

# A Movie Recommendation using Common Genre Relation on User-Item Subgroup

G. Suganeshwari, S.P Syed Ibrahim



**Abstract:** Movie recommendation system has played a vital role in retrieving the movies that are of interest to the user. Most of the traditional methods provide a unified recommendation without considering the individual preference of the user. To address this challenge, various recommender methods are currently employing side information like location, time, gender, and genre to provide a personalized recommendation. In this paper, we propose —Common Genre Relations (COGS), which incorporates the information on genre relationships between the movies. Meanwhile, the method reduces the search space for each user and helps to mitigate the sparsity problem. To improve the scalability, the methods are executed on user-item subgroups. Extensive experiments are conducted on a real-world dataset. The empirical analysis shows that the proposed method based on the graph model excels the accuracy at top-k than the state-of-art collaborative filtering methods.

**Keywords:** Collaborative filtering, data sparsity, genre relation, movie recommendation system, random walk.

## I. INTRODUCTION

With the increase of information on the web nowadays, a tremendous volume of data is being captured in digital form. However, this massive volume of data leads to information overload problem, especially in movie recommender systems, where users share their interest with friends. Recommendation Systems (RS) has been utilized as a core service in e-commerce applications to alleviate the information overload problem by automatically identifying the items that are of interest to the users. These systems discover the relevant information from the user's previous purchase history and provide a personalized recommendation.

Collaborative methods (CF) is the most successful RS widely adopted in the e-commerce industry to provide personalized recommendation due to its ease of implementation and justifiability [1], [2]. The CF methods recommend items based on similar user's ratings. They have been instrumental in the application where it is tough to analyze the content such as tourism [3], music [4], and movies [5]. The CF methods are generally built on the matrix with either explicit or implicit inputs [6]. The CF methods

suffer from data sparsity and scalability problems [7]. The sparsity problem occurs as a user tends to rate very few items from widely available options. Two users might not show any similarity in sparse data. The scalability problem arises due to the volumizing increase of users and items.

Some machine learning techniques are utilized to overcome the shortcomings of the traditional CF-based approaches like classification [8], clustering [7], [9], [10], and matrix decomposition [11], [12]. In the big data era, data has been growing explosively, and plenty of additional sources are available in addition to the  $R^{mn}$  matrix. In such cases, graph-based CF methods [13], [14] can be used to enhance the top-k recommendation by incorporating the rich side information associated with the users and the items.

The graph-based CF approach recommends items to a user based on the node-node proximity in a user-item bipartite graph. There is a weighted or unweighted edge between the user node and the item node which the user has rated. This approach has been extensively studied as link prediction in social networks [13]. Tong et al. [15] proposed a fast, space-efficient approximation method of the original RWR method. RWR was used to improve the similarity between items in item-based CF [16]. One of the pioneering works using Random-walk with restart (RWR) was proposed by Kontas et al. [14] to mine relation in a multipartite graph by incorporating user, item, and tag information.

In a movie recommender system, the genre information is more reliable than the rating data and provides a better recommendation. The genre combinations in movies give more insight to the preference of the user. It makes more sense to recommend movies with genre combination of children and comedy than children and crime. Although the graph-based methods are successful in improving the recommendation quality and addressing the sparsity problem, it suffers from time complexity.

To overcome these problems, we propose an improved graph-based CF method —COGS by exploiting the genre relation. The main contributions of the COGS are:

- An improved graph-based method is proposed by incorporating the user's historical preferences and common genre relation.
- The proposed method is compared with other CF-based state-of-art methods.

The remaining section is organized as follows: The related work is discussed in Section II. In Section III, we formulate the COGS method, and in Section IV, we describe the proposed method in detail. Section V reports on the experimental results followed by the conclusion in Section VI.

**Manuscript published on 30 December 2019.**

\* Correspondence Author(s)

**G. Suganeshwari** \*, School of Computing Science and Engineering, Vellore Institute of Technology Chennai, Chennai, TamilNadu, India. Email: g.suganeshwari2015@vit.ac.in

**S. P. Syed Ibrahim**, School of Computing Science and Engineering, Vellore Institute of Technology Chennai, Chennai, TamilNadu, India. Email: syedibrahim.sp@vit.ac.in

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an [open access](#) article under the CC-BY-NC-ND license <http://creativecommons.org/licenses/by-nc-nd/4.0/>

**Table- I: Table of Symbols.**

Symbol	Definition
U	Set of users
I	Set of items
R <sup>mxn</sup>	Utility matrix
u	User
i	Item
r <sub>ij</sub>	User i rating on item j
W	Co-clusters
K	The dimension of the user and item latent factors
P	User latent factor matrix
Q	Item latent factor matrix
λ	Regularization parameter
S <sub>ij</sub>	The similarity between item i and j
G(V, E)	User-item bipartite graph
V	Set of nodes V = U ∪ M
E	Set of unweighted edges (u, m, r <sub>ui</sub> ) in G
α	Restart probability in RWR

## II. BACKGROUND AND RELATED WORK

The collaborative Recommendation system can be classified as memory and model-based. This section introduces the preliminaries on memory-based and model-based collaborative filtering methods. Table I represents the symbol and their representation used in this paper.

