

## RESEARCH ARTICLE

# Semantic Recommendation Model via Fusing Knowledge Graph and Formal Concept Analysis

LINA WEI<sup>1,2</sup> AND XINGTU ZHU<sup>3</sup><sup>1</sup>School of Computer Science, South China Normal University, Guangzhou 510631, China<sup>2</sup>School of Foreign Studies, Guangxi University for Nationalities, Nanning 530006, China<sup>3</sup>Center for Teacher Development of Tianhe District, Guangzhou 510650, China

Corresponding author: Lina Wei (2018010188@m.scnu.edu.cn)

**ABSTRACT** The core idea of semantic recommendation is to incorporate semantic knowledge into the recommendation process. The semantic recommendation algorithm, based on knowledge graph, ignores the deep implicit semantics of the evaluation data. The semantic recommendation algorithm based on the deep matrix decomposition model is limited to the implicit semantics of the evaluation data. The semantic recommendation algorithm based on the collaborative filtering algorithm performs only the selection of the nearest neighbors of the user or the item unilaterally and ignores the influence of other aspects, which naturally leads to a decrease in the recommendation accuracy. To solve the above problems, this paper introduces Formal Concept Analysis (FCA) based on collaborative filtering. Using the property that the formal concept in FCA can cluster objects (users) and attributes (items) simultaneously, we propose a semantic recommendation algorithm (SRKGFCA) based on the knowledge graph and formal concept analysis to solve the problem of ignoring user or item factors. Finally, the proposed semantic recommendation algorithm is validated on two public datasets in this work. By using traditional algorithms and current semantic recommendation algorithms as benchmarks, extensive experiments show that our proposed algorithm consistently outperforms state-of-the-art methods.


**INDEX TERMS** Semantic recommendation, knowledge graphs, representation learning, formal concept analysis, deep matrix decomposition.

## I. INTRODUCTION

In recent years, researchers have introduced semantic information into traditional recommendation algorithms to alleviate problems in various types of algorithms. The most classical application is the personalized recommendation for Amazon online shopping mall [1]. Although semantic recommendation algorithms have facilitated the development of recommender systems [2], the main drawback is the cold-start and data sparsity problems. Zhang et al [3] achieved high quality movie recommendation by introducing movie synopsis, poster and ontology as movie knowledge embedded in a low-dimensional vector space, which largely alleviated the cold-start and data sparsity problems. Semantic recommendation algorithms recommend items based on user preferences,

but it is difficult to perform feature extraction when the items are multimedia data.

Deep learning-based recommendation algorithms [4] can extract semantic information about users or items from a large amount of text data, but make recommendation systems overly dependent on external text data and face the problem of data source and reliability. The emergence of knowledge graph that contain rich semantic information has brought semantic recommendation research to a new stage of development. The emergence of a large number of public knowledge graphs (e.g., DBpedia [5], YAGO [6], Wikidata [7], WordNet [8], and CN-DBpedia [9], etc.) has enabled researchers to more easily access and utilize semantic information and obtain the final recommendation list by merging semantic top-k nearest neighbors and collaborative top-k nearest neighbors to form the final recommendation list and improve recommendation accuracy. Translation model-based (Trans)

The associate editor coordinating the review of this manuscript and approving it for publication was Chong Leong Gan .

map embedding technology is the most widely used, including TransE [10], TransH [11], TransR [12] and TransD [13]. They embed knowledge graph triples into low-dimensional vector space through gradient update training, which is more interpretable and simple than the method based on graph neural network [14]. Wu et al. [15] learned the representation of movies by constructing a knowledge graph in the movie domain, and then combined the semantic similarity of movies and the similarity of elements of collaborative filtering to achieve higher accuracy and coverage of recommendations. Reference [16] the knowledge graph representation learning method is used to embed the semantic information of the items into a low-dimensional semantic space; then, the semantic similarity between the recommended items is calculated, and then, this item semantic information is fused into the collaborative filtering recommendation algorithm. This algorithm increases the performance of recommendation at the semantic level. Although the above knowledge graph-based recommendation methods reflect user preferences through rating data, the high sparsity of the rating matrix is the major drawback of the above algorithms. The key is to unlock the hidden semantics of the rating matrix.

For the past few years, with the development of deep learning, He et al. [17] proposed NeuMF model (Neural Matrix Factorization), which combines probabilistic computation and multi-layer perceptron. Xue et al. [18] proposed DMF (Deep Matrix Factorization) model to address the shortcomings of NeuMF in processing user ratings and obtained better results. Although matrix factorization can effectively determine the latent semantics in user-item interaction data, it ignores the external semantic knowledge. Because it is limited to the cryptic meaning mining of the items scored by users, and deep learning itself applies the idea of heuristic algorithm, when there are too many missing values in the scoring matrix, the deep matrix factorization model is easy to fall into the dilemma of local optimum in the training. Therefore, the input of matrix factorization model is processed accordingly, so that the input matrix or vector can describe the characteristics of users and items more accurately, which can often achieve better results. Jiarong et al. [19] performed DMF model training after scoring matrix completion in the way of mathematical matrix completion, and achieved good recommendation accuracy. Quan et al. [20] achieved better results by performing graph embedding through a modified TransHR model and then populating the scoring matrix with semantic similarity mixed with collaborative filtering similarity. However, Jiarong et al. [19] simply used mathematical formulas to complete the scoring matrix, while ignoring external semantic information. Yuan et al. [20] used only a superficial model for hidden semantic mining and ignored the deeper semantic information of the evaluation matrix. In recent years, researchers have found that FCA can cluster both objects (users) and attributes (items) and is able to find the nearest neighbors of users or items more efficiently than the k-nearest neighbor algorithm. Both user-based K-nearest

neighbor and item-based K-nearest neighbor consider only one-sided neighbor relationship. By introducing FCA into collaborative filtering recommendation and considering the neighbor relationship between users and items at the same time, they achieved good recommendation effects [21], [22]. Jiang [23] proposed an ontology semantic extension-based FCA to improve the computation of formal concept similarity by incorporating semantic information from knowledge graph. In FCA-based recommendation, Boucher-Ryan et al. [24] first introduced the concept lattice of FCA into the collaborative filtering recommendation algorithm to find the nearest neighbor users based on the concept lattice, so as to make product recommendation Zou et al. [21] used a concept lattice in cosine similarity computation to improve the prediction accuracy of a recommendation system with collaborative filtering. Mezni et al. [22] proposed a cloud server recommendation system based on fuzzy FCA to determine the relationship between users and items at a deeper level, alleviating the problems of data sparsity and cold start and achieving better results. The approach described above can only make a recommendation for small amounts of data, since a concept grid must be constructed. At present, the recommendation algorithm based on FCA is far from popular recommendation algorithms based on matrix factorization and knowledge graph in terms of recommendation accuracy [21], [22], [24]. However, how to effectively use the related theories of FCA and its semantic extension and integrate them into the existing recommendation algorithms to improve the recommendation effect of the existing algorithms is a problem worth studying.

In summary, the recommendation model based on knowledge graph ignores the deeper implicit semantics of the evaluation data, while the recommendation model based on the deep matrix decomposition is limited to the implicit semantics of the evaluation data FCA can effectively solve the problem of selecting the nearest neighbors of users or items by traditional recommendation algorithms of collaborative filtering, and has the theoretical basis for combining with related semantic techniques. However, there are also problems in creating concept lattices in large-scale datasets or the accuracy of heuristic concept algorithms is not high. In this paper, we combine knowledge graph and FCA to propose a new and better semantic recommendation algorithm.

