ISSN 2278-3091

Volume 9, No.5, September - October 2020

International Journal of Advanced Trends in Computer Science and Engineering

Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse268952020.pdf

https://doi.org/10.30534/ijatcse/2020/268952020

Fusion Knowledge Graph and Collaborative Filtering Recommendation Algorithm

Dr Leelavathi Rajamanickam[1], Qin FengPing[2]

[1] Center for Software Engineering,
Faculty of Engineering, the Built Environment and Information Technology, SEGI University, Malaysia,
leelavathiraj@segi.edu.my

[2] Center for Software Engineering,
Faculty of Engineering, the Built Environment and Information Technology, SEGI University, Malaysia,
331031322@qq.com

ABSTRACT

The influence of data sparse, the collaborative filtering recommendation algorithm has the problem of inaccurate recommendation. Therefore, this paper proposes a collaborative filtering algorithm that fuses knowledge graph. By introducing rich content information of items into the item-based collaborative filtering algorithm, this algorithm effectively makes up for the fact that the collaborative filtering algorithm ignores the content information of items. Experiments show that the algorithm is better than the original algorithm to some extent, so as to alleviate the problem of data sparse.

Key words: fusion, filtering, algorithm, methods

1. INTRODUCTION

In recent years, with the gradual increase in the size of Internet data, people have found it increasingly difficult to pick out products that are suitable for them from massive amounts of information. How to find the content that users are interested in from a large amount of information and recommend it to users is the main problem of recommendation system research. Collaborative filtering algorithm is the earliest and relatively well-known recommendation domain algorithm. Its main function is prediction and recommendation. Traditional recommendation algorithms only rely on the user's historical behavior data to model the user, and then make recommendations for the user. Because the scoring data used for collaborative filtering recommendation calculations is usually very sparse, the recommendation performance of collaborative filtering recommendation algorithms is significantly reduced. Therefore, scholars consider adding feature descriptions of items in collaborative filtering algorithms to improve the performance of the recommendation system, and knowledge graph contain item attributes and various types of relationships, which can enrich the description of users and items, enhance the mining ability of recommendation algorithms, and improve the accuracy, variety and

interpretation of the recommendations. Knowledge graph-based recommendation system architecture has become
another research hotspot. To effectively recommend, based on the basic idea of item collaborative filtering recommendation and knowledge graph, this paper proposes a collaborative filtering algorithm that fuses knowledge graph. The levels of fusion are used by applying weighted sum rule for the biometric fusion process. The main contents include:
1) Map the item entities in the knowledge graph into entity vectors and calculate the content similarity of these items. 2) Use the user's behavior matrix to calculate the similarity of the items, and then fuse the two items' similarities to generate an item fusion similarity matrix. 3) Based on the item similarity matrix, calculate the predicted scores for each user for items that have not generated behavior, and then generate a recommended item list for the user based on these predicted scores. 4) Tested on the public data set Movie Lens. The test results are compared in terms of accuracy, recall and coverage. The experimental results show that the algorithm in this paper has a significant improvement over the traditional item-based collaborative filtering algorithm.

