Hybrid Popularity Model for Solving Cold-start Problem in Recommendation System

Noor Ifada, Ummamah, Mochammad Kautsar Sophan,
Informatics Department
University of Trunojoyo Madura
Bangkalan, East Java, Indonesia
noor.ifada@trunojoyo.ac.id, ummamahuum@gmail.com, kautsar@trunojoyo.ac.id

ABSTRACT

This research proposes a new hybrid popularity model for solving the cold-start problem in the recommendation system. A cold-start problem arises when the target user has no rating history in the system. A hybrid popularity model combines the benefit of both the user and item popularities. The item popularity model assumes that a target user is most expected to like the top-rated items. Whereas the user popularity model presumes that a target user is likely to be influenced by the top users who have given a large number of ratings. Naturally, our proposed $HPop$ model is built in three phases: item popularity, user popularity, and hybrid popularity. The ratio of the item and user popularities are controlled by the use of α. We use the Normalized Discounted Cumulative Gain (NDCG), as well as Precision and Recall metrics to evaluate the performance of our model and its counterparts, i.e., $IPop$ and $UPop$. Using a real-world MovieLens dataset, our experiments show that the employment of the user popularity model is always more beneficial than the item popularity model. $HPop$ performs best when $\alpha = 0.9$ and worst when $\alpha = 1$. The NDCG average of increases from $HPop$ to $IPop$ and $UPop$ are respectively 12.22% and 8.02%. The results in terms of Precision-Recall also show a similar trend to those of NDCG. Hence, we conjecture that the performances of $HPop$, $IPop$, and $UPop$ are stable in any evaluation metrics.

KEYWORDS

Cold-start, Hybrid popularity, Item Popularity, Recommendation system, User Popularity

1 Introduction

Cold-start and scalability are common issues in recommendation systems. Cold-start happens when the system has no record or rating history of the target user [1-4]. Therefore, predicting user interest is puzzling since no personalized data can be used as a reference. Meanwhile, the scalability of a recommendation system is challenged when the complexity gradually increases due to the size of data [5, 6].

Popularity based models are more efficient than other models due to its simple concept and lower complexity [7]. Such a model is suitable for solving the cold-start problem since no previous activities of the user are required. Whereas clustering is an efficient option to tackle the scalability problem since it can scale-down the size of data [8-11].

The motivation of our work is to solve the cold-start problem in the recommendation system by implementing a hybrid popularity model. The proposed model is built such that the ratio of item popularity to user popularity can be controlled. We also solve the scalability concern by implementing the clustering techniques, i.e., to scale-down the expensive calculation of item and user popularity scores.

Through a series of experiments using the real-world MovieLens dataset, we show that the employment of the user popularity model is always more beneficial than the item popularity model in our hybrid popularity model. On the other hand, the performance comparison also shows that our proposed popularity model outperforms its counterparts in terms of Normalized Discounted Cumulative Gain (NDCG), Precision, and Recall. This outcome establishes that our proposed popularity model can unravel the cold-start problem.

Our list of contributions in this paper is: (1) propose a new hybrid popularity model for solving the cold-start problem in recommendation system, (2) scale-down the processes in the hybrid model by implementing clustering techniques, (3) conduct series of experiments to demonstrate the outperformance of our proposed model compared to other corresponding models.

The rest of this paper is arranged as follows: Section 2 describes the related works; Section 3 details the proposed hybrid popularity model for generating item recommendation; Section 4 presents the scalability approach; Section 5 shows empirical analysis, and Section 6 closes the paper.

2 Related Work

There are three common categories of popularity models: item popularity [12, 13], user popularity [14], and hybrid popularity [7]. The item popularity model assumes that the top-rated items are most likely to interest the target user. Whereas the user popularity model presumes that the top users are influencing the target user preference. Hence, the hybrid popularity model assumes that the target user is swayed by both the top-rated items and top users, that makes it suitable for solving the cold-start problem. Popularity models can be built by implementing matrix factorization [12], weighted graph [15], or binary symmetric and erasure channels [16]. However, those approaches are more complex than the normalized sum approach [7].

