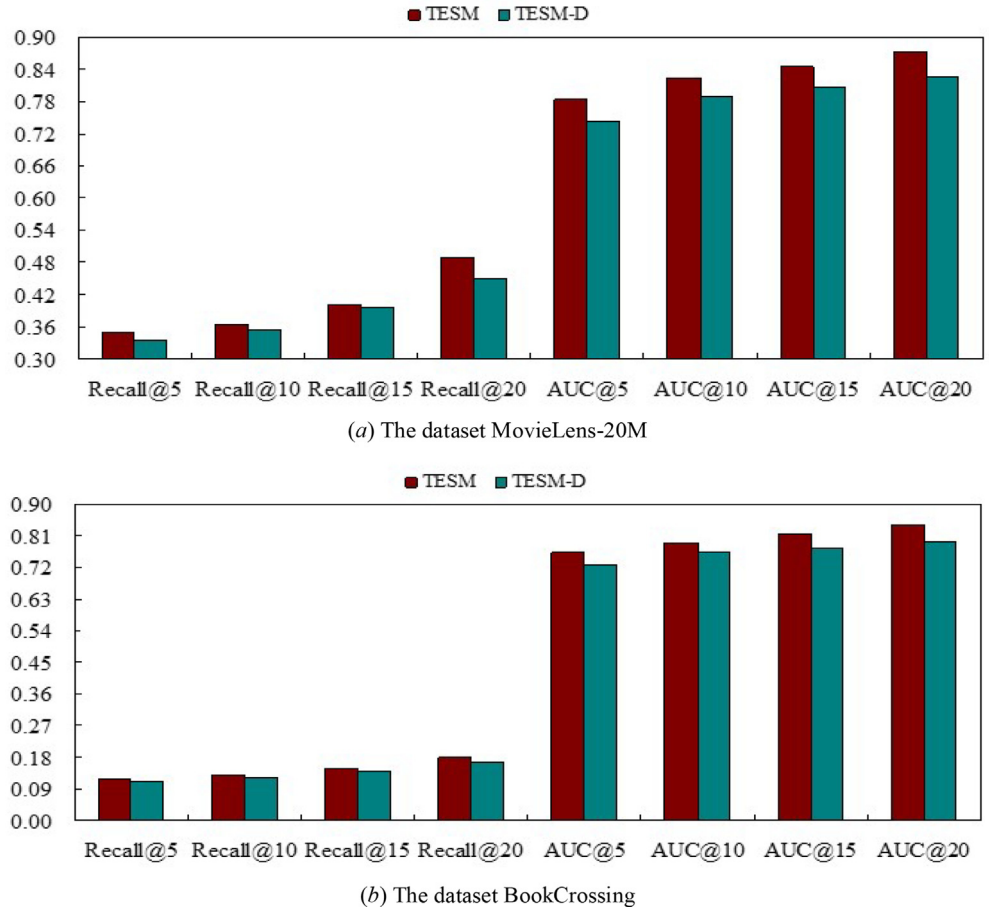


Fig. 5. Performance evaluation for TESM and TESM-D.