2. METHODS

Filter feature selection algorithms are capable to boost classification accuracy and diminish computational complexity by extracting relevant information through supervised learning. Collaborative filtering methods are mainly divided into neighborhood collaborative filtering methods [1] and model-based collaborative filtering methods [2]. Both methods are based on the user-item rating matrix, but as the number of users and items increases, the user-item rating matrix will become larger and larger. As a result, the performance of collaborative filtering algorithms continues to decrease. In order to improve the execution efficiency of the recommendation algorithm, Sarwar et al. [3] proposed an item-based prediction algorithm, established a pre-calculation model of item similarity, and improved the online scalability of recommendation system modifications. Roh et al. [4] used a hierarchical clustering algorithm to cluster users and items and made recommendations based on the weighted sum of the root node to the leaf node. Hernando [5] et al. Proposed a new technique for predicting user taste based on a collaborative filtering algorithm that decomposes the evaluation matrix into two non-negative matrices. Santhini et al. [6] proposed the basic idea of collaborative filtering recommendation on big data by clustering method, which improved the efficiency of the recommendation algorithm to a certain extent. However, these algorithms mainly use the external information of the item-user rating matrix, and do not fully consider the internal information of the item itself. Ignoring this important information will inevitably distort the recommended results. With the development of knowledge graph technology, the industry has accumulated a lot of open semantic data, such as Freebase [7], DBpedia [8], which contains rich connotation knowledge of items. There is also a lot of work to introduce knowledge graph into recommendation systems, such as: feature-based recommendation algorithms [9] weaken knowledge graphs into item attributes and uniformly use the attributes of users and items as input to the recommendation algorithm, but this method cannot be used efficiently All the information of the knowledge graph. Path-based recommendation algorithm [10] regards the knowledge graph as a heterogeneous information network, and then constructs features based on meta-path or meta-graph between items, fully and intuitively utilizes the network structure of the knowledge graph, but the workload Big. The literature [11] combines the structural features of the knowledge graph and fuses ontology into a collaborative filtering algorithm. Existing research shows that the knowledge graph representation learning method can embed the knowledge graph in a low-dimensional semantic space, and can use the vector of continuous values to reflect the structural characteristics of the knowledge graph. This method can efficiently calculate the semantic relationship between entities. The literature [12] combines the knowledge graph representation learning algorithm with collaborative filtering based on implicit feedback, and then converts the original data into preference sequences for parameter learning, which enhances the performance of collaborative filtering recommendation algorithms. Wu [13] et al. Based on knowledge graph representation learning method, embedded semantic data into low-dimensional space, and finally incorporated item semantic information into collaborative filtering recommendations. Zhang [14] design three components to extract items' semantic representations from structural content, textual content and visual content, respectively. And then propose a final integrated framework, to jointly learn the latent representations in collaborative filtering as well as items' semantic representations from the knowledge base. The results reveal that knowledge graph recommendation algorithm based on deep learning is superior to traditional recommendation models based on collaborative filtering in the recommendation effect. Existing methods are limited to considering the item-user rating matrix information external to the item, ignoring the information of the item itself. Based on this, the collaborative filtering algorithm proposed in this paper uses the knowledge graph representation learning algorithm to obtain the semantic information of items, calculates the semantic similarity between items, and integrates the semantic information of items in the recommendation process to better recommend for users.

3. FUSION KNOWLEDGE GRAPH AND COLLABORATIVE FILTERING RECOMMENDATION ALGORITHM

This paper proposes a collaborative filtering algorithm that incorporates a knowledge map. The basic idea is: map the item entities in the knowledge map to an entity vector matrix and calculate the content similarity of these items; use the user's behavior matrix on items to calculate the Similarity. The two items' similarities are then fused to generate an item fusion similarity matrix. Based on the item similarity matrix, each user calculates their predicted scores for items that have not produced behavior, and then generates a recommended item list for the user based on these predicted scores. Figure 1 shows the flowchart of the algorithm in this paper:

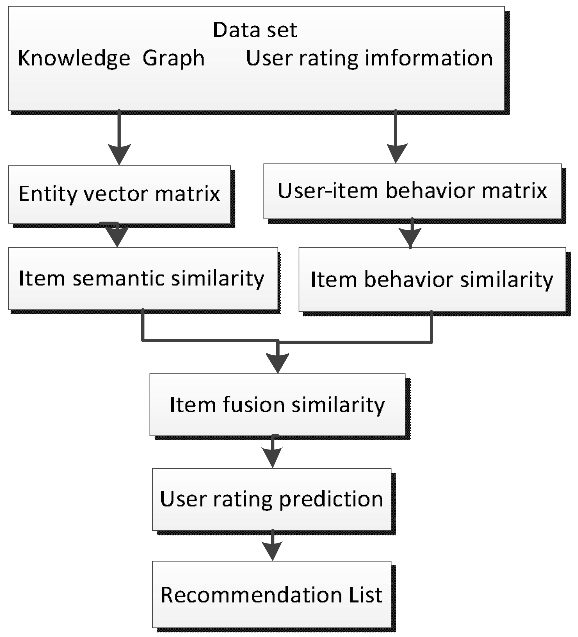

Figure 1: Fusion Knowledge Graph and Collaborative Filtering Recommendation Flowchart and Algorithm