Researchers have been using the clustering technique to solve the scalability issue in the recommendation systems [8-11]. The clustering technique is an approach that groups data, either numerical or categorical, into several numbers of subset data. In terms of numerical data, Fuzzy C-Means (FCM) has shown to be more accurate than other techniques [9, 11, 17]. On the other hand, K-Modes is efficient for clustering categorical data [18, 19].

In this paper, we propose a new hybrid model to combine the benefit as well as controlling the ratio of item and user popularities by implementing the normalized sum approach to solve the cold-start problem in the recommendation system. Unlike existing models that make use of social-based or implicit data [7, 12, 14, 15], our model uses explicit rating data. To scale-down the expensive computation of the popularity scores, we implement clustering techniques such that we do not have to calculate the scores out of all items and users.

3 Hybrid Popularity Model for Generating Item Recommendation

This section details the stages of building the proposed hybrid popularity model and generating the list of item recommendations. Additionally, we also present a toy example to demonstrate the calculation within each stage. Note that this paper denotes U and I as respectively the set of m users and n items. The rating data is modelled as a rating matrix $X \in \mathbb{R}^{m\times n}$, where x_{ui} represents the rating given by user u to item i.

3.1 Hybrid Popularity ($HPop$) Model

Our proposed Hybrid Popularity ($HPop$) model combines the benefit of both the user and item popularities. The model is built in three phases: (1) item popularity, (2) user popularity, and (3) hybrid popularity.

3.1.1 Item Popularity. The item popularity model assumes that target user u is most expected to like the top-rated items. The popularity score of item j is calculated as:

$$IP_j = \sum_{v\in U} x_{vj} \tag{1}$$

where $j \in I$ and $IP \in \mathbb{R}^{1\times n}$. To scale the range of scores, the item popularity score is normalized by the maximum popularity score:

$$\widehat{IP}_i = \frac{IP_i}{max(IP_*)} \tag{2}$$

where $i \in I$ and $\widehat{IP} \in \mathbb{R}^{1\times n}$.

3.1.2 User Popularity. The user popularity model presumes that the target user u is likely to be influenced by the top users who have given a large number of ratings. The popularity score of user v is calculated as:

$$UP_v = \sum_{j\in I} x_{vj} \tag{3}$$

where $v \in U$ and $UP \in \mathbb{R}^{m\times 1}$. The user popularity score is then normalized by the maximum popularity score:

$$\widehat{UP}_v = \frac{UP_v}{max(UP_*)} \tag{4}$$

Given that a user's preference varies over diverse items, the user's popularity can be further modelled as the user-item popularity. The popularity score of user v to item i is calculated as:

$$\widehat{UIP}_{vi} = \frac{\widehat{UP}_v}{\widehat{IP}_i} \tag{5}$$

where $i \in I$, $v \in U$, and $\widehat{UIP} \in \mathbb{R}^{m\times n}$.

3.1.3 Hybrid Popularity. The hybrid popularity model combines the influence of both item and user popularities for generating recommendations to a target user. Based on Equation (3), the popularity score of a cold-start target user u is 0 since he has no rating history. To solve this problem, we propose to use the tally of user popularity scores instead. The hybrid popularity score of target user u to item i is calculated as:

$$HP_{ui} = \alpha \times \widehat{IP}_i + \left[(1-\alpha) \times \sum_{v\in U} \widehat{UIP}_{vi}\right] \tag{6}$$

where $\alpha = [0,1]$ controls the ratio of item popularity to user popularity.

3.2 Generating Item Recommendation

Once $HPop$ model is built, we can generate the list of top-N item recommendations to a target user u, denoted as $TopN_u$. In this case, the top-N list will consist of items that have the top-N highest hybrid popularity scores H_{u*}.

