Journal of Applied Intelligent System (e-ISSN : 2502-9401 | p-ISSN : 2503-0493)
Vol. 8 No. 2, July 2023, pp. 162 – 172
DOI:

Harnessing Item Features to enhance Recommendation Quality of Collaborative Filtering

Folasade Olubusola Isinkaye
Department of Computer Science
Ekiti State University, Ado-Ekiti, Nigeria
Email: folasade.isinkaye@eksu.edu.ng
**Corresponding author*

Abstract - Recommendation systems provide ways of directing users to items that may be relevant to them by guiding them to relevant items that will be suitable to the users according to their profiles. Collaborative filtering is one of the most successful and mature techniques of recommender system because of its domain independent ability. Bayesian Personalized Ranking Smart Linear Model (BPRSLIM) is model-based collaborative filtering (CF) recommendation algorithm that usually reconstructs a scanty user-item matrix directly; also, using only user-rating matrix usually prevents the algorithm from accessing relevant information that could enhance its recommendation accuracy. Therefore, this work reconstructs BPRSLIM user-item rating matrix via item feature information in order to improve its performance accuracy. Comprehensive experiments were carried out on a real-world dataset using different evaluation metrics. The performance of the model showed significant improvement in recommendation accuracy when compared with other top-N collaborative filtering-based recommendation algorithms, especially in precision and nDCG with 30.6% and 22.1% respectively.

Keywords: Recommender Systems, Bayesian Personalized Ranking, Smart Linear Model, Item features, Collaborative Filtering.

1. INTRODUCTION

Recommender systems (RS) usually take advantage of the user's historical behaviors to suggest personalized ranked list of items to users [1]. RS could suggest items to user by using Collaborative Filtering (CF), Content-Based Filtering (CBF) or Hybrid Filtering (HF) techniques [2]. The emphasis in the present work is on collaborative filtering. CF utilizes the preferences of users and their similarities to generate reliable and personalized recommendations of items. The CF process of recommendation generation is illustrated in Figure 1.

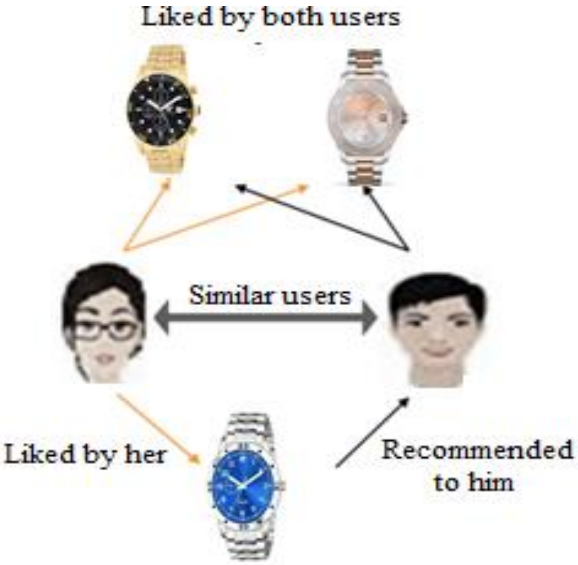
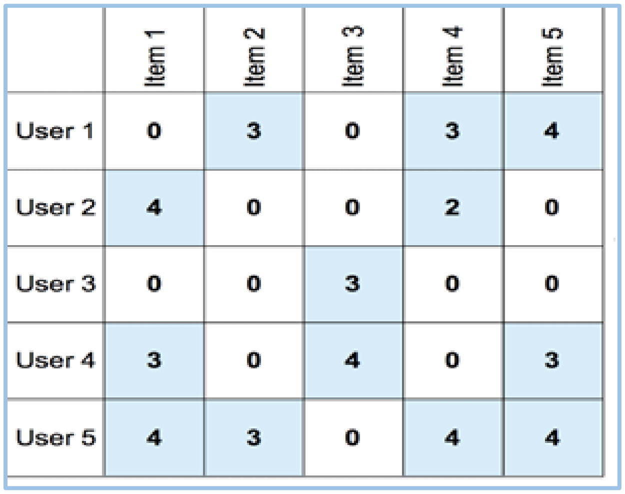