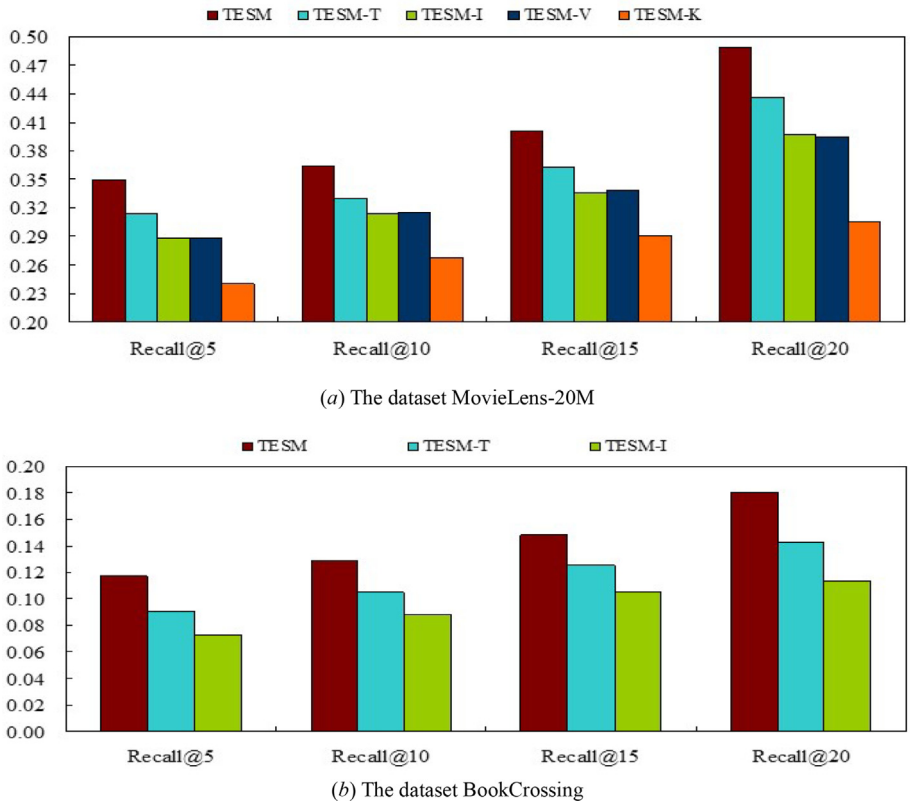


Fig. 6. Experimental study for the importance of different modalities.

Then, we compare TESM with its another simplified version TESM-D representing that four fuzzy modal classifiers are replaced with corresponding deterministic ones presented in [30]. Fig. 5 shows the values of Recall@n and AUC@n for the two datasets with n={5, 10, 15, 20}.

Similarly, from Fig. 5, we are able to observe that the complete TESM model outperforms TESM-D on both two datasets. For example, in the figure, TESM outperforms TESM-D by 5.66% and 6.29% for AUC@20 on MovieLens-20 M and BookCrossing, respectively. The main reason is that compared with deterministic modal classifiers, fuzzy ones are able to make better use of item multimodal auxiliary information, thus learning item feature embedding more accurately.

Finally, we focus on investigating the importance of different modalities and compare TESM with its four related variants:

- TESM-T: only text modality is utilized in TESM;
- TESM-I: only image modality is utilized in TESM;
- TESM-V: only video modality is utilized in TESM;
- TESM-K: only knowledge-base modality is utilized in TESM.

Fig. 6 reports the values of Recall@n for the two datasets with n={5, 10, 15, 20}. We observe a similar accuracy trend for AUC@n values and omit them here due to space limitation. Please note that BookCrossing does not have video and knowledge base modalities, therefore TESM-V and TESM-K have no experimental results on it.

From Fig. 6, we can clearly see that the complete TESM model has superior recommendation accuracy than all its four variants over both two datasets. For example, in Fig. 6 (a), TESM outperforms TESM-T, TESM-I, TESM-V and TESM-K by 12.19%, 23.23%, 24.02%, and 59.86%, respectively, for Recall@20 on MovieLens-20 M. It indicates that item auxiliary information over each modality contributes to final recommendations, and the joint use of item multimodal auxiliary information can produce the best effectiveness. Meanwhile, we find that compared with other variants, TESM-T achieves superior recommendation accuracy. For example, in Fig. 6 (a), the Recall@10 values of TESM-T, TESM-I, TESM-V and TESM-K are 0.3301, 0.3137, 0.3152, and 0.2675, respectively, on MovieLens-20 M. A possible reason is that in either of two datasets, item's text modality contains more useful semantic information than item's other modalities for final recommendations.

5. Conclusions

This paper introduces a novel model TESM for improving the recommendation effectiveness. It fully exploits item multimodal auxiliary information and includes two sequential stages. In the GCE stage, we first obtain a user-item interaction graph that is then combined with user demographic information to construct user and item backbone features via a SGCN network. While in the MJFE stage, we first employ item backbone features and descriptive information for constructing item SEF features via a three-layer CNN-based architecture. Then, six related task-components are simultaneously optimized to obtain user backbone features and item SEF features accurately. Specifically, four fuzzy classifiers are

obtain user backbone features and item SEF features accurately. Specifically, four fuzzy classifiers are proposed by jointly using item multimodal auxiliary information. Experimental results over two real-world datasets show the effectiveness of our TESM model.

In the future, we will continue to improve recommendation performance of our TESM model from two aspects. Firstly, we will propose more effective neural networks to increase the accuracy of feature embedding. Secondly, we will introduce more types of item auxiliary information to raise recommendation accuracy.

CRediT authorship contribution statement

Juan Ni: Methodology, Software, Investigation, Writing – original draft. **Zhenhua Huang:** Conceptualization, Supervision, Funding acquisition. **Yang Hu:** Resources, Data curation. **Chen Lin:** Validation, Writing - review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (No. 61772366) and the Natural Science Foundation of Shanghai (No. 17ZR1445900).