3.3 Toy Example

We present this sub-section to demonstrate how the hybrid popularity model for generating item recommendation is built given toy rating data. Table 1 shows a toy example of rating data from 4 users on 7 items. The range of rating is from 1 to 5. The zero value indicates that the user has not rated the item. Note that the target user, i.e., the cold-start user, of the toy example is u_4. Since $m = 4$ and $n = 7$, thus $X \in \mathbb{R}^{4\times 7}$.

To build the hybrid popularity model, we must first build the item and user popularity models. Table 2 shows the item popularity model built out of the toy example rating data, in which the scores are calculated according to Equation (1) and (2). Meanwhile, Table 3 displays the user popularity model where the scores are calculated using Equation (3) and (4). Hence, we construct the user-item popularity model according to Equation (5), as shown in Table 4. Afterwards, we can use Equation (6) to build the hybrid popularity model based on the data in Table 2 and Table 4. For $\alpha = 0.5$, the hybrid popularity score for target user u_4 to item i_1 is calculated as:

$$HP_{41} = (0.5 \times 0.4167) + [(1 - 0.5) \times (1.6800 + 0 + 2.0400)] = 2.0683$$

The complete hybrid popularity scores of the target user u_4 on all items are listed in Table 5. Based on the scores, we can generate the top-3 list of item recommendation to u_4 as $Top3_4 = \{i_5, i_7, i_1\}$.

Table 1. Toy example of rating data

$U \backslash I$	i_1	i_2	i_3	i_4	i_5	i_6	i_7
u_1	3	4	0	5	2	0	0
u_2	0	5	5	4	0	3	3
u_3	2	3	3	0	2	4	3
u_4	0	0	0	0	0	0	0

Table 2. Toy example of item popularity model $\widehat{IP}$

I	i_1	i_2	i_3	i_4	i_5	i_6	i_7
IP	5	12	8	9	4	7	6
$\widehat{IP}$	0.4167	1.0000	0.6667	0.7500	0.3333	0.5833	0.5000

Table 3. Toy example of user popularity model $\widehat{UP}$

U	UP	$\widehat{UP}$
u_1	14	0.7000
u_2	20	1.0000
u_3	17	0.8500
u_4	0	0

Table 4. Toy example of user-item popularity model $\widehat{UIP}$

$U \backslash I$	i_1	i_2	i_3	i_4	i_5	i_6	i_7
u_1	1.6800	0.7000	0	0.9333	2.1000	0	0
u_2	0	1.0000	1.5000	1.3333	0	1.7143	2.0000
u_3	2.0400	0.8500	1.2750	0	2.5500	1.4571	1.7000
u_4	0	0	0	0	0	0	0

Table 5. Toy example of hybrid popularity model HP for u_4

I	i_1	i_2	i_3	i_4	i_5	i_6	i_7
HP_{4i}	2.0683	1.7750	1.7208	1.5083	2.4917	1.8774	2.1000

4 Scalability

As per Equation (2) and (4), we must calculate the item and user popularity scores respectively out of all items and users. We propose to scale-down those expensive processes by implementing a clustering technique on the rating data to create the items clusters. Likewise, we also create clusters of users by employing another clustering technique to demographic information. Note that the characteristics of the two data are different since the former contains numerical data while the latter holds categorical information. In this paper, we implement the Fuzzy C-Means (FCM) to accurately generate item clusters out of numerical rating data [9, 11, 17]. Meanwhile, we use the K-Modes to efficiently create user clusters out of categorical demographic data [18, 19].

Given item clusters CI, we no longer have to calculate the hybrid popularity scores for $i \in I$. Instead, we just need to define which item cluster that we want to focus on generating the list of recommendations. In this paper, we calculate the hybrid popularity scores on items that belong to a cluster with $max(IP_*)$. In other words, first, we calculate $IP \in \mathbb{R}^{1\times n}$ and later scale it down to $\widehat{IP} \in \mathbb{R}^{1\times \tilde{n}}$ where $\tilde{n}$ is the number of items in the selected cluster.