Figure 1. Recommendation generation process in CF

In recent time, CF is employed in evaluating very large number of commercial datasets [3, 4]. The rating matrix used in collaborative filtering recommendation is often characterized by the items that users have rated [5]. For instance, supposing an online cinema site has 2.5 million movies, it means each user is identified by a Boolean feature vector of 2.5 million movie items. The score of every item is determined by whether a user has observed similar movie in the past. Usually, the score of 1-5 specifies that users have watched the movie while a score of 0 specifies that users have not watched the movie. When many users are involved, a matrix comprising all vectors that represent the users could be employed to capture all the movies that users have watched in the past. This is always referred to as user-item interaction matrix. Typically**,** most large-scale applications are often characterized with vast quantity of items and users. In such situations, when users have even given several ratings, the user-item interaction matrix remain very scanty, in other words, the quantity of ratings in the user-item interaction matrix whose score is not 0 are extremely few. This weakness is usually called sparsity of data problem [6, 7], and it has a significant negative effect on the efficiency of recommendation generation. As a result of sparsity, sometimes, correlation between two specific users is always equal to 0, making recommender system ineffective. At times for sets of users that are completely related, such correlation connection may sometimes not be dependable. Since response time is a critical factor employed to estimate the efficiency of recommendation systems. It becomes very vital to implement reliable algorithms, which has the potentials to handle this circumstance competently to produce very accurate suggestions. Generally, suppose that ***U*** represents users and ***I*** represent the set of items scored by a user ***U***. The user-item matrix could be designated as $|\boldsymbol{U}| \times |\boldsymbol{I}|$ matrix, while the rating matrix is designated as ***M*** and the scores of the rating matrix is define as:

$$\boldsymbol{M}_{\boldsymbol{i},\boldsymbol{j}} = \begin{cases} 1 \; or \; 1 \ldots .5 & if \; \boldsymbol{i} \; rates \; \boldsymbol{j} \\ 0 & otherwise \end{cases} \qquad 1$$

The rating scale could be binary or be on a 5-level rating scale. In most real-life applications, the number of users $|\boldsymbol{U}|$ and items $|\boldsymbol{I}|$ are not always very high and as such, the rating matrix ***M*** is always very scanty. This means that most of the ratings in matrix ***M*** have a value of 0 as depicted in Figure 2. This is a major problem in recommender system, especially CF.



	Item 1	Item 2	Item 3	Item 4	Item 5
User 1	0	3	0	3	4
User 2	4	0	0	2	0
User 3	0	0	3	0	0
User 4	3	0	4	0	3
User 5	4	3	0	4	4

Figure 2. Sparse user rating matrix of a typical CF

Since Bayesian Personalize Ranking Smart Linear Model (BPRSLIM) is a variant of collaborative filtering algorithm, it is also plagued with low-density user-item matrix which frequently makes recommendation generation inaccurate [8]. In order to combat this challenge, this work integrates item features into the Bayesian Personalized Ranking Smart Linear Method (BPRSLIM) algorithm in order to improve its predictive capability and hence provide more accurate recommendations.

2. RELATED WORKS

Current exponential progression of the internet and the evolving prominence of electronic commerce have increased the volume of datasets [9] and reduced the efficiencies of the information filtering tools essential to extract valuable information. Recommendation systems were specifically designed to assist in bridging the gap between information collection and analysis by sieving accessible information to deliver relevant results to the users. Collaborative filtering-based algorithm is perhaps among the highest generally and extensively applied methods for recommendation systems [10-12]. The underlying concept of this approach is that if users have similar rating pattern before, there is the probability that they will rate related items in the future. The strength of collaborative filtering lies in that fact that it does not need any prior domain knowledge before recommendation can be made. However, CF suffers from sparsity problems [13, 14]. Sparsity of data is very common in real-word applications of CF because, most times the density of user-rating matrix is always extremely low which invariably leads to low representation of users within the matrix.

Numerous approaches of hybrid techniques have been employed by researchers in solving this problem. For example, Yang et al. [15] used an item-based CF to address both the sparsity and scalability problem of CF, instead of using user correlation; they found correlation between the items previously accessed by the target user in view of the feedback given by the target user. In Xu et al. [16], a generative model that predicts user's rating on previously unrated items was proposed, the model considers reviews alongside user hidden community and item groups relationship. Co-clustering algorithm was thereby employed for simultaneous clustering of the variables in the model. Huang et al. [17], extended the conventional CF algorithm with a novel clustering approach with a Hybrid CoClustering recommendation framework. The framework leverage sparsity by allowing users and items to be clustered into multiple groups and exploiting information from different sources such as rating matrix, user social network and knowledge base. You et al. [18] presented a technique to solve sparsity problem that used the combination of item clustering CF and the weighted slope. Wang et al. [7] solved the data sparsity of CF by using a cross-domain item embedding that is based on Co-clustering.