- 1) In collaborative filtering, combined with the depth matrix decomposition model, a semantic recommendation algorithm based on knowledge graph is proposed in this paper. (Semantics based Recommendation using Knowledge Graph, SRKG)
- 2) A fuzzy FCA-based concept area calculation method and an attribute (item) based heuristic concept construction algorithm are proposed to solve the problems of lack of consideration of user ratings and difficulties in operating with large-scale datasets.
- 3) FCA is integrated into the SRKG algorithm. A semantics-based recommendation algorithm using

knowledge graph and formal concept analysis (Semantics based Recommendation using Knowledge Graph and Formal concept analysis, SRKGFCA) is proposed to solve the SRKG in PR1 evaluation prediction, which considers only the item factor and ignores the user factor.

The remainder of this paper is organized as follows. In Section II, we briefly introduce the related work. Section III presents semantic recommendation algorithms based on knowledge graph. Section IV introduces our SRKGFCA algorithm in detail. Section V evaluates our implementation based on experimental results. Section VI concludes this paper with future work.

II. RELATED RESEARCH

A. KNOWLEDGE GRAPH

Knowledge Graph(KG) is a practical approach to model factual information in the form of entity-to-entity relationships for representing extensive information from different domains. DBpedia Knowledge Graph [5] is a specific example of a Semantic Web application that automatically extracts semi-structured data from Wikipedia and transforms it into structured data that can be linked via Linked Data to other knowledge graphs such as Wikidata [7], YAGO [6], Freebase, etc. DBpedia’s Semantic Web technology facilitates the application of Wikipedia data and was awarded Best Application Service at the 2009 Semantic Web Awards. The atlas contains more than 6 million entities, and in particular all the entities of the Movielen movie recommendation dataset required for use in this work are included.

B. FORMAL CONCEPT ANALYSIS

Formal Concept Analysis (FCA) [25] is a theoretical framework that provides the basis for data analysis and knowledge processing and can represent the relationships between objects and attributes in a given domain. In real life, many relationships between objects and attributes are ambiguous, for example, even if someone has seen a certain movie, there are often a variety of levels of personal preferences. For example, not bad, average, not so good, etc. Fuzzy information like this cannot be defined by a simple 0 or 1. Therefore, fuzzy set theory was introduced to define Fuzzy Formal Concept Analysis (FFCA) [25].

Definition 1 (Fuzzy Formal Background [25]): A fuzzy formal background is a quadruple  $K = (O, A, V, R = \varphi(O \times A))$ , where  $O$  is the set of objects,  $A$  is the set of attributes,  $R$  denotes the fuzzy relation between the set of objects  $O$  and the set of attributes  $A$ , and  $V = [0, 1]$  denotes the affiliation taken in the fuzzy relation. Each object and attribute in FFCA has an affiliation  $\mu_1(o, a) \in V = [0, 1]$  between them. Given a grid of concept pairs  $(o, a) \in O \times A$  and the affiliation  $\mu_1$ , that is, an indication of the extent to which the object  $o$  has  $(\mu_1)$  the attribute  $a$ . Corresponding to the user rating relationship for the movie, how much  $(\mu_1)$  the user  $o$  likes the movie  $a$ , or how much the user  $o$  rates the movie  $a$  as  $\mu_1$ . The fuzzy formal background is shown in Table 1.

TABLE 1. User-film fuzzy formal background.

user	item				
	a	b	c	d	e
1	0.8	0.8	0	0.4	0.8
2	1	0.4	0.6	1	0

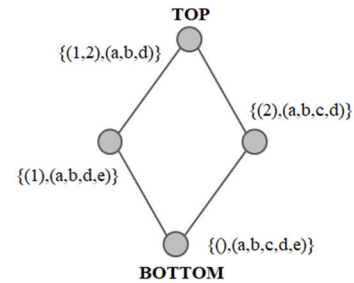


FIGURE 1. User-Film Fuzzy Concept Grid.

Definition 2 (Fuzzy Formal Concept [16]): Given a fuzzy background  $K = (O, A, V, R = \varphi(O \times A))$ , let  $E \subseteq O, I \subseteq A$ , and given an affiliation threshold  $\alpha$ , define the following operations.

$$E^* = \{a \in A \mid \forall o \in E, \mu_1(o, a) \geq \alpha\} \quad (2-1)$$

$$I^* = \{o \in O \mid \forall a \in I, \mu_1(o, a) \geq \alpha\} \quad (2-2)$$

where  $\mu_1(o, a)$  is the affiliation degree of  $o$  corresponding to  $a$ , and when the binary  $C = (\varphi(E), I)$  satisfies  $E^* = I, I^* = E$ , then  $C$  is said to be a fuzzy formal concept, where  $E$  is called the fuzzy concept externality and  $I$  is called the fuzzy concept implication.

Also in the fuzzy concept  $C$ , the affiliation  $\mu_o$  of each object  $o \in O$  for the fuzzy set  $\varphi(E)$  is computed publicly as

$$\mu_o = \min_{a \in I} \mu_1(o, a). \quad (2-3)$$

Taking the fuzzy formal background of Table 1 as an example, let the affiliation threshold  $\alpha$  be 0.5. It is necessary to state that since the only rating of user 2 for movie b is  $0.4 < \alpha$ , which indicates that this user 2 does not like movie b, this correspondence will not be considered when generating fuzzy concepts. One of the fuzzy concepts can be obtained from the above definition is  $\{(2/0.6), (a, c, d)\}$ , where ‘2/0.6’ indicates that user 2 has an affiliation of 0.6 for the movie set  $(a, c, d)$ .

In FFCA, the concept lattice is defined similarly to single-valued FCA, where two fuzzy concepts are given and then the corresponding fuzzy concept lattice can be obtained by connecting the concepts through the partial order relation ‘ $\leq$ ’ between them. Figure 1 is the fuzzy concept lattice corresponding to Table 1.

III. SEMANTIC RECOMMENDATION ALGORITHMS BASED ON KNOWLEDGE GRAPH

A. SIMILARITY CALCULATION METHOD BASED ON KNOWLEDGE GRAPH

For semantic recommendation algorithms, similarity computation is the core work, including semantic similarity of items

based on knowledge graph, and similarity of items based on ratings.

1) ITEM SIMILARITY BASED ON SEMANTICS OF KNOWLEDGE GRAPH

In the semantic recommendation algorithm, the items  $I_i$  are represented by vectors as

$$I_i = (e_{i,1}, e_{i,2}, \dots, e_{i,d})^T \quad (3-1)$$

where  $d$  is denoted as the embedding of items into a vector space of dimension  $d$  and  $e_{i,q}$  is the value corresponding to the  $q$ th dimensional vector.

The Euclidean distance formula is used to calculate the entity similarity of the recommended items.

$$dis(I_i, I_j) = \sqrt{\sum_{k=1}^d (e_{i,k} - e_{j,k})^2} \quad (3-2)$$

The Euclidean distance formula leads to a distance-based semantic similarity formula for the two items  $sim_{graph}$ ,

$$sim_{graph}(I_i, I_j) = \frac{1}{1 + dis(I_i, I_j)} \quad (3-3)$$

2) SCORING-BASED ITEM SIMILARITY

We use the user-item interaction data, i.e., the user-item rating data, to construct a rating matrix  $R_{m \times n}$  of user set  $U$  to item set  $I$

$$R_{m \times n} = \begin{pmatrix} r_{1,1} & \dots & r_{1,n} \\ \vdots & & \vdots \\ r_{m,1} & \dots & r_{m,n} \end{pmatrix} \quad (3-4)$$

The user set's rating of an item is used as a rating vector, e.g. the vector for item is denoted as  $I_i = \{r_{1i}, r_{2i}, \dots, r_{mi}\}^T$ . Then the similarity  $sim_{cos}$  of two items  $I_i$  and  $I_j$  is denoted as,

$$sim_{cos}(I_i, I_j) = \frac{\sum_{u=1}^m (r_{u,i} \times r_{u,j})}{\sqrt{\sum_{u=1}^m r_{u,i}^2} \times \sqrt{\sum_{u=1}^m r_{u,j}^2}} \quad (3-5)$$

where can be known from the vector  $I_i$  and  $r_{u,i}$  is denoted as user  $u$ 's rating of item  $i$ .