References

[1] S. Guo, Y. Wang, H. Yuan, et al, TAERT: Triple-attentional explainable recommendation with temporal convolutional network, Inf. Sci. 567 (2021) 185–200.
[2] R. Logesh, V. Subramaniyaswamy, D. Malathi, et al, Enhancing recommendation stability of collaborative filtering recommender system through bio-inspired clustering ensemble method, Neural Comput. Appl. 32 (2020) 2141–2164.
[3] J. Ni, Z. Huang, J. Cheng, S. Gao, An effective recommendation model based on deep representation learning, Inf. Sci. 542 (2021) 324–342.
[4] W. Yuan, H. Wang, X. Yu, et al, Attention-based context-aware sequential recommendation model, Inf. Sci. 510 (2020) 122–134.
[5] P. Bedi, S.K. Agarwal, V. Bhasin, in: ELM based imputation-boosted proactive recommender systems, IEEE, Jaipur, 2016, pp. 69–74.
[6] S. Seo, J. Huang, H. Yang, et al, in: Interpretable convolutional neural networks with dual local and global attention for review rating prediction, ACM, Como, 2017, pp. 297–305.
[7] C. Sundermann, J. Antunes, M. Domingues, et al, in: Exploration of word embedding model to improve context-aware recommender systems, IEEE, Santiago, 2018, pp. 383–388.
[8] Z. Yiru, B. Tassadit, W. Yewan, et al, A distance for evidential preferences with application to group decision making, Inf. Sci. 568 (2021) 113–132.
[9] X. He, L. Liao, H. Zhang, et al, in: Neural collaborative filtering, ACM, Perth, 2017, pp. 173–182.
[10] P. Covington, J. Adams, E. Sargin, in: Deep neural networks for youtube recommendations, ACM, Boston, 2016, pp. 191–198.
[11] Z. Huang, C. Yu, J. Ni, et al, An efficient hybrid recommendation model with deep neural networks, IEEE Access 7 (2019) 137900–137912.
[12] R. Yin, K. Li, J. Lu, Z.G. RsyGAN, in: Generative adversarial network for recommender systems, IEEE, Budapest, 2019, pp. 1–7.
[13] F. Zhao, M. Xiao, Y. Guo, in: Predictive Collaborative Filtering with Side Information, Elsevier, New York, 2016, pp. 2385–2391.
[14] D. Kim, C. Park, J. Oh, et al, in: Convolutional matrix factorization for document context-aware recommendation, ACM, Boston, 2016, pp. 233–240.
[15] J. Zhou, J. Wen, S. Li, W. Zhou, in: From content text encoding perspective: A hybrid deep matrix factorization approach for recommender system, IEEE, Budapest, 2019, pp. 1–8.
[16] C. Chen, M. Zhang, Y. Liu, S. Ma, in: Neural attentional rating regression with review-level explanations, ACM, Lodz, 2018, pp. 1583–1592.
[17] S. Xing, Q. Wang, X. Zhao, T. Li, A hierarchical attention model for rating prediction by leveraging user and product reviews, Neurocomputing 332 (2019) 417–427.
[18] Q. Zhang, J. Wang, H. Huang, et al, in: Hashtag recommendation for multimodal microblog using co-attention network, Elsevier, Melbourne, 2017, pp. 3420–3426.
[19] R. Ma, Q. Zhang, J. Wang, et al, in: Mention recommendation for multimodal microblog with cross-attention memory network, ACM, Michigan, 2018, pp. 195–204.
[20] Lin W, Li L, Li D. An item recommendation approach by fusing images based on neural networks. In: Proceedings of the 6th International Conference on Behavioral, Economic and Socio-Cultural Computing. Beijing: IEEE, 2019: 1-4.
[21] Q. Yang, G. Wu, Y. Li, et al, AMNN: Attention-based multimodal neural network model for hashtag recommendation, IEEE Trans. Comput. Social Syst. 7 (2020) 768–779.
[22] S. Oramas, O. Nieto, M. Sordo, et al, in: A deep multimodal approach for cold-start music recommendation, ACM, Como, 2017, pp. 32–37.
[23] Bougiatiotis K, Giannakopoulos T. Multimodal content representation and similarity ranking of movies. arXiv preprint arXiv:1702.04815, 2017.
[24] Y. Li, H. Wang, H. Liu, B. Chen, in: A study on content-based video recommendation, IEEE, Beijing, 2017, pp. 4581–4585.
[25] Y. Kumar, A. Sharma, A. Khaund, et al, in: IceBreaker: Solving cold start problem for video recommendation engines, IEEE, Taiwan, 2018, pp. 217–222.
[26] W. Xu, Y. Zhou, Course video recommendation with multimodal information in online learning platforms: A deep learning framework, British Journal of Educational Technology 51 (2020) 1734–1747.