3.1 Knowledge Graph Vectorization Representation

The knowledge graph contains the entities of the items, the entities related to the items, and the relationship attributes of these entities. In the knowledge graph, the richer the connections between the physical nodes corresponding to the items, the more similar the semantics of these items are. Therefore, the entities and relationships in the knowledge graph are used to describe the semantic information of the items in the recommendation system, thereby improving the accuracy of the vectorized representation of the items. Based on this idea, we can use the technology of knowledge representation to embed a heterogeneous network of knowledge graphs into a continuous low-dimensional vector space while retaining some information in it. Then based on the vectorized item information, a similarity algorithm is used to measure the similarity between the items. This paper uses the TransE algorithm [15] in the translation model to obtain the entity vector. Equation (1) gives the representation of the vector:

$$\|h + r\| \approx t \qquad (1)$$

Among them, *h* and *t* are the head entity vector and the tail entity vector, respectively, and r is the relationship vector. The normal form $\|\cdot\|$ in the expression can choose L1 norm and L2 norm.
The closer the entities in the knowledge graph, the more similar the vectors are

According to the definition of the TransE algorithm, a triple (h, r, t) in the knowledge graph can be trained using the loss formula shown in equation (2).

$$\mathrm{L} = \sum_{(h,r,t)\in S} \sum_{(h',r',t')\in S'_{(h,r,t)}} \max(0, \|h + r - t\| - \|h' + r - t'\| + \gamma) \qquad (2)$$

Among them, h' and t' are the wrong triples vectors. As negative samples for training, this type of negative samples is the positive samples defined by the TransE algorithm, that is, the original or tail entities of the correct triples are randomly replaced Derived from other entities. γ is the size of the distance, which is used to control the distance between the positive and negative samples, Generally, $\gamma = 1$. $\max(x, y)$ represents taking the largest value of x and y, so that the value of each accumulated child is not less than zero.

Throughout the training process, TransE uses the idea of maximum spacing to open the distance between the positive and negative sample vectors. Continuous iteration through stochastic gradient descent to optimize the loss function. The final training results are: 1) similar entities in the knowledge graph, the closest distance in low-dimensional space; 2) the vector of the head node plus the relationship vector is basically equal to the vector of the tail node.

3.2 Semantic similarity and behavioral similarity

Based on the embedded vector of the item obtained by the vectorized representation method, we give a calculation method of the item's semantic similarity. First, the item entities and relationships in the knowledge graph are mapped into an *n* dimensional space, and the item I_i is embedded into an *n* dimensional vector:

$$I_i = (E_{1i}, E_{2i}, \dots, E_{ni})^T \qquad (3)$$

Where E_{ni} represents the value of the embedding vector of item I_i in n dimensions. When embedding entities and relationships in the knowledge graph, the scoring function used is calculated based on Euclidean distance. Therefore, in order to accurately describe the similarity between items, the semantic similarity of item vectors is also measured using Euclidean distance. First calculate the distance between items:

$$d(I_i, I_j) = \sqrt{\sum_{k=1}^{n} (E_{ki} - E_{kj})^2} \qquad (4)$$

The Euclidean distance is a value greater than or equal to 0. It is reduced to [0,1] by the following operation:

$$sim_{KG}(I_i, I_j) = \frac{1}{1 + d(I_i, I_j)} = \frac{1}{1 + \sqrt{\sum_{k=1}^{n} (E_{ki} - E_{kj})^2}} \qquad (5)$$

The larger the calculation result of this formula, the greater the semantic similarity between the two items. When the value is 1, the two items have the greatest semantic similarity; when the value is close to 0, we think the two items are almost completely different.

User behavior data can reflect the classification of items from the user's perspective. The item-based collaborative filtering algorithm belongs to a neighborhood-based algorithm, which calculates the similarity between items based on the behavior data of all users on the items, and then recommends similar items to the user based on the user's historical behavior. There are many types of user behavior data on items. This paper uses the most common scoring information as the basis for item vectorization.