Dimensionality reduction techniques was used by Bokde et al. [19] to enhance neighborhood-based CF effectiveness and quality by decreasing the dimensions of the user-item space directly. They map large dimensional input space into a low dimensional latent space. Advanced approaches of dimensionality reduction that have been implemented include Principal Component Analysis (PCA) [20], Singular Value Decomposition (SVD) [21-23], Latent Semantic Indexing (LSI) and Probabilistic Latent Semantic Indexing (PLSI) [24]. Tang and Harrigton [25] and Deng et al. [26] proposed another matrix factorization solution that employed a two-stage randomized matrix factorization to handle large scale CF where Alternating Least Square (ALS) or Stochastic Gradient Descent is not available. Hao and Zheng [27] worked on the implementation and matrix decomposition based on collaborative filtering task on x 86 platforms.

Some attempts have been made to increase the performance accuracy of recommendation of CF with additional information. For instance, Zheng et al. [28] proposed ways to introduce contextual information into SLIM models in order to develop Contextual SLIM (CSLIM) models. Zheng et al. [29], their work pinpointed the significance of context similarity and integrated it into context-aware recommendation system. Zheng et al. [30] developed the contextual SLIM recommender techniques which were based on NBCF and MF by incorporating contextual information into the Sparse Linear Method (SLIM). The algorithms provided stability between efficiency and justification, and it was confirmed that CSLIM were more promising than context-aware recommendation systems. Fan and Ning [31] addressed sparseness of data and user item heterogeneity difficulties of CF by building a joint local sparse linear model. The model was able to learn several local sparse linear models (SLIM) for each user and item in the system. Christakopoulou and Karypis [32] were able to integrate higher-order data into SLIM to find the item-item and itemset-item relationships. Liu et al. [33] presented a unique approach that mathematically exploited flat and hierarchical side information concurrently with mathematical coherence. Then, a combined framework called HIRE which can model side information from varied sources for improved recommendation was introduced. The approach reported showed great promised. Also, Deng et al. [34] presented a three-level neural variational collaborative filtering (NVCF). In the first level, side information of user and item were integrated to mitigate sparsity of data problem of user rating matrix, in second level, a Stochastic Gradient Variational Bayes technique was used to estimate the range of latent user-item factors. The methods were reported showed significant improvement over the state-of-the-art hybrid CF and VAE-based approached. Also, Mahesh et al. [35] tried to improve the efficiency of collaborative filtering algorithm by introducing both user confidence and time context to the similarity measure techniques. They observed from the results of their experiments that there was performance accuracy improvement range of 16.2% in comparison with recent models.

The essence of this work is to determine if integrating item features into BPRSLIM could significantly enhance its recommendation quality in such a way that it could produce a more accurate, diverse and personalizes recommendation which will also enhance user experience in recommendation systems.

3. PROPOSED APPROACH

This section discusses the proposed approach, BPRSLIM that integrates item features for improved recommendation accuracy.

3.1. BPRSLIM with Item Features

SLIM is an efficient algorithm for detecting item to item associations with the help of statistical learning [36]. It estimates the correlation matrix as a least squares task instead of finding it with existing similarity measures [37], such as Pearson, Cosine, Euclidean distance etc. It does this by solving a regularized optimization problem (Eq. 2) [38] in order to identify the best coefficient matrix C for generating top-N recommendations of items to users.

$$\underset{C}{\text{minimize}} = \frac{1}{2} \| M - MC \| \frac{2}{F} + \frac{\beta}{2} \| C \| \frac{2}{F} + \lambda \| C \| 1 \qquad (2)$$
$$subject\ to \quad C \geq 0, \ \ diag(C = 0),$$

When SLIM uses Bayesian Personalized Ranking (BPR) as its learning algorithm instead of the least squares learning algorithm, the algorithm is termed BPRSLIM. Figure 3 shows the BPRSLIM integrated with item features (**IF**), here, SLIM uses a variant of Bayesian Personalized Ranking (BPR) to calculate the coefficient matrix **C** instead of using the least squares learning method. The BPR is made up of optimization criterion (the BPT-opt) which optimizes the model parameters and a gradient based-learning technique for personalized item recommendation. Item features were introduced into the BPRSLIM model. This permitted both user-rating matrix **RM** and the item feature matrix **IF** to be replicated using similar sparse linear aggregation coefficient matrix.