### A. Collaborative Filtering

Collaborative Filtering (CF) is the classical recommendation algorithm which has been successfully implemented in the commercial sites. The CF-based methods work on the preference of the other like-minded users. The CF method is further classified into the neighborhood (memory) and model-based approaches. The neighborhood-based method consists of two phases: 1. Neighbor selection, and 2. Rating aggregation. In neighbor selection, the similarity is computed based on the item and user in item-based CF [17] and user-based CF [6], respectively. The prediction task estimates how much an active user will rate an unrated item. It is formulated as:

$$P_{a,j} = r'_a + \sum_{u_x \in N(U_a)} s_{a,x} r_{x,i} \quad (1)$$

Here  $r'_a$  is the mean of the active user rating,  $N(U_a)$  is the set of neighbors for  $u_a$ ,  $s_{a,x}$  is the similarity score between the active user and neighbor user,  $r_{x,i}$  is the rating of the neighbor user for the item. The nearest neighbor computation grows exponentially with the increase in users and movies, and the similarity is also affected by sparsity.

In contrast to the memory-based approach, model-based methods utilize machine learning techniques to find patterns or hidden features. The model-based approaches were used to overcome the shortcomings of the memory-based approaches. A model is built based on algorithms like

classification [8], clustering [7], [9], [10], and matrix decomposition [11], [12], and later exploited to predict or recommend the items to the user. In model-based methods, the matrix factorization (MF) method is widely adopted in RS. They predict the unknown ratings by decomposing the high sparse matrix to low-rank representation. The MF methods gained importance after the Netflix price challenge [31], and much more research has been done in MF techniques like SVD [18], SVD++ [12], PMF [19], NMF [20] and ALS [21].

When the number of users and items grow exponentially the classical MF techniques suffer from scalability problem. Clustering has been widely adopted in RS to handle small subgroups of data, instead of the entire utility matrix.

### B. User-Item Subgroups and Collaborative Filtering

Clustering is an unsupervised learning technique commonly used in CF based recommendations to improve the scalability. It groups similar objects into clusters such that inter-class relations are strong, and intraclass relationships are dissimilar. K-Means is the most commonly researched method in recommender systems. Sarwar et al. [7] proposed an RS to cluster the users based on their similarities. Gong [22] proposed a similar approach to cluster the items. All these models are one-sided clustering. In recent years, co-clustering has gained importance based on similarities between pairwise interactions. It refers to simultaneously clustering of rows and columns to obtain homogeneous blocks. This idea was adopted in [23]. Bo et al. [24] proposed a co-clustering approach by block value decomposition based on non-negative matrix factorization. A latent Dirichlet co-clustering method [25] was proposed for document clustering in which each segment was associated with one document topic. Co-clustering is extremely useful when the dataset is very sparse, and pairwise interaction exists. The co-clustering methods have the following advantages,

- Compact representation
- Using duality of clustering
- Reducing the running time

In this paper, we pay our attention to model the co-clustering-based approach to identify the clusters based on the hidden concepts using MF. We adopt an alternating least square MF method.

### C. CF Methods and Graph-based Approaches

The graph-based CF methods gained importance with the plenty of availability of rich side information that can be utilized to enhance the accuracy of the recommendation. The random walk with restart (RWR) is the most commonly used graph-based method for CF [13], [15]. Researchers have employed different variations of random walks in RS to enhance performance. The item-based CF was improved by employing random walk (RW) by computing the item-item similarity matrix [16], [26]. Recent methods have adopted RWR by extending a bipartite network with other heterogeneous information.



Published By:

Blue Eyes Intelligence Engineering

and Sciences Publication (BEIESP)

© Copyright: All rights reserved.

TrustWalker method [27] was proposed by combining the trust-based and IBCF method. This method was computationally overhead as a rating for similar items was also considered along with the target item, and this increased the length of the random walk graph.