Assuming that the recommendation system contains m users U=（$U_1, U_2, \dots U_m$） and n items I=($I_1, I_2, \dots I_n$), the user's rating information for the items can be expressed as an $m \times n$ matrix $R_{m \times n}$.

$$R = \begin{bmatrix} R_{11} & R_{12} & L & R_{1n} \\ R_{21} & R_{22} & L & R_{2n} \\ L & L & L & L \\ R_{m1} & R_{m1} & L & R_{mn} \end{bmatrix}$$

Item I_i represents a vector of *m* dimensions, and its value in each dimension is the user's rating of it.

$$I_i = (R_{1i}, R_{2i}, \dots, R_{mi})^T \qquad (6)$$

According to the cosine similarity formula we get the item similarity based on user behavior.

$$sim_{UB}(I_i, I_j) = \frac{I_i I_j}{\|I_i\| \|I_j\|} = \frac{\sum_{k=1}^{m} R_{ki} \times R_{kj}}{\sqrt{\sum_{k=1}^{m} R_{ki}^2} \times \sqrt{\sum_{k=1}^{m} R_{kj}^2}} \qquad (7)$$

The larger the calculation result of this formula, the greater the similarity of the item I_i and the item I_j based on the user behavior. When the value is 1, the two items are the same; when it is 0, they are completely different. In particular, when the user u has not scored the item i, the variable R_{ui} is 0.

In order to improve the reliability of item similarity, this paper uses the following formula to fuse the item similarity $sim_{KG}(I_i, I_j)$ based on knowledge graph and the item similarity $sim_{UB}(I_i, I_j)$ based on user behavior.

$$\mathrm{sim}(I_i, I_j) = \alpha \cdot sim_{KG}(I_i, I_j) + (1 - \alpha) sim_{UB}(I_i, I_j) \qquad (8)$$

The parameter α is a weight that controls the proportion of two similarities in the recommendation process.
When $\alpha > 0.5$, the similarity of the items obtained based on the knowledge graph occupies a larger proportion in the final similarity; when $\alpha < 0.5$, the similarity based on user behavior is the main factor. After the similarity between all items is fused, we express the similarity of all items in the form of a matrix as follows:

Table 1: Similarity matrix of items

物品	$Item_1$	$Item_2$	…	$Item_j$	…	$Item_n$
$Item_1$	1	s_{12}	…	s_{1j}	…	s_{1n}
$Item_2$	s_{21}	1	…	s_{2j}	…	s_{2n}
⋮	⋮	⋮		⋮		⋮
$Item_i$	s_{i1}	s_{i2}	…	s_{ij}	…	s_{in}
⋮	⋮	⋮		⋮		⋮
$Item_n$	s_{n1}	s_{n2}	…	s_{nj}	…	1

In the similarity matrix of Table 1, s_{ij} is the similarity between item $Item_i$ and item $Item_j$, and$s_{ij} = s_{ji}$.
When $i = j$, then $s_{ij} = 1$, all items have a similarity to themselves of 1.

3.3 Prediction Score

The recommendation system mainly makes predictions on the unscored products of the user, and finally recommends the products with the highest Top-N scores to the user. It is not necessary to use all the items in the prediction; usually the K nearest neighbor sets with the highest similarity to the current item can be selected. In the process of selecting the nearest neighbors, if the value of K is too large, adding neighbors with low similarity to the prediction will bring additional noise to the prediction; if the value of K is too small, the nearest neighbor used for prediction is insufficient and the accuracy is poor. The prediction score of user u for item i is calculated by the following formula:

$$p_{ui} = \frac{\sum_{j \in N(u) I S(i,k)} (s_{ij} \times R_{uj})}{\sum_{j \in N(u) I S(i,k)} s_{ij}} \qquad (9)$$

Among them, s_{ij} indicates the similarity between item I_i and item I_j, R_{uj} indicates the user *u*'s rating of item j in the existing scoring matrix, $N(u)$ indicates the set of items that user u has rated, and $S(i,k)$ represents the set of *k* items that are most similar to item *i* The meaning of this prediction score formula is that the items that are more similar to the items with higher scores in user history can get higher prediction scores.