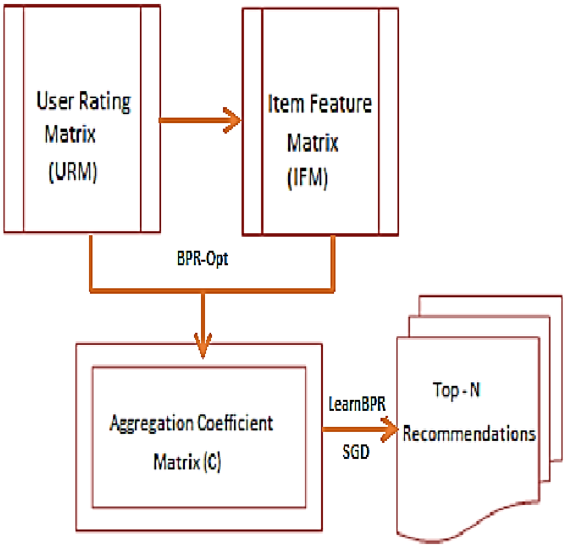


Figure 3. The BPRSLIM +IF

In other words, the coefficient matrix **C** should satisfy the Eq. 3 as well as Eq. 4

$$RM \sim RMC \qquad (3)$$

and

$$IF \sim IFC \qquad (4)$$

This is realized by learning the sparse $n \times n$ coefficient matrix **C** in Eq. (3 and 4) as the minimizer of the regularized optimization problem expressed as in Eq. 5.

$$\underset{C}{\text{minimize}} = \frac{1}{2} \left\| RM - RMC \right\| \frac{2}{F} + \frac{\alpha}{2} \left\| IF - IFC \right\| \frac{2}{F} + \frac{\beta}{2} \left\| C \right\| \frac{2}{F} + \lambda \left\| C \right\| 1 \qquad (5)$$
$$subject\ to\quad C \geq 0,\ \ diag(C) = 0),$$

where $\left\| IF - IFC \right\| \frac{2}{F}$ evaluate the level of fitness of **C** with the item attributes information. α is a regularization parameter used to adjust the relative importance of the user-item rating, **RM** and the item attributes information **IF** whenever they are utilized in learning the coefficient matrix **C.** To generate recommendation for user U_i on item I_j, then, $\widetilde{RM}_{ij} = RMm_i^T c_j$. **C** is learned from both **RM** and **IF** concurrently by employing **IF** to regularize the original BPRSLIM model. The constraint $diag(C) = 0)$ ensures that rm_{ij} is not used to compute itself. Without such constraint, an item may recommend itself. Also, the constraint $C \geq 0$ is put in place so that the learned C matches the positive aggregation over items. The algorithm for the BPRSLIM injected with Item Information is depicted in Table 1.

Table 1. Algorithm for BPRSLIM with item attribute information

BPRSLIM with Item Features
1: Input: **RM, IF, α, β, λ** 2: Generate a coefficient Matrix **C** from **RM, IF** in (1) using **BPT-Opt** criterion 3: Learn **C** as a minimizer of the optimization problem in (equation 4) using **LearnBPR** (SGD) 4: Generate recommendations of top-N ranked items to user from **C** Stop

In order to confirm if the proposed approach performs better than the existing approaches, we compare the proposed and existing approaches together as described in the following section.

4. EXPERIMENTS

4.1. Data Description

The recommender model was tried on MovieLens 1 M rating data, it comprises of 1,000,209 ratings of about 3,900 movies, generated by 6,040 MovieLens users. It is publicly available at https://grouplens.org/datasets/movielens/1m/, there are about 165 ratings per user and 256 ratings per movie on average on a 5-star rating scale. Table 2 describes the data used in the experiments. Also, within the datasets, every user has at least 20 ratings. A unique id identified every user and movie. The sparsity level of the rating matrix is 95.53% [39]. The MovieLens data does not contain any content attributes, therefore, the supplementary information was extracted from an external source. The item feature data frame consists of 18 variables (item features) and 6040 observations (users). The dataset was normalized to achieve standardization.

Table 2: Statistics of MovieLens (1M) experimental datasets

Movie_dataset_info	Analysis_of_the_dataset
Date_Range	2000-2003
Rating_Scale	1-5 stars scale
No_of_Users	6,040
No_of_Movie	3,900
Total_Ratings	1,000,209
Tag_info	0
Sparsity_level	95.53%
Density level	4.47%

The dataset was randomly split into two parts, 80% to 20% split ratio. The model was trained with 80% while the remaining 20% split was utilized in evaluating the recommendation accuracy of the model.

4.2. Experimental Settings

1) Baselines: The proposed model (BPRSLIM+ IF) was compared with other models specified below in order to verify its effectiveness.