Similarly, Clements et al. [28] used the RWR method to combine user tags and ratings. Bu et al. [29] combined user social relationships and content features for music recommendation. This method requires learning a ranking function, and it demands a much ample memory space. Different types of nodes were added in a multi-layer structure in RW to generate a context-aware recommendation. The critical issue in generating recommendations using the RWR approach is time complexity. Although most of the existing methods improve the recommendation quality user's individual preferences are not considered. Based on the above observation, we propose a graph-based CF method by incorporating the common genre relation on the subgroups of users and movies.

### III. COMMON GENRE RELATION

Fig 1. describes the design of the proposed work in detail. It consists of two main components: user-item subgroups and graph-based movie ranking. In CARE [32], the authors exploit the citation information in article recommendations to provide a personalized recommendation to the researchers. Our work is an extension where we fit the genre information in the movie recommendation. The movies are related if the users have a common interest. Although this method may improve the RS performance by exploiting the more reliable genre information, it suffers from scalability problems. To address this, we first frame user-item subgroups and later utilize genre combinations in each user-item subgroup.

#### A. Problem Formulation

Let us assume set of users  $U = \{u_1, u_2, \dots, u_m\}$  and set of movies  $M = \{m_1, m_2, \dots, m_n\}$ . The ratings by  $m$  users and  $n$  items are represented in a utility matrix  $R^{mxn}$ . Let  $r_{ij}$  denote the rating information given by  $u_i$  user to  $m_j$  movie. The goal is to provide a personalized recommendation by exploiting the genre information associated with the movies on

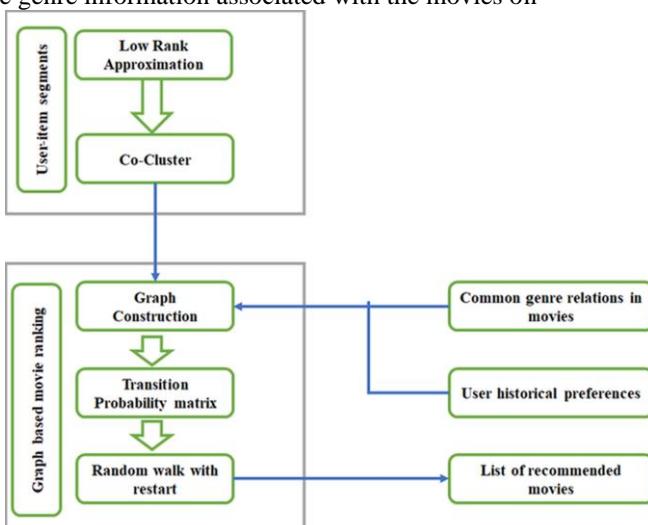


Fig. 1. Design of COGS.

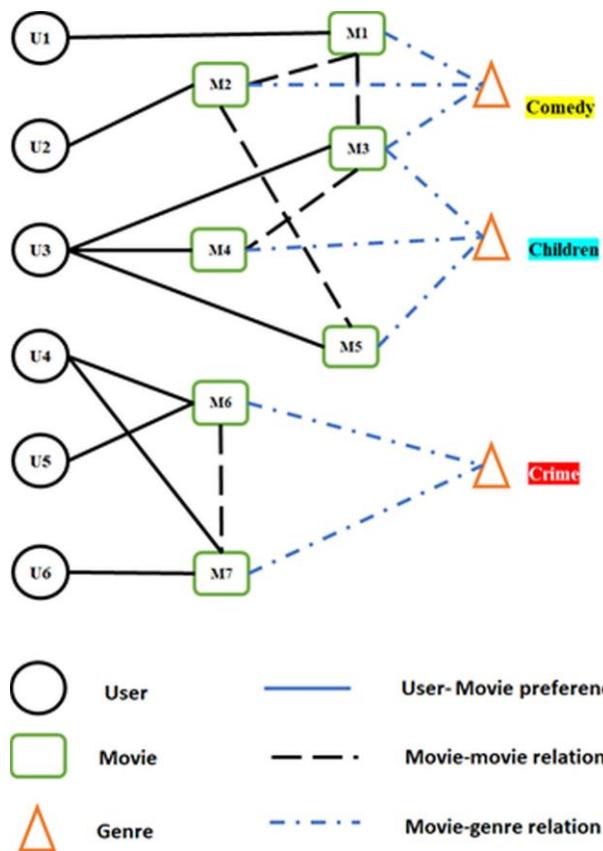


Fig. 2.Example movie ranking using graph.

subgroups of users and movies, as the genre information is more reliable compared to the ratings.