Based on the algorithm flow of Figure 1, the algorithm is described as follows:

Input: *Item-user rating matrix* $R_{m\times n}$, *the knowledge graph of the field where the item is located*

Output: *N recommendation sets*

1) *Transform the entities and relationships in the knowledge graph into the embedding vectors of the items, respectively.*

2) *According to the results obtained in formulas (5) and (1), the semantic similarity between items* $sim_{KG}(I_i, I_j)$ *is calculated.*

3) *Calculate the similarity* $sim_{UB}(I_i, I_j)$ *between the item and the item using the item-user rating matrix* $R_{m\times n}$ *according to formula (7)*

4) *Select the fusion ratio, calculate the fusion similarity* $\mathrm{sim}(I_i, I_j)$ *according to formula (8), and generate the item-item similarity matrix.*

5) *Based on 4), finally through the prediction scoring formula (9), the item set required for the final recommendation is recommended to the user.*

4. EXPERIMENT AND RESULT ANALYSIS

In order to verify the reliability and effectiveness of the algorithm in this paper, the Movie Lens data set was used to construct a training set and a test set using a five-fold cross-validation method, and then an experimental comparison was made between the proposed algorithm and the item-based collaborative filtering algorithm. We evaluate the recommendation list of Top N given by the recommendation system, and use the precision, recall, and coverage to measure the prediction accuracy of the recommendation system.

4.1 Data set

The dataset used in this paper is the Movie Lens dataset, which is organized by the Group Lens research group of the University of Minnesota and has been widely used in academia. The Movie Lens contains data sets of different sizes and the Movie Lens-1M data set used in this experiment contains 1000209 pieces of rating data, involving a total of 6040 users and 3883 movies. In order to verify the effectiveness of the algorithm in this paper, this experiment uses a 5-fold cross-validation method to randomly split the data set into five mutually exclusive data subsets. Each experiment randomly uses one of the subsets as the test set, and the remaining 4 subsets as the training set, and trains and tests the model. Finally, the average of the results of the 5 tests is compared and analyzed.

We evaluate the top N recommendation list given by the recommendation algorithm using precision, recall, and coverage indicators. Precision and recall are two indicators commonly used in evaluating recommendation systems. The recall can measure the recall rate of the recommendation system, reflecting the ratio of recommended items to the items that users like.
The precision measures the accuracy rate of the recommendation system and reflects the recommendation level of the recommendation system. Coverage refers to the proportion of total items recommended by the recommendation system for all users. High coverage indicates that the recommendation system's ability to recommend unpopular items is better. The first two indicators can be derived from the confusion matrix, as shown in Table 2.

Table 2: Confusion matrix

system	User dislikes	Users like
Not recommended	TN	FN
recommend	FP	TP

N_{TP} represents the number of TP samples, N_{FP} represents the number of FP samples, N_{TP} represents the number of TN samples, N_{FN} represents the number of FN samples.

According to the confusion matrix and definition, the formulas for precision and recall are as follows:

$$\text{Precision} = \frac{N_{TP}}{N_{TP} + N_{FP}}$$

$$\text{Recall} = \frac{N_{TP}}{N_{TP} + N_{FN}}$$

Coverage can be defined as:

$$\text{coverage} = \frac{U_{u\in U} R(u)}{|I|}$$

Among them, I is the total number of items in the recommendation system, $R(u)$ is a list of items of length N recommended by the recommendation system for the user u, and U is the set of all users.

4.2 Analysis of results

The recommendation algorithm combines the similarity of items based on the knowledge graph and the similarity of items based on user behavior, finds k neighboring movies for users, and recommends 10 movies for users through rating prediction. The nearest neighbor value k = 30 selected in the experiment. We select the fusion ratio of similarity in the sequence with Interval range [0, 1] and spacing 0.1, and compare with the item-based collaborative filtering algorithm. When the fusion degree is 0, it means that the entire recommendation algorithm is an item-based collaborative filtering algorithm; when it is 1, it means that the entire recommendation algorithm is a recommendation algorithm based on the content of the knowledge graph. The algorithm is tested on the validation set and the optimal fusion degree is determined to be 0.6 based on the precision of the recommended results. The following are the experimental results of the algorithm with different degrees of fusion on the test set.