On the other hand, given user clusters CU, we only have to calculate the popularity scores of users that belong to the same cluster as target user u. In this case, we calculate $UP \in \mathbb{R}^{\tilde{m}\times 1}$ and $\widehat{UP} \in \mathbb{R}^{\tilde{m}\times 1}$ where $\tilde{m}$ is the number of users in the selected cluster.

5 Empirical Analysis

This section presents our empirical analysis, based on a series of experiments, to evaluate and compare the performance of our proposed $HPop$ model to other popularity based models.

5.1 Dataset and Experiment Setup

We use the MovieLens 100K dataset [20] to evaluate the performance of our proposed model. The dataset is made up of 100,000 ratings (of 1-5 rating range) from 943 users on 1682 movies. It also contains demographic information for the users, i.e., age, gender, occupation, and zip. In this paper, the scalability problem is solved by generating the item and user clusters respectively out of the rating data and demographic (age, gender, and occupation) information.

The 5-fold cross-validation is implemented that the rating data is randomly split into training and test sets five times, such that $S_{train} = 80\%$ and $S_{test} = 20\%$. Meanwhile, the performance of recommendation is evaluated as the average values out of all folds based on the Normalized Discounted Cumulative Gain (NDCG). Additionally, we also analyze the performance based on Precision and Recall metrics. The evaluations are conducted by comparing the results of item recommendations generated based on the model built from S_{train} with the hidden items of target users in S_{test}, denoted as H_u. Note that since the focus of this paper is on the cold-start problem, we make sure that the (target) users in S_{test} have no rating history in S_{train}.

The scores of NDCG, Precision, and Recall for a target user u in S_{test} at top-N list of recommendations are calculated as:

$$NDCG_u(N) = \frac{DCG_u(N)}{IDCG(N)} \quad (7)$$

$$Precision_u(N) = 100 \cdot \frac{|TopN_u(N) \cap H_u|}{N} \quad (8)$$

$$Recall_u(N) = 100 \cdot \frac{|TopN_u(N) \cap H_u|}{|H_u|} \quad (9)$$

where

$$DCG_u(N) = \sum_{n=1}^{N} \frac{1}{\log_2(1+n)} \cdot \mathbb{I}(TopN_u(n) \in H_u) \quad (10)$$

$$IDCG(N) = \sum_{n=1}^{N} \frac{1}{\log_2(1+n)} \quad (11)$$

Note that $\mathbb{I}(\cdot)$ is the conditional function that results in 1 when true or 0 otherwise.

5.2 Benchmarking

We compare the performance of our proposed $HPop$ model with the two following models:

- The item popularity model ($IPop$). This model implements the assumption that the target user is more interested in the top-rated items. In this case, $IPop$ is equivalent to our $HPop$ model when we set $\alpha = 1$ in Equation (6).
- The user popularity model ($UPop$). This model implements the assumption that the target user is influenced by the top users who have given a large number of ratings. In this case, $UPop$ is equivalent to our $HPop$ model when we set $\alpha = 0$ in Equation (6).

We empirically set the parameters of all models used in this paper to achieve their best performances. Recall that we implement FCM as the item clustering technique and K-Modes as the user clustering technique for a scalability reason. In this case, we set the $number\ of\ item\ clusters = 100$ for FCM and the $number\ user\ clusters = 5$ for K-Modes.

5.3 Results and Discussion

Based on the results of the experiments, we present discussions based on the following three observations.

5.3.1 Impact of α. As shown in Figure 1, the performance of $HPop$ increases gradually along with the increment of α, except for the unexpected anomaly when $\alpha = 0.4$, and achieves its best quality when $\alpha = 0.9$. Afterwards, the performance significantly drops as well as reaches the worst quality when $\alpha = 1$. Based on Equation (6), the results indicate that the employment of the user popularity model is always more beneficial, for solving the cold-start problem in recommendation system than the item popularity model. Note that based on the results of this experiment, we use $\alpha = 0.9$ when comparing the performance of $HPop$ to other models.