The recommendation scenario of the COGS method consists of three objects users, movies, and genres, as given in Fig. 2. The first link is the traditional CF method, where each user is linked to the movie, which has been rated by the user. The second link represents the common genre relation between the movie. In a movie recommender system, the movies are often associated with the genre information.

This additional information may improve the quality of the recommendation system as users prefer specific genres over other genres. As a result, it is essential to include the genre relation in media recommender systems like movies and music.

#### B. User – Item Subgroup

Here  $R \in R^{mxn}$ ,  $G \in R^{pk}$ , and  $H^{kn}$ . The rank of  $R$  is much less than the rank of  $G$  and  $H$ . The user-movie subgroup consists of 2 steps:

- 1) Low-Rank Approximation
- 2) user-item subgroups

#### Low-rank Approximation

The MF's cost function is represented as

$$X = |R - G \times H^T|_2 + \lambda(|G|_2 + |H|_2) \quad (3)$$



The cost function in (3) is non-convex, as both  $G$  and  $H$  are unknown variables. The problem can be reduced to simple linear function by fixing  $G$  and optimizing  $H$  and vice-versa.

$$g_u = (H^T H + \lambda I)^{-1} H^T r_u \quad (4)$$

$$h_i = (G^T G + \lambda I)^{-1} G^T r_i \quad (5)$$

Here  $I$  is the unit matrix,  $r_u$ ,  $r_i$  is the  $u^{th}$  row and  $i^{th}$  column of the matrix  $R$ . The solution given by ALS is unique and the model is constructed until convergence. ALS is adopted in COGS since it has the advantage of parallel implementation with the ability to attain accurate recommendations in a sparse dataset.

### Co-clustering

The selected rows and columns are segmented first. Based on this initialization of the first row and column, the rest of the rows and columns are clustered.

### Steps for Co-Clustering

1. The input data matrix  $R$  is decomposed into  $U^{mxk}$  and  $M^{nxk}$
2. Randomly initialize row and column cluster  $r_0, c_0$
3. Single side row clustering based on  $r_0, c_0$
4. Single side column clustering based on  $c_0, r_0$
5. Repeat 4 and 5 until convergence.
6. Return user-item subgroups.

## C. Graph-Based Ranking

### Graph Construction

As an inspiration from [32], we extend our work with the  $R^{mxn}$  matrix to co-cluster the user and movies and then incorporate pairwise genre relation to producing a top-k recommendation. If a user expresses interest in the movie, then it is represented as  $P_{UM} = P_{UM}^{mxn}$  with  $P_{UM}(i, j) = P_{UM}(j, i)$ , as shown in (6). The  $P_{UU}$  matrix represents the user-user relation and is expressed as 0 since the user-user relationship is dynamic.

$$P_{UM}(i, j) = \begin{cases} 1 & \text{if } u_i \text{ has rated movie } m_j \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

The relationship between movies and genres is converted to common genre relations between movies. The pairwise relationships between the movies are denoted as  $P_{MM} = P_{MM}^{m \times n}$  with  $P_{MM}(i, j)$ , indicating whether there is a common genre relation between movie  $M_i$  and  $M_j$ . We consider undirected graph where  $P_{MM}(i, j) = P_{MM}(j, i)$  and is denoted as given in (7).

$$P_{MM}(i, j) = \begin{cases} 1 & \text{if there is a common genre relation} \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

Based on these two matrices, graph  $G$  is constructed. Let  $G = (V, E)$  represent the graph with  $V = (U, M)$ , where  $U$  is the user node, and  $M$  is the movie node. If a user has rated a movie, then the preference is represented as edge and  $P_{UM}(i, j) = 1$  or  $P_{UM}(j, i) = 1$ . If there is common genre relation between two movies, then there is edge with  $P_{MM}(i, j) = 1$  or

$$P_{MM}(j, i) = 1.$$

### Transition Probability Matrix

The transition probability matrix is build based on the three matrices  $P_{MM}$ ,  $P_{UM}$ , and  $P_{UU}$ . The transition probability of moving from user vertex to another user vertex is 0.

$$T_{UU} = 0 \quad (8)$$

The transition probability of moving to a movie vertex is

$$T_{UM}(i, j) = \frac{P_{UM}(i, j)}{\sum_k P_{UM}(i, k)} \quad (9)$$

The transition probability of moving to a user vertex is

$$T_{MU}(i, j) = \frac{P_{UM}(j, i)}{\sum_{k1} P_{MU}(i, k1) \sum_{k2} P_{MM}(i, k2)} \quad (10)$$

The transition probability of moving to another movie vertex is

$$T_{MM}(i, j) = \frac{P_{MM}(i, j)}{\sum_{k1} P_{MU}(i, k1) \sum_{k2} P_{MM}(i, k2)} \quad (11)$$