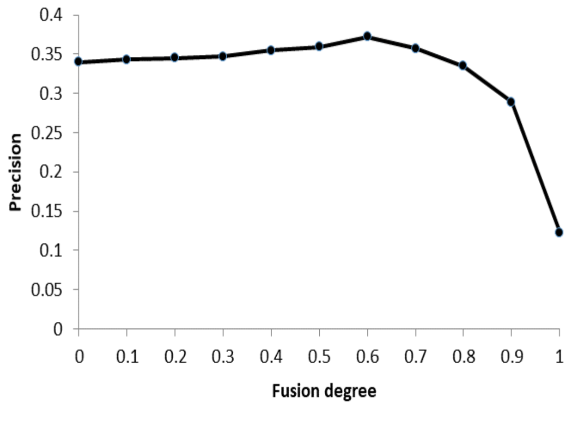

Figure 2: Precision at different fusion degree

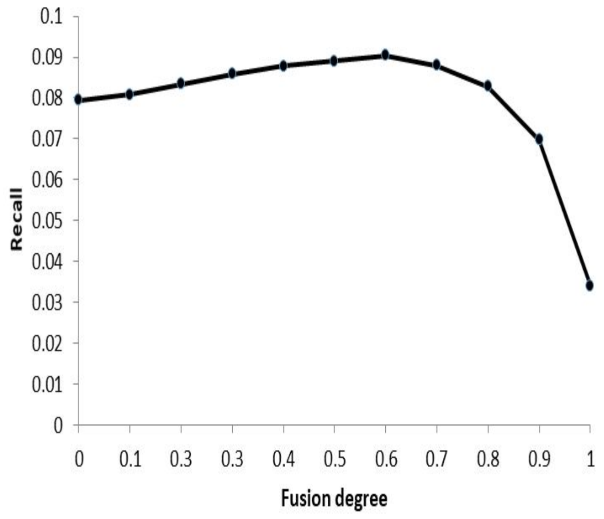

Figure 3: Recall at different fusion degree

The Figures 2 and 3 show the precision and recall of the algorithm under different fusion degree, where the fusion degree of 0 indicates the experimental results of the collaborative filtering algorithm based on items. From the experimental results, it can be seen that when the fusion degree is 0.6, the precision and recall of the algorithm reach the peak, among which the precision and recall are 0.372 and 0.0904 respectively, which is better than the precision and recall of the collaborative filtering algorithm based on items.

The Figure 4 shows the coverage of the algorithm under different fusion degrees. It can be seen from the experimental results that the coverage rate will continue to increase with the increase of the fusion degree of the algorithm.

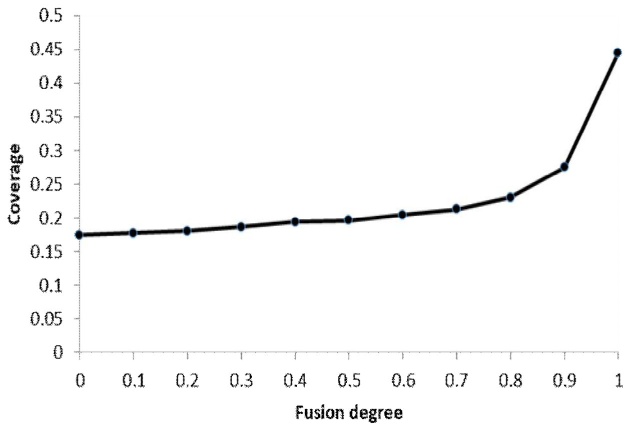

Figure 4: Coverage at different fusion degree

When the fusion degree is 1, it means that only the similarity of items based on the knowledge graph is used for recommendation, and the coverage reaches the highest. Experimental results show that the algorithm with a fusion degree α of 0.6 can also perform well on the test set. At this fusion degree, compared with the traditional item-based collaborative filtering algorithm, the collaborative filtering recommendation algorithm fusion knowledge graph proposed in this paper has a significant improvement in precision, recall and coverage.

5. CONCLUSION

Aiming at the sparsity problem of traditional collaborative filtering algorithms, this paper proposes a recommendation algorithm that fuses knowledge graph, and gives the framework of the algorithm, and describes each step of the algorithm in detail. This method combines the vectorized representation of items in the knowledge graph with a collaborative filtering algorithm, and uses the semantic information of the items extracted from the knowledge graph to compensate for the shortcomings of the collaborative filtering algorithm based on items that does not consider the content information of the item itself. The essence of the algorithm is to use the semantic information of the items in the knowledge graph and the item information contained in the user's behavior data on the items. By describing the similarity of the items from multiple angles, the recommendation effect is improved. Finally, this paper evaluates the performance of the algorithm by k-fold cross-validation and proves that the proposed collaborative filtering recommendation algorithm based on knowledge graph is effective.