- BPRSLIM: It is a SLIM that uses Bayesian Personalized Ranking (BPR) to learn the item-to-item associations within the user-rating matrix to generate recommendations.
- ItemKNN (Item k-nearest neighbors): It does not learn parameters, but form neighborhoods based on item similarities to generate recommendations.

2) Evaluation Metrics

Since users are always interested in a few top-ranked recommended items, top-n evaluation measures were utilized to examine the performance accuracy of the model. These measures include precision and nDCG. They are described as follow:

- The precision measure gives the number of movies recommended that are relevant to user. It is defined as Eq. (6):

$$Precision\ (PR) = \frac{Correctly\ recommended\ movies}{Total\ recommended\ movies} \qquad 6$$

- nGCG estimates the capacity of the model to rank the movies recommended to user in correct order in which the user wants it. It is defined as Eq. (7):

$$nDCG_r \ = \frac{DCG_r}{IDCG_r} \qquad 7$$

3) Top-N recommendation experiments were performed using MyMediaLite application programming interface (API) version 3.11, an open-source software that contains library of recommendation system algorithms**.** In order to evaluate the performance of the proposed approach, Pre@k and nDCG@k were used respectively with varying number of recommendations, that is, different thresholds.

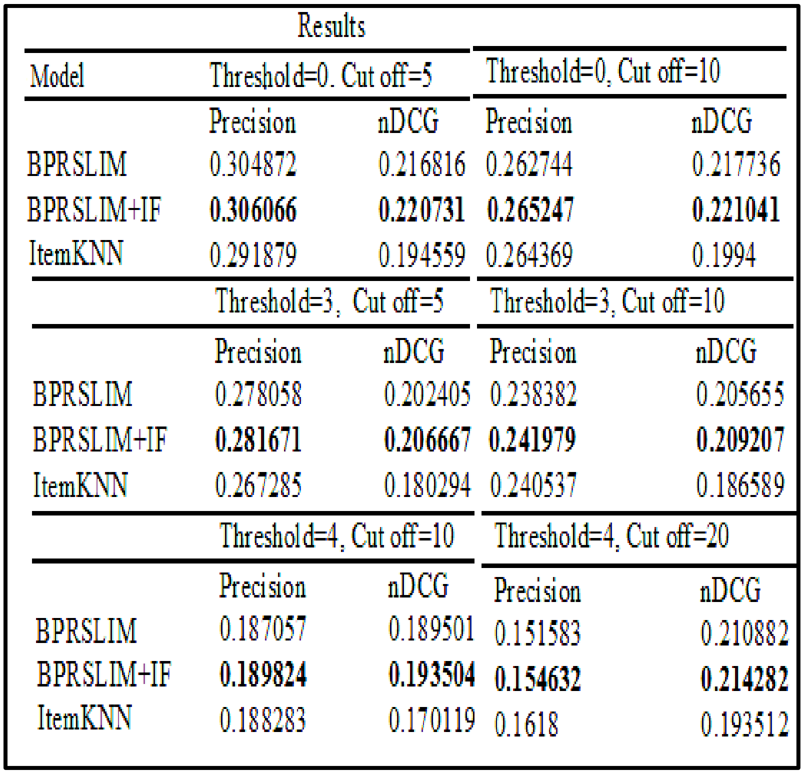
4.3. Results and Discussion

The results of the evaluation metrics at different thresholds (**T**) and cutoffs (**CF**) are shown in Table 3. The different points where BPRSLIM+ IF outperform the BPRSLIM and the ItemKNN algorithms are indicated in bold font in the table under different thresholds and cut-offs. As seen in the table, for different threshold and cut-off, BPRSLIM+IF outperform BPRSLIM and ItemKNN in precision and nDCG. For example, at threshold 0 and cut-off 5, precision and nDCG were 30.6% and 22.1% respectively, at threshold 0 and cut-off 10, precision and nDCG were 26.5% and 22.1% respectively. At threshold 3 and cut-off 5, precision and nDCG were 28.2% and 20.7% respectively, at threshold 3 and cut-off 10, precision and nDCG were 24.2% and 20.9% and finally, at threshold 4 and cut-off 10, precision and nDCG were 19.0% and 19.4% and at threshold 4 and cut-off 20, precision and nDCG were 15.5% and 21.4%.

The implication of these results is that the performance of BPRSLIM improves when it learned from both user-rating and item-feature matrices. It also showed the ability of BPRSLIM +IF to recommend items that are relevant to users in correct order in which the user would prefer it better than BPRSLIM only.