### ALGORITHM 1: GRAPH-BASED MOVIE RANKING

**Input:** An active user  $u_a \in U$

Random walk probability,  $\alpha$ ;

Transition Probability matrix,  $T$ ;

Maximum number of iterations,  $\text{maxSteps}$ ;

**Output:**

The ranking score for all movies  $\text{movieRank}\{m_1, \dots, m_k\}$

**1: Generate Co-clusters**

**2: Initialize** ranking scores for all vertices,  $\text{RateAll}(1 : m+n)$  vertices ( $m$  users and  $n$  movies)

**3: Set Score(v) = 0** // ranking scores for all vertices at initial stage is 0

**4: Score( $u_a$ ) = 1**

**5: for**  $i=0$ ;  $i > \text{maxSteps}$ ;  $i++$  **do**

**6: for each**  $v_x \in V_u \cup V_m$  **do**

**7: for each**  $v_y \in V_u \cup V_m$  **do**

**8: tScore( $v_y$ ) =  $\alpha \times \text{Score}(v_x) \times T(v_x, v_y) + \text{tScore}(v_y)$**

**9: end for**

**10: if**  $v_x == u_a$  **then**

**11: tScore( $v_x$ ) =  $tScore(v_x) + 1 - \alpha$**

**12: end if**

**13: end for**

**14: RateAll = tScore**

**15: end for**

**16: movieRank (1 : m) = RateAll(n+1 : m+n)**

**17: return**  $\text{movieRank (1:m)}$ ;



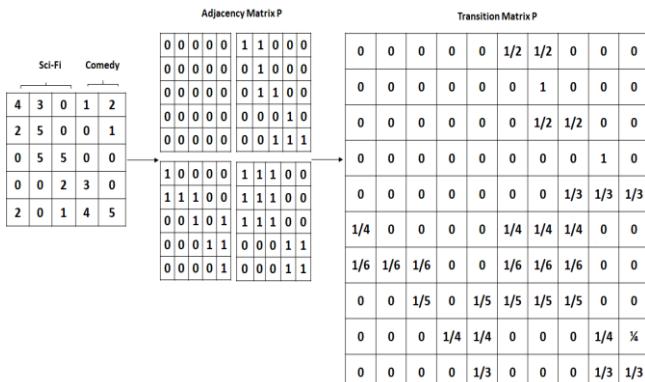


Fig. 3. COGS Example.

The transition probability matrix is denoted as

$$T = \begin{pmatrix} T_{UU} & T_{UM} \\ T_{MU} & T_{MM} \end{pmatrix} \quad (12)$$

#### Random Walk with restart

A random walk is applied to the graph after obtaining the  $T$  matrix to compute movie rankings for each user. The proposed method incorporates common genre relations along with the user's historical preferences. The random walk is initiated from a vertex  $u_x$  (active user) and moves to the next vertex  $x$  with probability  $\alpha$ . It moves to any other vertex  $y$  connected to the  $x$  with the transition probability  $T(x,y)$ . With probability  $(1-\alpha)$  it returns to the source vertex  $u_x$ . The pseudocode of the proposed method is shown in algorithm 1. Fig. 3 shows an example of the proposed method.

## IV. EXPERIMENT ANALYSIS

This section enlists the experiments conducted on the benchmark MovieLens datasets [5], which includes one million ratings from 6040 users on 3900 movies and one lakh ratings from 943 users and 1682 movies. The ratings are represented in numerical form (1~5) along with the rating timestamp. All experiments reported in this section were performed on the machines with Intel(R) Xeon(R)2 CPU 3.36GHz and 32GB RAM and implemented in the Spark framework. Precision @ k and Mean Square Error (MSE) has been used to evaluate the COGS method. Here we choose four CF-based algorithms User-based CF (UBCF), Item-based CF (IBCF), Single + UBCF – we use K-means clustering to cluster users and then implement UBCF, CoC + UBCF – we use UBCF on user-item co-clusters, CoC + COGS – we use common genre relations on user-item co-clusters.

