ISBN 978-93-86878-06-9
11th International Conference on Data Mining, Computers, Communication and Industrial Applications (DMCCIA-2017)
Kuala Lumpur (Malaysia) Dec. 14-15, 2017

A Novel Imputation-Boosted Technique to Overcome the Unrated Items Issue and Improving the Performance of Collaborative Filtering

Morad Ali Hassan[1], Ali Mansoor Alsahaq[2] and Salem Alseed Alatrash[1]

[1]Computer Science Department School of Graduate Studies, MSU University, 40100, Shah Alam, Malaysia (morad31jun@gmail.com, salemsa83@yahoo.com)
Department of Software Engineering Faculty Of Computer Science & IT, University Of Malaya, 50603, Kuala Lumpur, Malaysia (ali.mansoor@um.my.edu)

***Abstract*—*Personalized recommendation is one of the most popular marketing methods, and collaborative filtering is one of the most successful recommendation technologies [1]. Collaborative filtering recommends mainly according to the rating of users to items; the greater the ratings, the better recommendation performance it will achieve. The premise underlying this process is that a user's previous comparable taste will inform that user's future taste as well. In reality, users usually cannot rate all items and it therefore follows that there is a large number of items that are never rated [2]. In such cases, the user-item rating matrix, will generate a high-dimension sparse (missing values) and so, the similarity calculated by this matrix will not be accurate. So the data sparsity is the key factor influencing the quality of recommendation results. However, the existing CF technique suffers from missing data (unrated items) in the user-item rating matrix. In this paper, an imputation-boosted collaborative filter is applied to deal with this issue and fill in unrated items with predicted values. The proposed method works on finding the different tastes between the active user and the other users, then predicting values for these users to items that remain unrated. The next step involves identifying similarities between the active user and the other user and predicting a value for the active user's unrated item. The results generated are compared with those proposed by other work. The experimental results demonstrate that our method improves the traditional CF technique and outperforms other methods in the case of predicting missing values for each user separately depending on the history of the active user's ratings.*

Keywords: *collaborative filtering, recommender systems, unrated items, missing values, sparsity, imputation-boosted techniques.*

1. Introduction

There has been an accelerated expansion in the amount of data available on the web since web services were first introduced. Of major significance in this context is the need for the extraction of relevant information. Items of potential interest are brought to users' attention by a recommender system. People frequently make choices regarding the purchase of various types of items (e.g. books, movies, etc.) based on others' opinions and experiences [3]. The recommender system makes suggestions of items to a user based on how similar that user is to other users and on item description. Thus, users who are unable to search the web for the items they want on their own could benefit greatly from such a system. Indeed, it can be said that the recommender system fulfills the role of salesperson in the context of e-commerce.

However, collaborative filtering [1] is such a technique that has exhibited promising results in various applications (e.g. Netflix, TiVo, Google news and Amazon). Collaborative filtering leverages the history of neighbors in order to make recommendations to a specific user [4, 5]. There are a number of problems that

influence the accuracy of recommendation using collaborative filtering. One of the most challenging issues is missing rated items. Many users and items are usually included in a dataset. However, not all items receive ratings, as users tend to rate just a handful of popular items, while numerous other items remain unrated. Nevertheless, a recommendation can normally be made for every item. In other words, the user-item matrix is a sparse matrix populated primarily with blank.

To overcome the issue of unrated items, we focus on filling in the values for the missing rating items in the user-item rating matrix while also improving the performance of collaborative filtering to. Our proposed method is to identify the different tastes between the active user and other users to predict the blank cells in the sparse user-item matrix.

In general, the process of filling in the missing values is called *imputation* [6]. The question "which missing values should be imputed and how to impute them?" is of great interest and importance, since some imputation errors will be caused by filled data. In addition, the data sparsity will be alleviated. The case study demonstrates that our proposed work contributes to effectively predicting the unrated items in the user-item rating matrix. The result also demonstrate whether the accuracy of the imputed rating is acceptable and reasonable. This work is organized as follow: The related work is presented in Section 2. Section 3 provides the methodology of our work. Experimental design and result analysis is presented in Section 4. The conclusion is presented in Section 5.

2. Related Work

Collaborative filtering recommends mainly according to the given rating of users for specific items, the greater the number of rated items, the higher the quality recommendation performance it will get. The main issue that recommendation deals with is an approximation of ratings for items that a user has not seen so far. Not all users can view all items or not all items can be rated by every user due to the extremely high number of items and users in the system. So, the quality of recommendation results is highly influences by data sparsity which is considered to be a key factor [2]. Consequently, a technique for approximating the ratings of items that have not been viewed is necessary [7].

A possible solution found to overcome this issue is to use an imputation-boosted collaborative filter techniques [8]. One of the most popular approaches to deal with missing values is to fill them with predicted values (imputation)], to which this work belongs. The missing data can be managed by training models for various combinations of modalities and by selecting an convenient model for each combination [9, 10] or by applying generic methods to combine all modalities in the presence of missing data, such as imputation of the missing data or modification of the fusion algorithm [11]. [12, 13] proposed an imputation boosted collaborative filtering technique (IBCF). They imputes the user-item rating matrix with predicted ratings to alleviate the missing data using different machine learning models. they then use a traditional collaborative filtering technique on the new matrix to generate predicted ratings for specific users. The observations in this technique identified that, it does not take into account how many and which missing data should be imputed?. Moreover, the imputed data is included for all users even the users who had only rated a few items. This action influences the step on finding similar user to the active use which leads to inaccurate` prediction. Filling in all missing ratings with constant values is considered to be the major drawback to these types of researches [14]. Our proposed method solve this problem by filtering the users according to the neighbors who influence the process of prediction.