5.3.2 Comparison in terms of NDCG. Figure 2 shows the comparison of $HPop$ to $IPop$ and $UPop$ in terms of NDCG. We can observe that $HPop$ always outperforms other models when $N \geq 2$. Only at $N = 1$ that $IPop$ beats $HPop$ and then it remains to fail afterwards. $IPop$ performs worse than $UPop$ when $N \geq 3$. These results confirm our previous finding described in Section 5.3.1, i.e., the $IPop$ must not be exclusively implemented for solving the cold-start problem. The NDCG percentage increases for $N = [1,20]$, displayed in Table 6 show that the average of increases from $HPop$ to $IPop$ and $UPop$ are respectively 12.22% and 8.02%.

5.3.3 Comparison in terms of Precision-Recall. Figure 3 shows the comparison of $HPop$ to $IPop$ and $UPop$ in terms of Precision-Recall. Note that Precision and Recall are less sensitive, compared to NDCG, in terms of the rank of recommended items in a list of recommendations. The experiment results show a similar trend as those described in Section 5.3.2. In this case, we conjecture that the performances of $HPop$, $IPop$, and $UPop$ are stable in any evaluation metrics. In other words, we can generally expect that $HPop$ performs the best while $IPop$ performs the worst in any evaluation metrics.

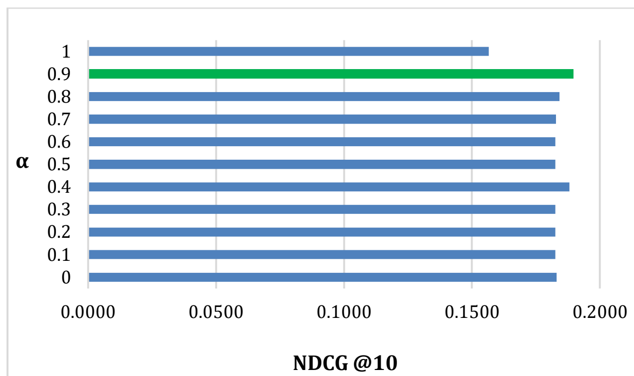

Figure 1. Impact of α to $HPop$

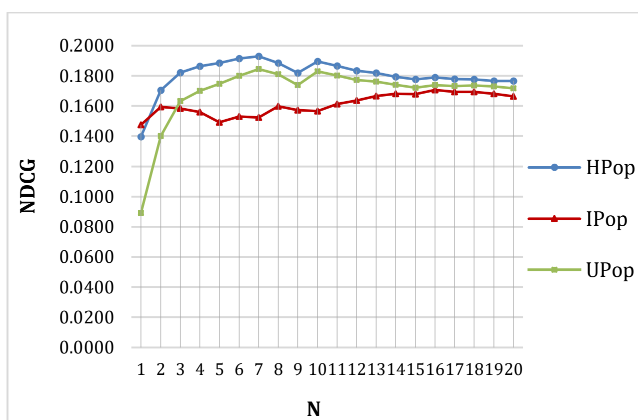

Figure 2. Performance comparison in terms of NDCG

Table 6. NDCG Percentage Increase

N	DCG Score			Percentage Increase	
	$HPop$	$IPop$	$UPop$	$HPop$ VS $IPop$	$HPop$ VS $UPop$
1	0.1397	0.1474	0.0892	-5.21%	56.69%
2	0.1705	0.1594	0.1401	6.96%	21.71%
3	0.1821	0.1583	0.1633	15.04%	11.51%
4	0.1864	0.1560	0.1699	19.47%	9.70%
5	0.1885	0.1493	0.1746	26.30%	7.96%
6	0.1915	0.1530	0.1800	25.15%	6.39%
7	0.1930	0.1524	0.1844	26.66%	4.67%
8	0.1884	0.1599	0.1810	17.87%	4.10%
9	0.1820	0.1573	0.1738	15.74%	4.72%
10	0.1896	0.1566	0.1830	21.09%	3.58%
11	0.1866	0.1613	0.1802	15.73%	3.55%
12	0.1833	0.1638	0.1773	11.96%	3.42%
13	0.1820	0.1666	0.1761	9.21%	3.31%
14	0.1794	0.1681	0.1740	6.71%	3.11%
15	0.1776	0.1680	0.1722	5.71%	3.15%
16	0.1791	0.1707	0.1738	4.92%	3.03%