### A. The Mean Average Precision

It is the popular performance measure used for measuring the significance of the top-k items. It is the mean of the average precisions (MAP) [33] at K, as indicated in (13). Higher the MAP values more top the quality of the recommendation system.

$$MAP = \frac{\sum_{q \in Q} AP(q)}{|Q|} \quad (13)$$

### B. Discounted Cumulative Gain

Discounted Cumulative Gain (NDCG) [14] is used to

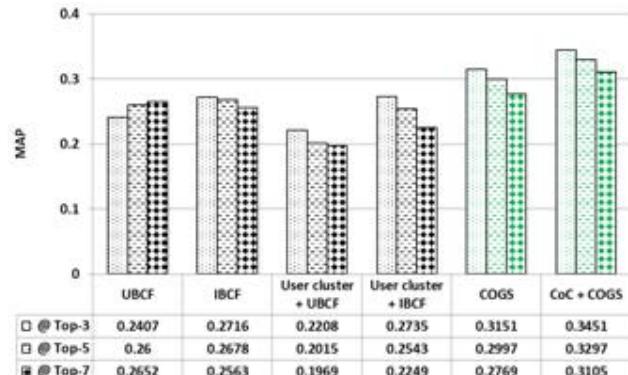


Fig. 4. Comparison of COGS in MAP with 10 sungroups.

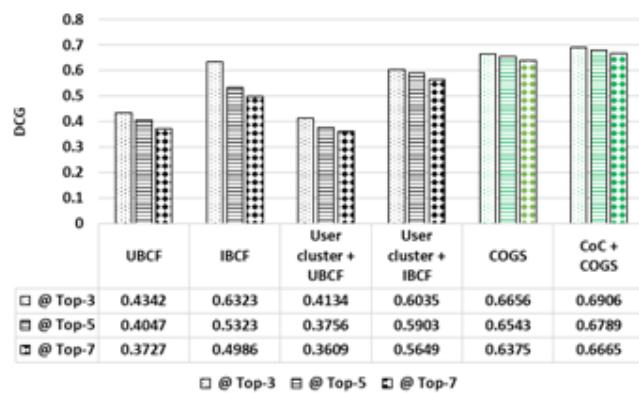


Fig. 5. Comparison of COGS in DCG with 10 subgroups.

evaluate the significance of the recommended top-K items. The NDCG is calculated as:

$$NDCG@T = \sum_{i=1}^T \frac{rel(i)}{\log_2(i+1)} \quad (14)$$

Here,  $rel(i)$  is the grade relevance of the movie,  $t$  is the rank position of the movie in the top-k recommendation list. Fig. 4. shows the comparison of the MAP values of the UBCF, IBCF, COGS, Single + UBCF, CoC + UBCF and CoC + COGS method with 10 user-item groups. As shown in figure COGS, and CoC + COGS method exceeds the other state-of-art methods in MAP and its percentage values are shown in Table II.

Table- II: The Percentage Values that COGS Excels Other Methods in MAP with 10 Sub-groups

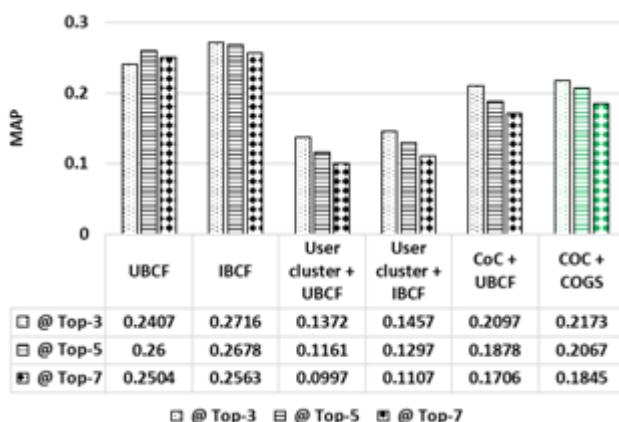
MAP	Top-3	Top-5 %	Top-7 %
COGS/ UBCF	43.37%	26.81%	17.08%
COGS / IBCF	27.06%	23.11%	21.15%
CoC + COGS / Single + UBCF	56.30%	63.62%	57.69%
CoC + COGS / CoC + UBCF	26.18%	29.65%	38.06%

# A Movie Recommendation using Common Genre Relation on User-Item Subgroup

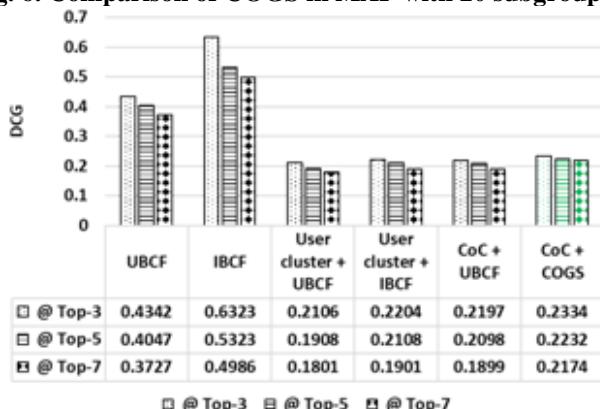
Similarly, the DCG values are compared for the ml-1M dataset with 10 groups in Fig. 5. Our model excels with the other methods at the top-k DCG, and the percentage it surpasses other methods is illustrated in Table III.