[27] Z. Tao, Y. Wei, X. Wang, et al, MGAT: Multimodal graph attention network for recommendation, Inf. Process. Manage. 57 (2020) 102277.
[28] F. Zhang, N.J. Yuan, D. Lian, et al, in: Collaborative knowledge base embedding for recommender systems, ACM, San Francisco, 2016, pp. 353–362.
[29] R. Sun, X. Cao, Y. Zhao, et al, Multi-modal knowledge graphs for recommender systems, in: In: Proceedings of the 29th ACM International Conference on Information and Knowledge Management. Online: ACM, 2020, pp. 1–10.
[30] Z. Huang, X. Xu, J. Ni, et al, Multimodal representation learning for recommendation in Internet of Things, IEEE Internet Things J. 6 (2019) 10675–10685.
[31] R. Ying, R. He, K. Chen, et al, in: Graph convolutional neural networks for web-scale recommender systems, ACM, London, 2018, pp. 974–983.
[32] X. Wang, X. He, M. Wang, et al, in: Neural graph collaborative filtering, ACM, Paris, 2019, pp. 165–174.
[33] J.C. Bezdek, R. Ehrlich, W. Full, FCM: The fuzzy c-means clustering algorithm, Comput. Geosci. 10 (1984) 191–203.
[34] N.R. Pal, J.C. Bezdek, On cluster validity for the fuzzy c-means model, IEEE Trans. Fuzzy Syst. 3 (1995) 370–379.
[35] H. Sun, S. Wang, Q. Jiang, FCM-based model selection algorithms for determining the number of clusters, Pattern Recogn. 37 (2004) 2027–2037.
[36] X. Glorot, Y. Bengio, in: Understanding the difficulty of training deep feedforward neural networks, IEEE, Sardinia, 2010, pp. 249–256.
[37] H. Fanta, Z. Shao, L. Ma, SiTGRU: Single-tunnelled gated recurrent unit for abnormality detection, Inf. Sci. 524 (2020) 15–32.
[38] R. Sheikhpour, M.A. Sarram, S. Gharaghani, et al, A Robust graph-based semi-supervised sparse feature selection method, Inf. Sci. 531 (2020) 13–30.
[39] Z. Huang, X. Xu, H. Zhu, et al, An efficient group recommendation model with multiattention-based neural networks, IEEE Trans. Neural Networks Learn. Syst. 31 (2020) 4461–4474.
[40] C. Lin, R. Xie, X. Guan, L. Li, T. Li, Personalized news recommendation via implicit social experts, Inf. Sci. 254 (2014) 1–18.
[41] J. Ni, Z. Huang, C. Yu, et al, Comparative convolutional dynamic multi-attention recommendation model, IEEE Trans. Neural Networks Learn. Syst. (2021), https://doi.org/10.1109/TNNLS.2021.3053245.
[42] Huang T, She Q, Zhang J. BoostingBERT: Integrating multi-class boosting into BERT for NLP tasks. arXiv preprint arXiv:2009.05959, 2020.
[43] P. Dhankhar, ResNet-50 and VGG-16 for recognizing Facial Emotions, International Journal of Innovations in Engineering and Technology 13 (2019) 126–130.
[44] X. Yang, T. Zhang, C. Xu, Semantic feature mining for video event understanding, ACM Trans. Multimedia Comput. Commun. Appl. 12 (2016) 1–22.
[45] Q. Wang, Z. Mao, B. Wang, et al, Knowledge graph embedding: A survey of approaches and applications, IEEE Trans. Knowl. Data Eng. 29 (2017) 2724–2743.
[46] Z. Huang, X. Lin, H. Liu, et al, Deep representation learning for location-based recommendation, IEEE Trans. Comput. Social Syst. 7 (2020) 648–658.
[47] C. Lin, W. Chen, C. Qiu, et al, LibD3C: ensemble classifiers with a clustering and dynamic selection strategy, Neurocomputing 123 (2014) 424–435.
[48] H.A. Khorshidi, U. Aickelin, Constructing classifiers for imbalanced data using diversity optimisation, Inf. Sci. 565 (2021) 1–16.