The cosine similarity takes values in the range of  $[-1,1]$ , so normalization is performed. The normalized cosine similarity formula for item similarity based on ratings  $sim_{rate}$  is.

$$\begin{aligned} sim_{rate}(I_i, I_j) &= 0.5 + 0.5 \frac{\sum_{u \in U_{i \cap j}} (r_{u,i} - \bar{r}_u)(r_{u,j} - \bar{r}_u)}{\sqrt{\sum_{u \in U_i} (r_{u,i} - \bar{r}_u)^2} \times \sqrt{\sum_{u \in U_j} (r_{u,j} - \bar{r}_u)^2}} \quad (3-6) \end{aligned}$$

where,  $\bar{r}_u$  denotes the average rating of user  $u$ , and  $U_{i \cap j}$  denotes the set of users who rated both items  $i$  and  $j$ .

B. SEMANTIC RECOMMENDATION ALGORITHMS BASED ON KNOWLEDGE GRAPH

In this paper, we propose a semantic recommendation algorithm based on knowledge graph for similarity (SRKG), which mainly includes the fusion of two similarities and makes initial score prediction, then improves and trains the DMF model by including semantic information in the predicted scores, and finally results in the recommendation.

1) SIMILARITY FUSION AND SCORE PREDICTION

The semantic information of the knowledge graph and the rating information of each user are incorporated into the algorithm, i.e., the two item similarities mentioned above are fused. According to (3-3) and (3-6), we can obtain the fusion formula  $sim$  for the similarity of the two items.

$$sim(I_i, I_j) = \begin{cases} sim_{rate}(I_i, I_j), & \text{iff } sim_{graph}(I_i, I_j) = 0 \\ sim_{graph}(I_i, I_j), & \text{iff } sim_{rate}(I_i, I_j) = 0 \\ a \cdot sim_{rate}(I_i, I_j) + (1 - a) \cdot sim_{graph}(I_i, I_j) \end{cases} \quad (3-7)$$

where,  $a < 1$  is the fusion factor and when  $a = 0$ , the proposed SRKG algorithm considers only the item similarity calculated based on the knowledge graph.

We segmented the function, mainly to alleviate the cold start ( $sim_{rate} = 0$ ) or cases where the knowledge graph cannot be the similarity (e.g., problems such as movie entities not existing,  $sim_{graph} = 0$ ). By introducing the item similarity computed by the knowledge graph, we can more accurately predict the user's rating of unrated items. The predicted rating PR1 of user  $u$  for item  $i$  is,

$$PR1(u, i) = \frac{\sum_{j \in (N(u) \cap S(i,k))} (sim(I_i, I_j) \times (r_{u,j} - \bar{r}_u))}{\sum_{j \in (N(u) \cap S(i,k))} sim(I_i, I_j)} + \bar{r}_u \quad (3-8)$$

where  $N(u)$  denotes the items rated by user  $u$ ,  $S(i, k)$  denotes the top  $k$  items with the greatest similarity to item  $i$ , and the intersection of the two is the reference object for rating prediction.

2) DMF MODEL TRAINING

The purpose of the DMF depth matrix decomposition model is actually to represent item and user features with low-dimensional vectors, while accurately measuring the similarity of users and items. For this purpose, its objective function is as follows

$$p_i = f_{\theta_n}^i(\dots W_{i3} f_{\theta_3}^i(W_{i2} f_{\theta_2}^i(W_{i1} Y_{i*})) \dots) \quad (3-9)$$

$$q_j = f_{\theta_n}^j(\dots W_{j3} f_{\theta_3}^j(W_{j2} f_{\theta_2}^j(W_{j1} Y_{*j}^T)) \dots) \quad (3-10)$$

$$\hat{Y}_{ij} = F_{DMF}(u_i, v_j | \theta) = sim_{cos}(p_i, q_j), u_i \in Y_{i*}, v_j \in Y_{*j}. \quad (3-11)$$



**Algorithm 1** DMF model training algorithm Train\_DMF

Input: number of model *iterations iter*, negative sample rate *nratio*, original scoring matrix *R*, predicted scoring matrix *R<sub>PR1</sub>*

Output: user and item weight matrix *W<sub>u<sub>i</sub></sub>, W<sub>v<sub>i</sub></sub>*, where (*i* = 1, . . . , *N* - 1), the similarity of users and items  $\hat{Y}_{ij}$

Initialization :

Randomly generate *W<sub>u<sub>i</sub></sub>, W<sub>v<sub>i</sub></sub>*; // where *W<sub>u<sub>i</sub></sub>, W<sub>v<sub>i</sub></sub>* is the weight matrix

*Y* = the values in the matrix *R*, *Y<sup>+</sup>* = the non-zero values in *Y*, *Y<sup>-</sup>* = all zero values in *Y*, *Y<sub>sampled</sub><sup>-</sup>* = randomly select *nratio*\*|*Y*| values from *Y<sup>-</sup>*; // where *Y<sup>+</sup>* is the positive sample, and *Y<sub>sampled</sub><sup>-</sup>* is the negative sample

*T* = *Y<sup>+</sup>* ∪ *Y<sub>sampled</sub><sup>-</sup>*; // *T* is the training set

**for** *it* from 1 to *iter* **do**

**for** each user *i* and item *j* in *T* **do**

*Y<sub>i\*</sub>, Y<sub>\*j</sub>* ← *R*+*R<sub>PR1</sub>*; // *Y<sub>i\*</sub>, Y<sub>\*j</sub>* are the row and column vectors representing user and item characteristics in *R*+*R<sub>PR1</sub>*

*p<sub>i</sub>, q<sub>j</sub>* ← Input *Y<sub>i\*</sub>, Y<sub>\*j</sub>* to (3-9), (3-10); // *p<sub>i</sub>, q<sub>j</sub>* denotes the low-dimensional vector obtained after training

$\hat{Y}_{ij}$  ← input *p<sub>i</sub>, q<sub>j</sub>* to (3-11); //  $\hat{Y}_{ij}$  cosine similarity, i.e. normalized prediction score

*L* ← Enter  $\hat{Y}_{ij}$  and *T* to (3-12); // Loss calculation

        Reverse learning update *W<sub>u<sub>i</sub></sub>, W<sub>v<sub>i</sub></sub>*; // Gradient update model parameters

**end for**

**end for**

**return** *W<sub>u<sub>i</sub></sub>, W<sub>v<sub>i</sub></sub>,  $\hat{Y}_{ij}$* ;

Then, we measure the loss of training by the cosine similarity, and the specific loss function is shown below

$$L = - \sum_{(i,j) \in Y^+ \cup Y^-} \left( \frac{Y_{ij}}{\max(\text{rate})} \log \hat{Y}_{ij} + \left( 1 - \frac{Y_{ij}}{\max(\text{rate})} \right) \log(1 - \hat{Y}_{ij}) \right) \quad (3-12)$$

where, *Y<sup>+</sup>*, *Y<sup>-</sup>* denote the non-zero and zero values in the rating matrix, respectively; *max(rate)* denotes the highest rating of the recommendation system, which is mainly used to normalize the ratings; *Y<sub>ij</sub>* and  $\hat{Y}_{ij}$  denote the true and predicted values of the ratings, respectively.