**Table- III: The Percentage Values CoC + COGS Excels Other Methods in DCG with 10 Sub-groups**

MAP	Top-3 %	Top-5 %	Top-7 %
COGS/ UBCF	59.05%	67.75%	78.83%
COGS / IBCF	09.22%	27.54%	33.67%
CoC + COGS / Single + UBCF	67.05%	80.75%	84.68%
CoC + COGS / CoC + UBCF	14.43%	15.01%	17.99%



**Fig. 6. Comparison of COGS in MAP with 20 subgroups.**



**Fig. 7. Comparison of COGS in DCG with 20 subgroups.**

The MAP and DCG are compared with 20 user-item groups in Fig. 6 and Fig 7. From the results, we can infer that results with 20 groups do not yield better accuracy compared to the other state-of-art methods. The low quality of efficiency arises because, with more number of clusters, the sparsity increases, leading to incapability to find the neighbors especially for users with few transactions.

## V. CONCLUSION

In this paper, a movie recommendation system is proposed that exploits the common genre relation and historical preference of the user. The user-item subgroups are formed based on the low-rank approximation. Then the information with common genre relation is incorporated into each user-item subgroup to build a graph-based movie recommendation model. The empirical analysis shows that

the proposed COGS method provides an accurate and more scalable recommendation than the other state-of-the-art collaborative methods. The future work can be extended by forming better user or item subgroups by adding additional information based on social information.

## REFERENCES

- G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 6, pp. 734–749, Jun. 2005.
- C. Desrosiers and G. Karypis, "A comprehensive survey of neighborhood-based recommendation methods," in *Recommender Systems Handbook*, F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, Eds. New York: Springer-Verlag, 2011, ch. 4, pp. 107–144.
- Jiang, Shuhui, Xueming Qian, Tao Mei, and Yun Fu. "Personalized travel sequence recommendation on multi-source big social media," *IEEE Trans. Big Data* 2, no. 1, pp. 43-56, 2016.
- Jiajun Bu, Shulong Tan, Chun Chen, Can Wang, Hao Wu, Lijun Zhang, and Xiaofei He. "Music recommendation by unified hypergraph: Combining social media information and music content," In Proc. Int. Conf. on MM. ACM, New York, NY, 391–400, 2010.
- Harper F Maxwell, and Joseph A Konstan. 2016. "The movie lens datasets: History and context,". ACM Tran. interactive intelligent sys. (tiis) 5, no. 4: 19.
- J. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," in Proc. 14th Conf. Uncertainty Artif. Intell., pp. 43–52, 1998.
- B. Sarwar, G. Karypis, and J. Konstan, "Recommender systems for largescale e-commerce: Scalable neighborhood formation using clustering," in Proc. 5th Int. Conf. Comput. Inf. Technol., 2002, pp. 1–6.
- X. Su and T. Khoshgoftaar, "Collaborative filtering for multi-class data using belief nets algorithms," in Proc. IEEE 18th IJCAI, pp. 497–504, Nov. 2006.
- T. Hofmann and J. Puzicha, "Latent class models for collaborative filtering," in Proc. Int. Joint Conf. Artif. Intell., vol. 16, pp. 688–693, 1999.
- D. Lemire and A. Maclachlan, "Slope one predictor for online rating based collaborative filtering," in Proc. SDM, pp. 1–5, 2005.
- D. Billsus and M. Pazzani, "Learning collaborative information filters," in Proc. 15th Int. Conf. Mach. Learn., pp. 46–54, 1998.
- Y. Koren, "Factorization meets the neighborhood: A multifaceted collaborative filtering model," in Proc. SIGKDD, pp. 426–434, 2008.
- David Liben-Nowell and Jon Kleinberg, "The link prediction problem for social networks," in Proc. of the 12th Int. Conf. Inf. Knowledge Management (CIKM'03). ACM, New York, NY, 556–559, 2003.
- Ioannis Konstas, Vassilios Stathopoulos, and Joemon M. Jose, "On social networks and collaborative recommendation," in Proc. 32nd Int. ACM SIGIR Conf. Research and Dev. Inf. Retrieval (SIGIR'09). ACM, New York, NY, 195–202, 2009.
- Kensuke Onuma, Hanghang Tong, and Christos Faloutsos. "TANGENT: A novel, surprise me, recommendation algorithm," in Proc. of the 15th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (KDD'09). ACM, New York, NY, 657–666, 2009.
- Marco Gori and Augusto Pucci, "ItemRank: A random-walk based scoring algorithm for recommender engines," in Proc. of the 20th Int. Joint Conf. on AI. Morgan Kaufmann, San Francisco, CA, 2766–2771, 2007.
- B. Sarwar, G. Karypis, J. Konstan, and J. Reidl, "Item-based collaborative filtering recommendation algorithms," in Proc. 10th Int. Conf. WorldWideWeb, pp. 285–295, 2001.
- Badrul M. Sarwar, George Karypis, Joseph Konstan, and John Riedl, "Application of dimensionality reduction in recommender system-a case study," in Proc. ACM WebKDD Web Mining for E-Commerce Workshop.
- Mnih, A and Salakhutdinov, R, "Probabilistic matrix factorization," in Adv. in neural inf. Processing, pp. 1257-1264, 2008.
- Lee, D. D., and Seung, H. S, "Algorithms for non-negative matrix factorization," in Adv. Neural inf. Processing Systems, pp. 556-562, 2000.