17	0.1779	0.1693	0.1732	5.05%	2.70%
18	0.1776	0.1694	0.1736	4.82%	2.26%
19	0.1766	0.1681	0.1729	5.06%	2.09%
20	0.1765	0.1664	0.1718	6.11%	2.73%
Average				**12.22%**	**8.02%**
[Min,Max]				**[-5.21%, 26.66%]**	**[2.09%, 56.69%]**

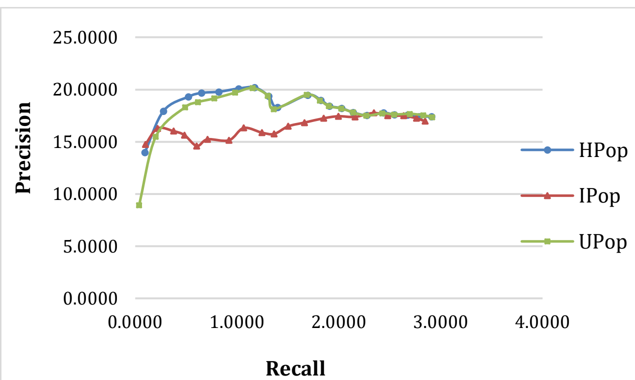

Figure 3. Performance comparison in terms of Precision-Recall

6 Conclusion

This study proposes a new hybrid popularity model for solving the cold-start problem in the recommendation system. Our $HPop$ model is built in three stages: (1) item popularity, (2) user popularity, and (3) hybrid popularity. The ratio of the item and user popularities are controlled by the use of α. From a series of experiments using real-world MovieLens dataset, we can conclude that the employment of the user popularity model is always more beneficial than the item popularity model as $HPop$ performs best when $\alpha = 0.9$ and worst when $\alpha = 1$. To confirm, our experiment results also show the dominancy of $HPop$ over $IPop$ and $UPop$, as well as $UPop$ over $IPop$.

To advance the findings in this study, we plan to test our proposed $HPop$ model on a non-cold-start recommendation system as well as on other datasets. Additionally, it is also worthwhile to study the impact of scalability problem solving by implementing other clustering techniques.

ACKNOWLEDGMENTS

This research is supported by the Directorate General of Higher Degree Education of Indonesia under the “Penelitian Dasar” grant scheme, for the financial year 2020.

REFERENCES

[1] Guo, G., Zhang, J., and Thalmann, D., "Merging trust in collaborative filtering to alleviate data sparsity and cold start," *Knowledge-Based Systems*, vol. 57, pp. 57-68, 2014.

[2] Loh, S., Lorenzi, F., Granada, R., Lichtnow, D., Wives, L. K., and de Oliveira, J. P., "Identifying Similar Users by their Scientific Publications to Reduce Cold Start in Recommender Systems," in Proceeding of *5th International Conference on Web Information Systems and Technologies (WEBIST 2009)*, 2009, pp. 593-600.

[3] Aggarwal, C. C., *Recommender Systems: The Textbook*. Switzerland: Springer International Publishing, 2016.