where both *R* and *R<sub>PR1</sub>* are *m*\**n* matrices, and the missing values of the *R* matrix are filled with 0, the *R<sub>PR1</sub>* matrix in Algorithm 1 represents the prediction rating matrix, which is derived from the above computed *PR1* prediction ratings, and is filled with 0 values for items that have been selected by the user. Compared to the original DMF training approach [18], the improvement in this paper is mainly reflected in the 8th

row of the Algorithm 1, which is populated into the input vectors *p<sub>i</sub>, q<sub>j</sub>* by the predicted ratings *PR1* containing the external semantics. It is worth noting that the positive and negative samples of the model, as well as the true ratings *Y<sub>ij</sub>* required by the loss function, are still derived from the original rating matrix *R* in order to guarantee the accuracy of the model. We mainly populate the input vectors *p<sub>i</sub>, q<sub>j</sub>* with the main purpose of making the input vectors *p<sub>i</sub>, q<sub>j</sub>* more accurate in characterizing the user and the item. Because the training positive and negative samples of the original model are not changed, the training method of the DMF model proposed in this paper is not much different in time complexity compared with the training method of the original model.

After the DMF model is trained, we can get the *u* and *i* cosine similarity by inputting the target user *u* and the target item *i* into the model, i.e., the normalized prediction score *sim<sub>DMF</sub>*,

$$\text{sim}_{DMF}(u, i) = DMF(u, i). \quad (3-13)$$

Since the normalized cross-entropy loss function of the model is normalized considering the ratings as well as the bias term, the predicted rating *PR2* of the target user *u* for item *i* is

$$PR2(u, i) = \max(\text{rate}) * \text{sim}_{DMF}(u, i) \quad (3-14)$$

where *max(rate)* indicates the maximum value of the rating in the recommender system, e.g., if the highest user rating in MovieLens is 5, then *max(rate)* = 5.

Rating predictions are made for each target user, and by ranking the predicted ratings, the SRKG algorithm can be summarized as the following process,

- 1) Retrieve the corresponding entity and its triples from DBpedia by the name of the item.
- 2) The triad is trained to learn the TransE mapping representation, and is given a low-dimensional vector containing items.
- 3) Knowledge graph-based similarity of items and evaluation-based similarity of items are computed from the low-dimensional vector of items and the evaluation information, respectively, and the similarity fusion is performed.
- 4) The fused item similarity of the item is used to calculate the prediction score *PR1* using *k*-nearest neighbors;
- 5) Fill the predicted scores *PR1* into the input vector *p<sub>i</sub>, q<sub>j</sub>* and train the DMF depth matrix decomposition model.
- 6) After the model is trained, the prediction results *PR2* and recommendation results are obtained by inputting the target users and items into the model.

#### IV. SEMANTIC RECOMMENDATION ALGORITHMS FOR FORMAL CONCEPT ANALYSIS

##### A. STRONG CONCEPT CALCULATION METHODS

Following the FCA of Yuncheng Jiang et al. in semantic retrieval [26] and the semantized FCA [23], for fuzzy formal concepts *C*( $\varphi(E), I$ ), the main purpose is to determine

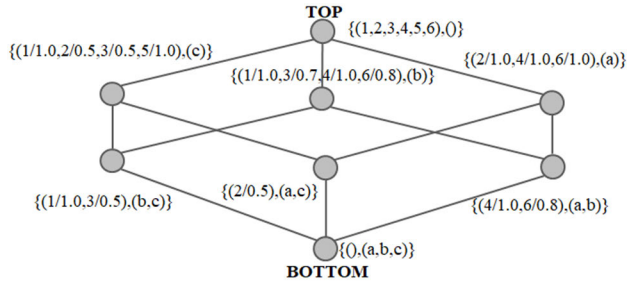


FIGURE 2. Concept Grid.

the closest set of items, and we will also include semantic information in the computation of the fuzzy concept area calculation. For item  $i$ , the formula for the fuzzy concept area formula as follows

$$S'_{concept}(\varphi(E), I, i) = \sum_{i' \in I} sim(i, i') \times \sum_{\mu_i \in \varphi(E)} \mu_i \quad (4-1)$$

where  $\mu_i$  denotes the affiliation of each object in the extents  $\varphi(E)$ , which follows from (2-3), and  $sim(i, i')$  denotes the similarity of the two items, which follows from (3-7).

To find the set of nearest neighbors of an object or item, we need to subtract concepts that contain only one item or object from the concept lattice. For example, in the concept lattice of Figure 2, there are three fuzzy concepts that contain item  $b$ . For example, to find the nearest neighbors of item  $b$ , we first need to subtract the concept  $C_1 = \{(1/1.0, 3/0.7, 4/1.0, 6/0.8), (b)\}$ . Among the remaining 2 concepts, for the sake of example, let the similarity between attributes be both 1. Concept  $C_2 = \{(4/1.0, 6/0.8), (a, b)\}$  is a strong concept of item  $b$  because it has the largest concept area  $S' = (1 + 0.8) * (1 + 1) = 3.6$ , so concept  $C_2$  is a strong concept of item  $b$ .

**B. STRONG CONCEPT-BASED SCORING PREDICTION METHODS**

By the above work we can calculate a strong concept for each item. The implication  $I$  of a strong concept is the set of nearest items of the item. For example, if the concept  $C_2$  is a strong concept of item  $b$ , then the implication  $(a, b)$  is the set of closest users of item  $b$ , and  $a$  is the closest user of item  $b$ .

With the set of nearest neighbor users, we can improve (3-8), so that the improved  $PR1_{FCA}$  is

$$PR1_{FCA}(u, i) = \frac{\sum_{j \in (N(u) \cap (I(C_i) \cup S(i, k)))} (sim(I_i, I_j) \times (r_{u, j} - \bar{r}_u))}{\sum_{j \in (N(u) \cap (I(C_i) \cup S(i, k)))} sim(I_i, I_j) + \bar{r}_u} \quad (4-2)$$

where  $I(C_i)$  denotes the set of implication of the formal concept  $C_i$ .

Compared to (3-8), this equation differs mainly in the selection of nearest neighbors by replacing  $S(k, i)$  with  $I(C_i) \cup S(k, i)$ , which solves the problem that only item factors are considered in the SRKG semantic recommendation

**Algorithm 2** Heuristic Concept Construction Algorithm HCC

Input: Attribute  $a$ , fuzzy formal background  $K'_\alpha$  ;

Output: Strong concept  $C(E, I)$  of the attribute  $a$  ;

Initialization.

$E = \emptyset, I = \emptyset, ME = 0$  ; // Initialize extents and connotations, maximum conceptual area

**while** (true) :

$e = \emptyset, me = 0$  ; //initialize this epoch and concept area

**for** (each  $o \in (a^* - E)$ ) **do**

// iterate through all ephemeral objects of  $a$  that are not in  $E$  and add them to the ephemeral in turn

**if**  $S'_{concept}(E \cup o, (E \cup o)^*, a)$  is max

**then** //find the epitope with the largest conceptual area  $e = E \cup o, me = S'_{concept}(E \cup o, (E \cup o)^*, a)$  ; //save this epoch and area

**end if**

**end for**

**if**  $e > ME$  or  $(|I| + 1) < 2$  **then**

// Determine if the current maximum area is larger than the global maximum area, and the number of inner attributes cannot be less than 2

$E = e, ME = me, I = e^*$  ; // stored as

global extents and intents, and maximum concept area

**else then**

**Break;**

**end if**

**end while**

**return**( $E, I$ );

algorithm. Of course, we will also perform an experimental comparative analysis of the effectiveness of the substitution in the experiment.