Published By:

Blue Eyes Intelligence Engineering

and Sciences Publication (BEIESP)

© Copyright: All rights reserved.

21. Chen, J., Fang, J., Liu, W., Tang, T., Chen, X., and Yang, C, "Efficient and portable ALS matrix factorization for recommender systems," in IEEE Int. Parallel and Distributed Processing Symposium Workshops (IPDPSW) pp. 409-418, May 2017.
22. Songjie Gong, "An efficient collaborative recommendation algorithm based on item clustering," in Adv. in Wireless Networks and Inf. Systems. Springer, 381–387, 2010.
23. Thomas George and Srujana Merugu, "A scalable collaborative filtering framework based on coclustering," in 5th IEEE Int. Conf. Data Mining. IEEE. 625–628, 2005.
24. Long, B., Zhang, Z. M and Yu, P. S, "Co-clustering by block value decomposition," in Proc. ACM SIGKDD int. conf. Knowledge discovery in data mining, pp. 635-640, August 2005.
25. Shafiei, M. M., and Milios, E. E, "Latent Dirichlet co-clustering," in Sixth Int. Conf. on Data Mining (ICDM'06), pp. 542-551, 2006.
26. Yildirim, H., and Krishnamoorthy, M. S., "A random walk method for alleviating the sparsity problem in collaborative filtering," in Proc. 2008 ACM conf. On Recommender Systems, pp. 131-138, October 2008.
27. M. Jamali and M. Ester, "Trustwalker: a random walk model for combining trust-based and item-based recommendation," in Proc. of the 15th ACM SIGKDD Int. Conf. Knowledge discovery and data mining, pp. 397-406, June 2009.
28. M. Clements, Vries A. P., and Reinders M. J, "The task-dependent effect of tags and ratings on social media access," ACM Trans. Inf. Systems (TOIS), 28(4), 21, 2010.
29. J. Bu, S. Tan, C. Chen, C. Wang, H. Wu, L. Zhang, and X. He, "Music recommendation by unified hypergraph: combining social media information and music content," in Proc. 18th ACM Int. Conf. Multimedia pp. 391-400, October 2010.
30. H. Wang, F. Nie, H. Huang, and F. Makedon, "Fast nonnegative matrix tri-factorization for large-scale data co-clustering," in Twenty-Second Int. Joint Conf. A.I. June 2011.
31. Bell R. M and Koren, Y, "Lessons from the Netflix prize challenge", SiGKDD Explorations, 9(2), 75-79, 2007.
32. F. Xia, H. Liu, I. Lee, and L. Cao, "Scientific article recommendation: Exploiting common author relations and historical preferences," IEEE Trans. Big Data, 2(2), pp. 101-112, 2016.

## AUTHORS PROFILE



**S. P. Syed Ibrahim** received a Ph.D. degree in Computer Science and Engineering from Anna University, India, in 2006. He is currently a Professor at VIT Chennai in the School of Computing Science and Engineering. His research interests include data mining, big data analytics and machine learning. He has published more than 45 research papers in refereed international journals and conferences.



**Suganeshwari** received the M.C.A degree in Computer Applications from Bharathiar University, India, with an industrial experience of 4 years and academic experience of 8 years. She is currently pursuing a Ph.D. degree from the School of Computing Science and Engineering, VIT Chennai, India. Her research interests include machine learning, recommendation systems and dimensionality reduction techniques.