[4] Ifada, N. and Nayak, R., "An Efficient Tagging Data Interpretation and Representation Scheme for Item Recommendation," in Proceeding of *The 12th Australasian Data Mining Conference*, Brisbane, Australia, 2014, pp. 205-215.

[5] Jadhav, S. L. and Mali, M. P., "Pre-Recommendation Clustering and Review Based Approach for Collaborative Filtering Based Movie Recommendation," *International Journal of Information Technology and Computer Science*, vol. 8, pp. 72-80, 2016.

[6] Sarwar, B., Karypis, G., Konstan, J., and Riedl, J., "Item-based Collaborative Filtering Recommendation Algorithms," in Proceeding of *The 10th International Conference on World Wide Web*, Hong Kong, 2001, pp. 285-295.

[7] Wang, J. and Luo, N., "A New Hybrid Popular Model for Personalized Tag Recommendation," *Journal of Computers*, vol. 11, pp. 116-123, 2016.

[8] Sarwar, B. M., Karypis, G., Konstan, J., and Riedl, J., "Recommender Systems for Large-scale E-Commerce: Scalable Neighborhood Formation Using Clustering," in Proceeding of *The 5th International Conference on Computer and Information Technology*, Dhaka, Bangladesh, 2002, pp. 23-28.

[9] Koohi, H. and Kiani, K., "User based Collaborative Filtering using fuzzy C-means," *Measurement*, vol. 91, pp. 134-139, 2016.

[10] Kumar, N. P. and Fan, Z., "Hybrid User-Item Based Collaborative Filtering," *Procedia Computer Science*, vol. 60, pp. 1453-1461, 2015.

[11] Wei, S., Ye, N., Zhang, S., Huang, X., and Zhu, J., "Collaborative filtering recommendation algorithm based on item clustering and global similarity," in Proceeding of *The 5th International Conference on Business Intelligence and Financial Engineering (BIFE)*, Lanzhou, China, 2012, pp. 69-72.

[12] Nagatani, K. and Sato, M., "Accurate and Diverse Recommendation based on Users' Tendencies toward Temporal Item Popularity," in Proceeding of *RecTemp@RecSys*, Como, Italy, 2017, pp. 35-39.

[13] Steck, H., "Item popularity and recommendation accuracy," in Proceeding of *The fifth ACM conference on Recommender systems*, 2011, pp. 125-132.

[14] Wang, X., Zhang, Y., and Yamasaki, T., "Earn More Social Attention: User Popularity Based Tag Recommendation System," in Proceeding of *Companion Proceedings of the Web Conference 2020*, Taipei, Taiwan, 2020, pp. 212-216.

[15] Gueye, M., Abdessalem, T., and Nacke, H., "A Social and Popularity-Based Tag Recommender," in Proceeding of *The 2014 IEEE Fourth International Conference on Big Data and Cloud Computing* 2014, December. , pp. 318-325.

[16] Barman, K. and Dabeer, O., "Local popularity based collaborative filters," in Proceeding of *The 2010 IEEE International Symposium on Information Theory*, 2010, pp. 1668-1672.

[17] Kapoor, A. and Singhal, A., "A comparative study of K-Means, K-Means++ and Fuzzy C-Means clustering algorithms," in Proceeding of *The 2017 3rd international conference on computational intelligence & communication technology (CICT)*, Ghaziabad, India, 2017, pp. 1-6.

[18] Cao, F., Liang, J., Li, D., Bai, L., and Dang, C., "A dissimilarity measure for the k-Modes clustering algorithm," *Knowledge-Based Systems*, vol. 26, pp. 120-127, 2012.

[19] Khan, S. S. and Ahmad, A., "Cluster center initialization algorithm for K-modes clustering," *Expert Systems with Applications*, vol. 40, pp. 7444-7456, 2013.

[20] Harper, F. and Konstan, J., "The movielens datasets: History and context," *ACM Transactions on Interactive Intelligent Systems*, vol. 5, pp. 1-19, 2015.