**C. HEURISTIC CONCEPT CONSTRUCTION ALGORITHM**

Integrating FCA into semantic recommendation solves the problem that the SRKG algorithm lacks consideration of user factors. However, to implement FCA-based rating prediction, we need to construct concept lattices whose construction requires extremely high time complexity. We propose a heuristic concept construction algorithm to heuristically construct strong concepts for each item.

For a fuzzy form background  $K'_\alpha = (O, A, V, R = \varphi(O \times A))$  processed by affiliation thresholding, the Algorithm 2 to heuristically find the strong concept of any attribute (item)  $a \in A$  is as follows:

Taking the processed fuzzy background and item  $b$  in Table 2 as an example, we describe an example of the operation of the Algorithm 2. We first obtain the extents  $b^* = \{1, 3, 4, 6\}$  of  $b$ , where  $b$  does the  $*$  operation when the affiliation of user 5 is filtered because it is lower than the affiliation threshold ( $a \geq 0.5$ ), and also for the convenience of demonstration, we set the inter-attribute similarity in the inner set to 1. After combining the objects in the extents with  $b$  in

**TABLE 2.** Fuzzy formal background processed by threshold.

user	item		
	a	b	c
1	0	1.0	1.0
2	1.0	0	0.5
3	0	0.7	0.5
4	1.0	1.0	0
5	0	0	1.0
6	1.0	0.8	0

turn, we can get the users 1 and 4 that make the largest concept area, and the alternative area 2; randomly select user 1 or 4 as the alternative user, and continue adding the remaining objects to the epitaxy. If user 1 is chosen as the alternative user, we end up with a strong concept  $\{(1/1.0, 3/0.5), (b, c)\}$  with an area of 3. If user 4 is chosen, we get the above concept  $C_2 = \{(4/1.0, 6/0.8), (a, b)\}$  with an area of 3.6. Obviously the heuristic algorithm does not always give the optimal result, but it ensures that the product of the number of extents and connotations is the largest.

Finally, we apply the HCC algorithm to the FCA-based rating prediction to replace the concept lattice generation and thus reduce the complexity of the algorithm.

#### D. SRKG FCA SEMANTIC RECOMMENDATION ALGORITHM

We achieve the purpose of dynamically setting the number of nearest neighbors through the cluster property of formal concepts by using the strong concept connotation of items as the nearest neighbors of items. For the semantic recommendation algorithm, the number of nearest neighbors for each item is obviously not fixed, so dynamically setting the number of nearest neighbors by FCA can improve the scoring prediction efficiency to some extent. As for the accuracy, we will also conduct comparative experiments in the experiment. To solve the problem that the concept lattice is difficult to generate in large-scale datasets, we propose a heuristic method to construct concepts and a method to measure the range of fuzzy concepts, which can reduce the complexity of the algorithm to a great extent. By incorporating the heuristic concept construction algorithm into SRKG FCA, our algorithm can be executed in large-scale datasets. The SRKG FCA algorithm can be summarized as the following process.

- 1) preprocessing and correlation as in SRKG algorithm and computation of two similarities followed by similarity fusion.
- 2) construction of a fuzzy background using the scoring matrix  $R$ .
- 3) Generation of a strong fuzzy concept for each item using the heuristic concept construction algorithm HCC.
- 4) The set of inclusions by the strong concept of an item as the nearest neighbour set for that item and  $PR1_{FCA}$  score prediction.

**TABLE 3.** Datasets.

Dataset	Number of users	Number of items	Number of ratings
ML100K	944	1638	100,000.
ML1M	6041	3953	1,000,000.

**TABLE 4.** Triplet information table used for knowledge graph embedding.

Dataset	Number of entities	Number of relations	Number of triples
ML100K	111729	27	394100
ML1M	297112	29	1037098

- 5) replacing  $PR1_{FCA}$  in SRKG for the same input vector filling operation, DMF model training and finally the prediction score  $PR2$  and recommendation results.

Obviously, the problem of ignoring the user factor in the SRKG algorithm is solved by FCA, using heuristic ideas to generate strong concepts that allow the algorithm to run in large-scale datasets. The improved SRKG FCA algorithm, whose recommendation accuracy changes, is demonstrated in the experiment.

## V. EXPERIMENT AND RESULT ANALYSIS

### A. EXPERIMENTAL DATA

In our experiments, we use two movie standard datasets in Table 3 (MovieLens 100K (ML100K) and MovieLens 1M (ML1M)) to test the evaluation prediction error and recommendation effect of the proposed algorithm. ML100K is a dataset with relatively small data volume and ML1M is a dataset with relatively large data volume.

In this paper, DBpedia is selected as the knowledge graph of the experiment, and the version number of DBpedia is 2019-8-30. We used DBpedia Spotlight [27] to map the movie name data contained in the ML100K and ML1M datasets into DBpedia, respectively. Although all movie name information has been integrated into DBpedia knowledge graph [5], there are cases where movie names cannot match the knowledge graph. The main reasons are as follows, 1)The movie noun in ML100K and ML1M data sets is missing, and the movie marked as 'unknown'. 2)The movie name exists in the special characters, such as ' $\frac{1}{3}$ '. 3)Movies are released in different years. Finally, for ML100K we get 1621 movie entities corresponding to DBpedia and for ML1M we get 3892 movie entities corresponding to DBpedia. After obtaining the corresponding movie entities in DBpedia, we extract and filter triples (head entity, relation, tail entity). The triples corresponding to 1621 and 3892 movie entities in ML100K and ML1M data sets were obtained. All the relations which appear less than 10 times in the triplet set have been cleaned out. Finally, 'dbo:wikiPageWikiLink' and 'dbo:starring' to be some 20 semantic relations of knowledge graph which indicate the relations among categories, actors and actors. The triplet related information that we finally use for knowledge graph representation learning is shown in Table 4.

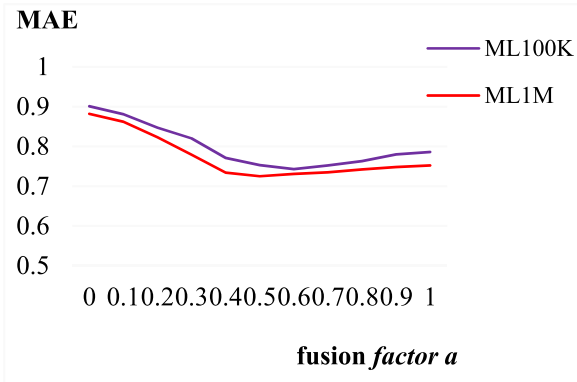


FIGURE 3. Variation of MAE for PR1 prediction scores with different fusion factors  $\alpha$ .

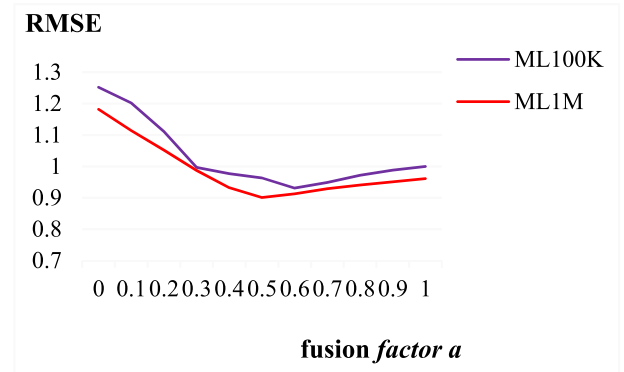


FIGURE 4. Variation of RMSE for PR1 prediction scores with different fusion factors  $\alpha$ .