In Figure 4, it shows clearly that at threshold 0 and cut-off 5, BPRSLIM+ IF achieved the best performance with precision and nDCG at 30.6% and 22.1% respectively. It establishes the fact that at this specific threshold and cut-off, the BPRSLIM+IF can recommend relevant items that are interesting to users in the order in which they want it.

Table 3: Comparison of the Performance of Top-N Recommendation Algorithms over BPRSLIM+IF



Results

Model	Threshold=0. Cut off=5		Threshold=0, Cut off=10	
	Precision	nDCG	Precision	nDCG
BPRSLIM	0.304872	0.216816	0.262744	0.217736
BPRSLIM+IF	**0.306066**	**0.220731**	**0.265247**	**0.221041**
ItemKNN	0.291879	0.194559	0.264369	0.1994
	Threshold=3, Cut off=5		Threshold=3, Cut off=10	
	Precision	nDCG	Precision	nDCG
BPRSLIM	0.278058	0.202405	0.238382	0.205655
BPRSLIM+IF	**0.281671**	**0.206667**	**0.241979**	**0.209207**
ItemKNN	0.267285	0.180294	0.240537	0.186589
	Threshold=4, Cut off=10		Threshold=4, Cut off=20	
	Precision	nDCG	Precision	nDCG
BPRSLIM	0.187057	0.189501	0.151583	0.210882
BPRSLIM+IF	**0.189824**	**0.193504**	**0.154632**	**0.214282**
ItemKNN	0.188283	0.170119	0.1618	0.193512

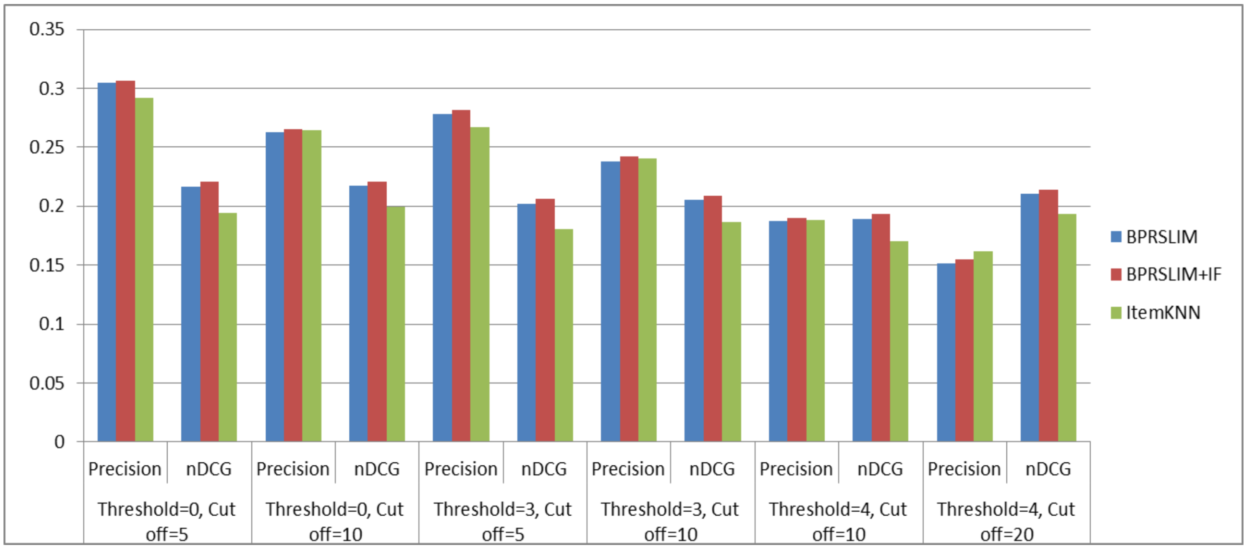


Figure 4. Comparison results of the three models with respect to different thresholds and cut off s

5. CONCLUSION

CF techniques depend on the quantity of available information. BPRSLIM is a variant of model-based top-N collaborative filtering technique. Although, it generates recommendations fast but usually the quantity of rating in its user-item matrix is always very scanty as against the quantity of ratings that are needed to generate reliable recommendations. This is because users do not always have interest to rate enough items. This always leads to very poor and unreliable recommendations of items to users. This work integrates item features into Bayesian Personalized Ranking Smart Linear Model (BPRSLIM) in order to improve its recommendation capability. The performance of the model showed significant improvement in recommendation accuracy when compared with other top-N item-based collaborative filtering recommendation algorithms over evaluation metrics such as Precision and nDCG with the best performance at 30.6% and 22.1% respectively. Therefore, the model can be integrated with existing collaborative filtering tools to alleviate sparseness problems and hence provide reliable recommendations of items to users in different domains.