ACKNOWLEDGEMENT

The authors thank SEGI university, Malaysia for providing financial support.

REFERENCES

1. J. S. Breese, D. Heckerman, C. Kadie. **Empirical analysis of predictive algorithms for collaborative filtering**, *in Proc. Conference of Uncertainty in Artificial Intelligence*, *Morgan Kaufmann,* 1998, pp. 43-52.
2. G. Adomavicius, A. Tuzhilin, **Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions**, *IEEE Transactions on Knowledge &Data Engineering*, 2005, vol. 17, no. (6), pp. 734-749.
3. B. Sarwar, G. Karypis, J. Konstan, et al. **Item-based collaborative filtering recommendation algorithms,** *in Proc. of the tenth international conference on World Wide Web,* 2001, pp. 285-295.
4. T.H. Roh, K.J. Oh, I. Han, **The collaborative filtering recommendation based on SOM cluster-indexing CBR**, *Expert Systems with Applications*, 2003, vol. 25, no. (3), pp. 413-423.
5. H. Antonio, B. Jesus, O. Fernando, **A non-negative matrix factorization for collaborative filtering recommender systems based on a Bayesian probabilistic model**. *Knowledge-Based Systems,* 2016, pp. 188-202.
6. M. Santhini, M. Balamurugan, M. Govindaraj, Collaborative **Filtering Approach for Big Data Applications based on Clustering**, International *Journal of Mathematics Computer Science and Information Technology*, 2015, vol. 2, no. (1), pp. 202-208.
7. K. Bollacker, C. Evans, P. Paritosh, et al. **Freebase: A Collaboratively Created Graph Database for Structuring Human Knowledge**, *in Proc. of ACM SIGMOD International Conference on Management of Data. New York, USA: ACM Press*, 2008, pp. 1247-1250.
8. C. Bizer, J. Lehmann, G. Kobilarov, et al, **A Crystallization Point for the Web of Data**, *Journal of Web Semantics*, 2009, vol. 7, no. (3), pp. 154-165.
9. S. Rendle, **Factorization machines with libFM**, *in ACM Conf. Transactions on Intelligent Systems and Technology,* May 2012.
10. Davidson, James, et al. **The YouTube video recommendation system**, *in Proc. of the 2010 ACM Conference on Recommender Systems, 2010, Barcelona, Spain,* 2010, DOI: 10.1145/1864708.186477.
11. Zijian ZHANG, Lin GONG, Jian XIE. **Ontology-based Collaborative Filtering Recommendation Algorithm**, *in Proc. of International Conference on Brain Inspired Cognitive Systems Berlin, Germany, Springer,* 2013, pp. 172-181.
12. F. Zhang, N. J. Yuan, D. Lian, et al. **Collaborative Knowledge Base Embedding for Recommender Systems**, in *Proc. of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York, USA: ACM Press,* 2016, pp. 353-362 ·
13. W. X. Yu, C. Q. Mai, H. Liu, C. He. **Collaborative filtering recommendation algorithm based on representation learning of knowledge graph**, *Journal of Electrical and Computer Engineering*, vol. 44, no. 2, pp. 226–232, 2018.
14. Zhang F, N. J. Yuan, D. Lian, et al. **Collaborative Knowledge Base Embedding for Recommender Systems,** *in Conf. 22nd ACM SIGKDD International Conference. ACM,* 2016.
15. A. Bordes, N. Usunier, A. Garcia-Duran, et al. **Translating Embeddings for Modeling Multi-Relational Data,** *Advances in Neural Information Processing Systems*, 2013, pp. 2787-2795.
16. M. Tengku, **Ensamble Based Multi Filters Algorithm for Tumor Classification in High Dimensional Microarray Dataset,** International Journal of Advanced Trends in Computer Science and Engineering, 2019, vol. 8, no.(1) pp. 116-123. DOI: 10.30534
17. G. Amirthalingam, **Multi-Biometric Authentication Using Deep Learning Classifier for Securing of Healthcare Data**, International Journal of Advanced Trends in Computer Science and Engineering, 2019, vol. 8, no. (4), pp.1340-1347. DOI: 10.30534