B. BENCHMARK AND EVALUATION CRITERIA

We used the evaluation method used in the literature [17], [28], [29], with the following formula.

$$HR = \frac{1}{n} \sum_{i=1}^N hit(i) \tag{5-1}$$

$$NDCG = \frac{1}{n} \sum_{i=1}^n \frac{1}{\log_2(p_i + 1)} \tag{5-2}$$

$$MAE = \frac{1}{n} \sum_{i=1}^n |\hat{y}_i - y_i| \tag{5-3}$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2} \tag{5-4}$$

where,  $hit(i)$  of (5-1) indicates whether the target item appears in the Top- $k$  recommendation list, 1 if it exists and 0 if it does not exist,  $p_i$  of (5-2) indicates the position of the target item appearing in the recommendation list, and  $p_i \rightarrow \infty$  if the target item does not exist in the list; both take values in the range [0,1].  $y, k$  in (5-3) and (5-4) denote the predicted and true scores, respectively, and  $n$  denotes the number of samples in the test set.

C. EXPERIMENTAL SETUP AND ANALYSIS OF RESULTS

1) SEMANTIC VALIDATION EXPERIMENTS FOR FUSED KNOWLEDGE GRAPHS

We fuse the rating-based item similarity with the semantic similarity of the knowledge graph by a fusion factor  $\alpha$ , which increases from 0 to 1 in steps of 0.1, i.e., from using only the semantics of the knowledge graph for recommendations to using only collaborative filtering for recommendations. The graph embedding dimension is 200 and default values are used for all other parameters. We run the SRKG algorithm on the ML100K and ML1M datasets (with a nearest neighbor  $k$ -value of 20) and obtain the experimental results in Figure 3 and Figure 4.

As seen in Figure 3 and Figure 4, both MAE and RMSE error values decrease as the fusion factor  $\alpha$  increases. For

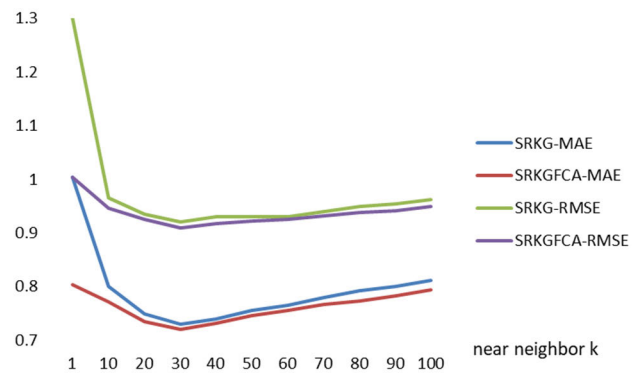


FIGURE 5. MAE and RMSE variation of PR1 and PR1<sub>FCA</sub> in ML100K dataset.

ML100K, the lowest error value is reached at  $\alpha = 0.6$ , while for ML1M, the lowest error value is reached at  $\alpha = 0.5$ . The reason is that ML1M has a larger amount of evaluation data to measure the similarity of items based on evaluations only, and has a higher accuracy than ML100K, so it does not need more knowledge graph similarity to be optimized relatively. It is worth noting that the two errors are actually relatively larger when  $\alpha = 0$  or 1. This shows the effectiveness of merging two similarities. In the following experiments, we choose 0.5 as the fusion factor for ML1M and 0.6 as the fusion factor for ML100K.

2) INTEGRATED FCA VALIDATION EXPERIMENTS

To calculate MAE and RMSE, we use two prediction scores, PR1 and PR1<sub>FCA</sub>. To find the best value for nearest neighbor  $k$ , for the ML100K dataset,  $k$  is set from 1 to 100 with a step size of 10. For the ML1M dataset,  $k$  is set from 1 to 200 with a step size of 20. Finally, we obtain the experimental results in Figure 5 and Figure 6 below.

As seen in Figure 5, the smallest MAE and RMSE are obtained for  $k = 30$  in the ML100K dataset. It can be seen in the figure that SRKG<sub>FCA</sub> has lower error values than the SRKG algorithm in all cases for the same  $k$ . In the extreme case of  $k$  ( $k = 1$ ), the error of SRKG<sub>FCA</sub> can still be at a low level, which indicates that SRKG<sub>FCA</sub> has a



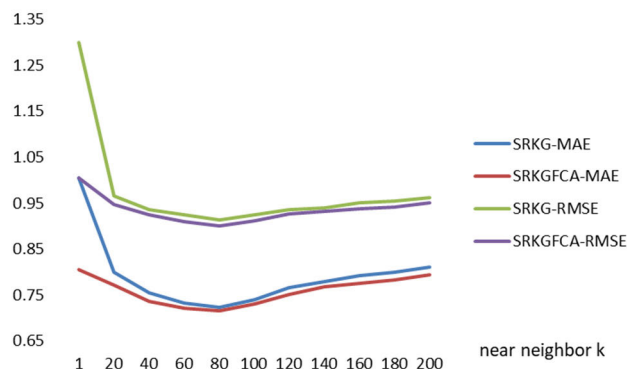


FIGURE 6. MAE and RMSE variation of PR1 and PR1<sub>FCA</sub> in the ML1M dataset.

better recommendation for ‘cold start’; at the same time, the extreme cases of  $k$  value often occur in practical applications, which means that the algorithm has a stronger ability to deal with extreme cases after integrating FCA. In this case, we can basically conclude that the integrated FCA effectively solves the problem of the SRKG algorithm ignoring user factors. In particular, the significant improvement in the effectiveness of the SRKGFCA algorithm in extreme cases ( $k = 1$ ) in terms of RMSE error is due to the fact that the algorithm is able to find the set of nearest neighbors of each item more accurately instead of generalizing to  $k$  nearest neighbors.

For the ML1M dataset with a large amount of data (e.g., Figure 6), the two algorithms achieve the lowest error at  $k = 40$ , and the SRKGFCA algorithm has a lower error deviation, confirming the conclusions drawn in Figure 5. In general, the SRKGFCA algorithm shows better recommendation results and confirms the effectiveness of the integrated FCA.

### 3) COMPARISON EXPERIMENTS AND ANALYSIS OF RESULTS

To verify the effectiveness and rationality of SRKG and SRKGFCA, we experimentally compared them with the current recommendation algorithms, which performed well on the MovieLens dataset. Benchmark methods are as follows

**ItemKNN** [28]: Item-based collaborative filtering recommendation algorithm proposed by Rosing et al. By integrating the score similarity into the classic Amazon item collaborative filtering recommendation algorithm [17], this algorithm achieves good recommendation effect and is one of the benchmark comparison algorithms for most current recommendation algorithms.

**UserKNN+FCA** [21]: A user-based collaborative filtering recommendation algorithm based on FCA and concept lattice proposed by Zou C et al in 2015. By integrating FCA into the coordinated filtering recommendation algorithm, the authors can more effectively mine the neighbor relationship between users. It is a good algorithm in the field of FCA recommendation in recent years, but it cannot be run in large-scale data sets. It is worth noting that compared with recommendation algorithms based on knowledge graph or

deep matrix factorization, the recommendation effect based on FCA is relatively poor. Therefore, only the representative UserKNN+FCA [21] algorithm is selected in this section, instead of other FCA-based algorithms [22].