REFERENCES

1. F. O. Isinkaye, Y. O Folajimi, A.B. Adeyemo, "On collaborative filtering model optimised with multi-item attribute information space for enhanced recommendation accuracy", Int J Intell Syst Tech. Appl vol.19, no.3, pp. 207-215, 2020.
2. H. Zitouni, S. Meshoul, C. Mezioud, "New contextual collaborative filtering system with application to personalized healthy nutrition education", J. King Saud Univ., Comp. Inf. Sci https://doi.org/10.1016/j.jksuci.2020.04.012, 2020.
3. M. Nilashi, O. Ibrahim, K. Bagherifard, "Recommender system based on collaborative filtering using ontology and dimensionality reduction techniques", Expert Syst. Appl vol.92, pp.507-520, 2018.
4. M. Si, Q. Li, "Shilling attacks against collaborative recommender systems: a review", Artif Intell Rev vol.53, no.1, pp. 291-319, 2020.
5. F. O. Isinkaye, Y.O. Folajimi B. A. Ojokoh, "Recommendation systems: Principles, methods and evaluation", Egypt Inform J vol.16, no. 3, pp.261-273, 2015.
6. Y. Chen, C. Wu, M. Xie, X. Guo, "solving the sparsity problem in recommender systems using association retrieval", J. Comput vol.6, no. 9, pp. 1896-1902, 2011.
7. Y. Wang, C. Feng, C. Guo, Y. Chu, J. N. Hwang, "Solving the sparsity problem in recommendations via cross-domain item embedding based on co-clustering", In: Proceedings of the twelfth ACM international conference on web search and data mining (pp. 717-725), 2019.
8. S. Kabbur, X. Ning, G. Karypis, "Fism: factored item similarity models for top-n recommender systems", In: Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 659-667), 2013.
9. V. C. Ostuni, T. Di Noia, E. Di Sciascio, R. Mirizzi, "Top-n recommendations from implicit feedback leveraging linked open data", In: Proceedings of the 7th ACM conference on recommender systems (pp. 85-92), 2013.
10. J. Wei, J. He, K. Chen, Y. Zhou, Z. Tang, "Collaborative filtering and deep learning-based recommendation system for cold start items", Expert Syst Appl vol. 69, pp.29-39, 2017
11. Y. Jian Cheng, L. Yang, J. Li, H. Yan, X. Wang, "A trust-based collaborative filtering algorithm for E-commerce recommendation system", J Amb Intel Hum Comp vol. 10, no. 8, pp.3023-3034, 2019.