**NeuMF** [17]: He et al. proposed the deep matrix factorization NeuMF model combining probabilistic computation and multi-layer perceptron in 2017. This method performs deep decomposition of the rating matrix through deep learning, so as to achieve high-quality semantic recommendation. It is the reference model for most deep matrix factorization models at present.

**DMF** [18]: Xue et al. proposed an improved deep matrix factorization model through normalized cross-entropy loss to deal with the shortcomings of NeuMF in dealing with user ratings. It is a good algorithm in cryptic meaning recommendation algorithms in recent years, and the DMF model proposed by Xue et al. is also a model used in this paper to mine cryptic meaning.

**TransHR+SVD** [20]: A semantic recommendation algorithm implemented by Yuan Quan et al. in the past two years through TransHR model and SVD matrix factorization model. They introduce external semantic knowledge through knowledge graph representation learning, and then use matrix factorization to mine cryptic meaning, and achieve good recommendation results.

**MC+DMF** [19]: implicit recommendation algorithm implemented by Shi Jiarong et al., through matrix completion method and DMF model in recent two years. The sparse user rating matrix is filled to reduce the sparsity of the matrix, and then the deep matrix factorization is carried out to realize the implicit recommendation algorithm with better recommendation effect.

For training the DMF model, the number of hidden layers is set to 2 for the ML100K dataset and 3 for the ML1M dataset, the maximum number of training iterations is 500 in each case, and the other hyperparameters are the default parameters of the DMF model. The experimental results can be found in Table 5.

As shown in Table 5, the SRKG and SRKGFCA algorithms proposed in this paper, which combine external knowledge graph semantics and internal hidden semantics, achieve excellent recommendation performance. It is worth noting that MAE and RMSE are error indicators and their corresponding lower values indicate better recommendation results, while HR and NDCG are recommendation accuracy indicators and their corresponding higher values indicate better recommendation results. In particular, compared to several existing semantic recommendation algorithms (NeuMF, DMF, TransHR+SVD, MC +DMF), SRKGFCA achieves the best results in both datasets, except for the RMSE index, which does not show much advantage. The reason for the lack of advantage in RMSE metric is that some movie titles are not matched with DBpedia and some movie entities correspond to too few triads.

Our experimental results with UserKNN+FCA and ItemKNN on the ML100K dataset confirm the effectiveness

**TABLE 5.** Experimental results for each type of recommendation algorithm.

Dataset	Algorithm	MAE	RMSE	HR@10	NDCG@10
ML100K	ItemKNN	0.778	0.985	0.606	0.334
	UserKNN+FC A	0.771	0.980	0.615	0.341
	NeuMF	0.740	0.936	0.670	0.395
	DMF	0.704	0.895	0.686	0.409
	TransHR+SV D	0.678	<b>0.864</b>	0.683	0.389
	MC+DMF	0.685	0.875	0.702	0.422
	SRKG	0.680	0.873	0.701	0.423
	SRKGFCFA	<b>0.673</b>	<b>0.864</b>	<b>0.710</b>	<b>0.431</b>
ML1M	ItemKNN	0.728	0.924	0.637	0.372
	UserKNN+FC A	-	-	-	-
	NeuMF	0.613	0.891	0.722	0.440
	DMF	0.613	0.877	0.732	0.451
	TransHR+SV D	0.595	<b>0.851</b>	0.738	0.447
	MC+DMF	0.603	0.866	0.745	0.469
	SRKG	0.607	0.862	0.749	0.472
	SRKGFCFA	<b>0.592</b>	0.854	<b>0.758</b>	<b>0.479</b>

of previous work on improved collaborative filter recommendation algorithms based on FCA. The main improvement lies in the fact that the k-nearest neighbor algorithm only considers the item factor in nearest neighbor selection and simply determines the nearest neighbor set of items through item similarity ranking. In contrast, the FCA-based nearest neighbor determination performs clustering of both users and items, improving both the prediction rating error (MAE, RMSE) and the accuracy of recommendation results (HR, HDCG). Since UserKNN+FCA on the ML1M dataset is not able to create a concept lattice in a short time, we did not test the four categories of metrics for this algorithm on this dataset.

Moreover, by comparing the experimental results of grouping implicit semantic-based recommendation algorithms (NeuMF, DMF) and knowledge graph-based algorithms (TransHR+SVD), we can conclude that the implicit semantic-based approaches are better at making sequential recommendations, since their HR and NDCG values are relatively high, which means that the content they recommend to users is more consistent with their preferences, while knowledge graph-based methods are better able to perform the rating prediction task since their rating error values (MAE and RMSE values) are relatively low. Comparing the DMF and NeuMF algorithms, the MC +DMF algorithm, and the SRKG and SRKGFCFA algorithms, it is found that effectively padding the model input vectors increases the accuracy of the recommendations, i.e., the problem of deep neural networks falling into local optima during the training process is mitigated. Combining the advantages of the two types of algorithms, it can be said that the combined SRKG and SRKGFCFA algorithms have excellent representation in prediction evaluation and also perform well in sequential recommendation tasks.

**TABLE 6.** Comparison of algorithm recommendation results before and after processing of the dataset.

Algorithm	Dataset	MAE	RMSE
SRKG	ML100K	0.680	0.873
	<b>ML100K after treatment</b>	<b>0.672</b>	<b>0.864</b>
	ML1M	0.607	0.862
	<b>Treated ML1M</b>	<b>0.599</b>	<b>0.855</b>
SRKGFCFA	ML100K	0.673	0.864
	<b>ML100K after treatment</b>	<b>0.661</b>	<b>0.856</b>
	ML1M	0.592	0.854
	<b>Treated ML1M</b>	<b>0.584</b>	<b>0.843</b>

In Table 5, the obvious shortcomings of our proposed semantic recommendation algorithm can be found by comparing the differences in MAE and RMSE between our proposed semantic recommendation algorithm and the TransHR+SVD algorithm in the two datasets. In both datasets, the MAE values are not significantly improved, and the RMSE values are even lower than those of the TransHR+SVD algorithm in the ML1M dataset. When analyzing the ternary data from DBpedia, we found that the main reason is that there is little or no ternary data for some of the movies in the data. To check whether the MAE and RMSE values are not significantly improved due to the small number of triad data of some movies, we deleted the records of movies with less than 10 occurrences in the DBpedia triad in the ML100K and ML1M datasets, i.e., 1544 movie records were retained in ML100K and 3460 movie records were retained in ML1M. We reran the SRKG and SRKGFCFA algorithms and found that both MAE and RMSE were reduced, as shown in Table 6.

After processing the data, the MAE and RMSE error values are further reduced, and it can be verified that some movie entities are missing or the corresponding triad data are too few to affect the recommendation effect of the SRKG algorithm and the SRKGFCFA algorithm in the above experiments. In addition, we find that the main reason for the missing movie entities or the too few corresponding triad data is that this work uses the public knowledge map of encyclopedias instead of the domain knowledge map constructed by the authors of the TransHR+SVD algorithm. However, exploring more accurate knowledge graph representation methods and constructing domain knowledge graphs is our main research direction for the future. In the meantime, we find that the semantic recommendation algorithm proposed in this paper still dominates the recommendation accuracy in the comparison between HR and NDCG.

## VI. CONCLUSION

In this paper, we propose a semantic recommendation algorithm based on a knowledge graph and formal concept analysis (FCA) to address the shortcomings of the semantic recommendation algorithm based on a collaborative filtering algorithm. We address the problem that the collaborative

filtering algorithm does not consider user or item factors when selecting nearest neighbors by combining formal concepts that can simultaneously cluster objects (users) and attributes (items), further reducing the error in predicting ratings. At the same time, we address the problem that the conceptual grid of FCA cannot be generated in large-scale datasets. Based on previous work, we propose a method to calculate the fuzzy concept area and a heuristic to construct formal concepts, so that the semantic recommendation algorithm integrating FCA in this paper can run on different datasets. We also selected several representative recommendation algorithms as benchmark methods for comparison and verified that our proposed semantic recommendation algorithm based on a knowledge graph and FCA achieves the best results on various metrics. Our next work will address how to effectively use FCA for better nearest neighbor discovery, how to effectively construct an intra-domain knowledge graph, e.g., a knowledge graph for the movie domain, and how to effectively use deep neural networks for representation learning of large-scale knowledge graphs to further improve the accuracy of semantic recommendations.

## REFERENCES

- [1] Y. Cao, X. Wang, X. He, Z. Hu, and T.-S. Chua, "Unifying knowledge graph learning and recommendation: Towards a better understanding of user preferences," in *Proc. World Wide Web Conf.*, May 2019, pp. 151–161.
- [2] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: Item-to-item collaborative filtering," *IEEE Internet Comput.*, vol. 7, no. 1, pp. 76–80, Jan. 2003.
- [3] F. Zhang, N. J. Yuan, D. Lian, X. Xie, and W.-Y. Ma, "Collaborative knowledge base embedding for recommender systems," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 353–362.
- [4] Q. Shambour, "A deep learning based algorithm for multi-criteria recommender systems," *Knowl.-Based Syst.*, vol. 211, Jan. 2021, Art. no. 106545.
- [5] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann, "DBpedia—A crystallization point for the Web of Data," *SSRN Electron. J.*, vol. 7, no. 3, pp. 154–165, 2009.
- [6] J. Hoffart, F. M. Suchanek, K. Berberich, and G. Weikum, "YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia," *Artif. Intell.*, vol. 194, pp. 28–61, Jan. 2013.
- [7] D. Vrandečić and M. Krötzsch, "Wikidata: A free collaborative knowledgebase," *Commun. ACM*, vol. 57, no. 10, pp. 78–85, Sep. 2014.
- [8] G. A. Miller, "WordNet: A lexical database for English," *Commun. ACM*, vol. 38, no. 11, pp. 39–41, 1995.
- [9] B. Xu, Y. Xu, J. Liang, C. Xie, B. Liang, W. Cui, and Y. Xiao, "CN-DBpedia: A never-ending Chinese knowledge extraction system," in *Proc. Int. Conf. Ind., Eng. Other Appl. Appl. Intell. Syst.*, 2017, pp. 428–438.
- [10] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 1–9.
- [11] Z. Wang, J. Zhang, J. Feng, and Z. Chen, "Knowledge graph embedding by translating on hyperplanes," in *Proc. AAAI Conf. Artif. Intell.*, vol. 28, 2014, pp. 1–8.
- [12] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, "Learning entity and relation embeddings for knowledge graph completion," in *Proc. AAAI Conf. Artif. Intell.*, Austin, TX, USA, 2015, vol. 29, no. 1, pp. 1–7.
- [13] G. Ji, S. He, L. Xu, K. Liu, and J. Zhao, "Knowledge graph embedding via dynamic mapping matrix," in *Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics, 7th Int. Joint Conf. Natural Lang. Process.*, 2015, pp. 687–696.
- [14] H. Wang, M. Zhao, X. Xie, W. Li, and M. Guo, "Knowledge graph convolutional networks for recommender systems," in *Proc. World Wide Web Conf.*, May 2019, pp. 3307–3313.
- [15] Y. Wu, Y. ZHao, and S. Wei, "Collaborative filtering recommendation algorithm based on interval-valued fuzzy numbers," *Int. J. Speech Technol.*, vol. 50, no. 9, pp. 2663–2675, Sep. 2020.
- [16] G. Xu, G. Jia, L. Shi, and Z. Zhang, "Personalized course recommendation system fusing with knowledge graph and collaborative filtering," *Comput. Intell. Neurosci.*, vol. 2021, pp. 1–8, Sep. 2021.
- [17] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proc. 26th Int. Conf. World Wide Web*, 2017, pp. 173–182.
- [18] H.-J. Xue, X. Dai, J. Zhang, S. Huang, and J. Chen, "Deep matrix factorization models for recommender systems," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 3203–3209.
- [19] S. Jiarong and L. Jinhong, "Recommendation algorithm via fusing matrix completion and deep matrix factorization," *Appl. Res. Comput.*, vol. 38, no. 8, pp. 2376–2380, 2021.
- [20] Y. Quan, C. Zhenhua, and J. Yang, "A movie recommendation algorithm based on knowledge graph and collaborative filtering," *Comput. Eng. Sci.*, vol. 42, no. 4, pp. 714–721, 2020.
- [21] C. Zou, D. Zhang, J. Wan, M. M. Hassan, and J. Lloret, "Using concept lattice for personalized recommendation system design," *IEEE Syst. J.*, vol. 11, no. 1, pp. 305–314, Mar. 2017.
- [22] H. Mezni and T. Abdeljaoued, "A cloud services recommendation system based on fuzzy formal concept analysis," *Data Knowl. Eng.*, vol. 116, pp. 100–123, Jul. 2018.
- [23] Y. Jiang, "Semantifying formal concept analysis using description logics," *Knowl.-Based Syst.*, vol. 186, Dec. 2019, Art. no. 104967.
- [24] P. du Boucher-Ryan and D. Bridge, "Collaborative recommending using formal concept analysis," *Knowl.-Based Syst.*, vol. 19, no. 5, pp. 309–315, Sep. 2006.
- [25] B. Ganter and R. Wille, *Formal Concept Analysis: Mathematical Foundations*. Berlin, Germany: Springer, 2012.
- [26] Y. Jiang and M. Yang, "Semantic search exploiting formal concept analysis, rough sets, and Wikipedia," *Int. J. Semantic Web Inf. Syst.*, vol. 14, no. 3, pp. 99–119, Jul. 2018.
- [27] P. N. Mendes, M. Jakob, A. Garcia-Silva, and C. Bizer, "DBpedia spotlight: Shedding light on the Web of Documents," in *Proc. 7th Int. Conf. Semantic Syst.*, Sep. 2011, pp. 1–8.
- [28] Y. Koren, "Factorization meets the neighborhood: A multifaceted collaborative filtering model," in *Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2008, pp. 426–434.
- [29] D. M. Abdulqader, A. M. Abdulazeez, and D. Q. Zeebaree, "Machine learning supervised algorithms of gene selection: A review," *Mach. Learn.*, vol. 62, no. 3, pp. 233–244, Apr. 2020.



**LINA WEI** received the B.S. and M.S. degrees in computer science, in 2006 and 2009, respectively. She is currently pursuing the Ph.D. degree with the School of Computer Science, South China Normal University. After graduation, she has been working in colleges and universities. She has published several papers on natural language processing and deep learning. Her research interests include solving challenging problems in recommender systems with knowledge graphs and deep learning. Her current research interests include natural language processing and data mining.



**XINGTU ZHU** received the B.S. degree in computer science from Guangdong Pharmaceutical University, in 2018, and the M.S. degree in software engineering from South China Normal University, in 2022. Since his graduation, he has been with the Center for Teacher Development of Tianhe District, Guangzhou. His research interests include natural language processing, data mining, and solving challenging problems in recommender systems with mathematical models.