12. M. Duma, B. Twala, “Sparseness reduction in collaborative filtering using a nearest neighbour artificial immune system with genetic algorithms”, Expert Syst. Appl vol. 132, pp. 110-125, 2019.
13. S. Kant, T. Mahara, “Merging user and item based collaborative filtering to alleviate data sparsity”, Int. J. Syst. Assur. Eng. Manag vol. 9, no. 1, pp. 173-179, 2018.
14. S. Bag, S. Kumar, A. Awasthi, M. K. Tiwari, “A noise correction-based approach to support a recommender system in a highly sparse rating environment”, Decis. Support Syst vol.118, pp. 46-57, 2019.
15. X. Yang, Y. Guo, Y. Liu, H. Steck, “A survey of collaborative filtering based social recommender systems”, Comput. Commun vol. 41, pp. 1-10, 2014.
16. Y. Xu, L. Wai, L. Tianyi, "Collaborative filtering incorporating review text and co-clusters of hidden user communities and item groups", In: Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, (pp. 251-260), 2014.
17. S. Huang, J. Ma, P. Cheng, S.Wang, “A hybrid multigroup coclustering recommendation framework based on information fusion”, Acm T Intel Syst Tec (TIST), 6(2):1-22, 2015.
18. H. You, H. Li, Y. Wang, Q. Zhao, “An improved collaborative filtering recommendation algorithm combining item clustering and slope one scheme”, In: Proceedings of the international multiconference of engineers and computer scientists, (pp. 313-316), 2015.
19. D. Bokde, S. Girase, D. Mukhopadhyay, “Matrix factorization model in collaborative filtering algorithms: A survey. Procedia Comput. Sci vol. 49, pp. 136-146, 2015.
20. D. C. Anastasiu, E. Christakopoulou, S. Smith, M. Sharma, G. Karypis, “Big data and recommender systems”, Novtica: J Spanish Comp Sci Associat vol. 240, pp. 1-28, 2016.
21. A. Saluja, M. Pakdaman, D. Piao, A. P. Parikh, “Infinite mixed membership matrix factorization”, In: 2013 IEEE 13th international conference on data mining workshops (pp. 800-807), 2013
22. S. T. Tu, L. J. Zhu, “A bandit method using probabilistic matrix factorization in recommendation”, J. Shanghai Jiaotong University (Sci.) vol. 20, no. 5, pp. 535-539, 2015.
23. Y. Liu, D. Liu, H. Xie, L. Wang, “A research on the improved slope one algorithm for collaborative filtering”, Int. J. Comput. Sci. Math, vol. 7, no. 3, pp. 245-253, 2016.
24. A. Godoy-Lorite, R. Guimerà, C. Moore, M. Sales-Pardo, “Accurate and scalable social recommendation using mixed-membership stochastic block models”, Proceedings of the Natl. Acad. Sci vol. 113, no. 50, pp. 14207-14212, 2016.
25. T. Tang, P. Harrington, “Scaling matrix factorization for recommendation with randomness”, In: Proceedings of the 22nd international conference on world wide web (pp. 39-40), 2013.
26. X. Deng, F. Zhuang, Z. Zhu, “Neural variational collaborative filtering with side information for top-K recommendation”, Int J Mach Learn Cyb vol. 10, no. 11, pp. 3273-3284, 2019.
27. T. Hao, Z. Zheng, “The implementation and optimization of matrix decomposition based collaborative filtering task on x86 platform”, In: International Symposium on Benchmarking, Measuring and Optimization (pp. 110-115), 2019.
28. Y. Zheng, B. Mobasher, R. Burke, “Deviation-based contextual SLIM recommenders”. In: Proceedings of the 23rd ACM international conference on conference on information and knowledge management (pp. 271-280), 2014a
29. Y. Zheng, B. Mobasher, R. Burke, “Integrating context similarity with sparse linear recommendation model”, In: international conference on user modeling, adaptation, and personalization (pp. 370-376), 2015.

30. Y. Zheng, B. Mobasher, R. Burke, R (2014b) CSLIM: Contextual SLIM recommendation algorithms", In: Proceedings of the 8th acm conference on recommender systems (pp. 301-304), 2014b.
31. Z. Fan, X. Ning, "Local sparse linear model ensemble for top-n recommendation", Retrieved 2/8/2020 at: https://doogkong.github.io/2017/papers/paper10.pdf, 2017.
32. E. Christakopoulou, G. Karypis, "Local item-item models for top-n recommendation", In: Proceedings of the 10th ACM Conference on Recommender Systems (pp. 67-74), 2016.
33. T. Liu, Z. Wang, J. Tang, S. Yang, G. Y. Huang, Z. Liu, "recommender systems with heterogeneous side information", In: The World Wide Web Conference (pp. 3027-3033), 2019.
34. W. Deng, P. Wang, J. Wang, C. Li, M. Guo, "Psl: Exploiting parallelism, sparsity and locality to accelerate matrix factorization on x86 platforms", In: International symposium on benchmarking, measuring and optimization (Bench 2019).
35. T. R Mahesh, V. Vinoth Kumar, L. Se-Jung, "UsCoTc: Improved Collaborative Filtering (CFL) recommendation methodology using user confidence, time context with impact factors for performance enhancement." *Plos one* vo. 18, no. 3, 2023: e0282904.
36. E. Christakopoulou, G. Karypis, "HOSLIM: Higher-order sparse linear method for top-n recommender systems", In: Pacific-Asia conference on knowledge discovery and data mining (pp. 38-49), 2014.
37. S. Zhang, L. Yao, A. Sun, S. Wang, G. Long, M. Dong, "Neurec: On nonlinear transformation for personalized ranking", In: Proceedings *of the* twenty-seventh international joint conference on artificial intelligence (pp. 3669-3675), 2018
38. X. Ning, G. Karypis, "Slim: Sparse linear methods for top-n recommender systems", In: 11th IEEE International Conference on Data Mining (pp. 497-506). 2011.
39. G. Li, Q. Chen, "Exploiting explicit and implicit feedback for personalized ranking. Math. Probl. Eng pp. 1-11, http://dx.doi.org/10.1155/2016/2535329, 2016.