Recently, [15] they proposed a method that imputes a predicted rating values to a sparse item rating matrix. A trust network (matrix) is provided from the original user-item rating matrix with values of 1 to illustrate the trust relationship between active user and each user to consider the reliability of the predicted values. The goal of this step is to calculate trust values between this user and other users . Then, computing users' confidence values using traditional Pearson correlation coefficient. It follows that, users with confidence values will be computed

and users with non-confidence values will be removed from the active user's trust network. In **Fig I(a)** and **(b)** the goal is to predictthe rating value for u_1 to i_2, where u_2 and u_4 chosen as trust users and u_8 removed from trust network. Since u_2 has a big impact that influences the prediction process, it is noted that, u_2 does not have any rated items in common with u_1 and the taste is completely different. This leads to inaccurate recommendations in some cases.

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	i_9
u_1	4	-	-	5	-	-	4	5	-
u_2	-	2	4	-	-	3	-	-	5
u_3	3	-	-	-	4	-	3	-	-
u_4	-	1	-	3	-	2	-	-	4
u_5	3	-	4	-	4	-	3	5	-
u_6	2	-	-	2	-	-	4	-	-
u_7	-	-	3	-	-	-	-	-	2
u_8	2	3	4	-	4	1	3	-	5

(a)

	u_1	u_2	u_3	u_4	u_5	u_6	u_7	u_8
u_1	1	1	1	-	-	-	-	-
u_2	-	1	1	1	-	-	-	-
u_3	-	-	1	-	1	-	-	-
u_4	-	-	-	1	-	1	-	-
u_5	-	-	-	-	1	1	-	-
u_6	-	-	-	-	-	1	1	1
u_7	1	-	-	-	1	-	1	1
u_8	-	-	-	1	-	-	-	1

(b)

Fig. 1: (a) User-Item Rating Matrix and (b) Trust Network

3. Predicting Unrated Items

The main idea of this approach is to identify the different tastes between the active user and neighbors that have similar rated items in common. We propose the following methodology: classify the user-item rating matrix according to the users who rated the most of items and who rated items in common with the active user; identify the differences between the active user and other users by subtracting the rating scores of items given by both separately. Next, determine the average between each user and active user according to the results obtained from the previous step. Once the average determined, impute the missing rated items that belong to neighbor users by subtracting the rated score given by active users from average result for each neighbor separately (note that the predicted value should be an absolute value and if the result is equal to zero then this means that the missing value should be filled in (imputed) with the rating given by active user to that item, this would interpret that, Zero means no difference in taste). In order, to determine which users have similar or close taste with the active user, we need to define a similarity function. For this, the Pearson Coefficient Correlation is adopted as one of the most used similarity metrics in recommender systems [16]. A prediction step will be performed as the last step to predict missing rating items for active user. The proposed algorithm is below:

3.1. Filtering User-Item Rating Matrix

In this step the user-item matrix will be filtered according to the users who have strong relationships with active user. In other words, users who share enough rated items in common with active user will be selected to act as an active user's neighbors. Hence, users who rated items lower than a specific threshold value θ will be removed from the new user-item rating matrix:

$$\textit{User-Item matrix} = \begin{cases} filter & if\ r_{a,i} \cap r_{b,i} = \theta \\ & 0 \le \delta < K \\ keep & if\ r_{a,i} \cap r_{b,i} = \theta \\ & \delta \ge K \end{cases}$$

Where: $r_{a,i}$ is the rating given to item i by Ua; $r_{b,i}$ is the rating given to item $\boldsymbol{i}$ by Ub; K is defined by admin to describe if the number of items in common with active user less than K then the user-item matrix must be filtered and θ is the threshold value.

3.2. Identifying Different Tastes Between Users:

This step illustrates how much neighbors' opinion is different from that of active user on the items they both rated.

$$\boldsymbol{Diff(Ua, Ub)} = \sum_{i=1}^{m} \left| r_{a,i} - r_{b,i} \right| \quad (1)$$

Where: Ua is the active user; Ub describes all the users in dataset; and *m* is the total number of items .

3.3. Identifying the Average Rating Between Ua, Ub:

In this step the results of the previous step will be divided by the number of items that both the active user and his neighbors rated.

$$\boldsymbol{AVG\ (Ua, Ub)} = \frac{Diff(Ua,Ub)}{n} \quad (2)$$

3.4. Step 4: Imputing the Missing Value of Item $r_{u,i}$ for Ub:

The imputation process in this step predicts a rating to items which the neighbors are not rated but which the active user is already.

$$\boldsymbol{fill\ (Ub, i \rightarrow 0)} = |r_{a,i} - AVG(\boldsymbol{Ua, Ub})| \quad (3)$$

Where: $\boldsymbol{fill\ (Uib \rightarrow 0)}$ is the missing value of $\boldsymbol{Ub}$ on item $\boldsymbol{i}$.

3.5. Neighbor Selection and Similarity Between Ua Ub:

To predict and provide suggestions using the collaborative filtering method, we must first identify the most similar users to the active user and consider them as a set of neighbors of the active user. The active user is a user for whom the goal of the prediction of the target item's score are predicted for [17]. In order to create the set of neighbors, The standard Pearson Correlation Coefficient [] is used. Only the users who voted for the target item are studied. They can be calculated as follows:

$$\boldsymbol{Sim_{Ua,Ub}} = \frac{\sum_{i=1}^{m} (r_{a,i} - \bar{r}_a) \times (r_{b,i} - \bar{r}_b)}{\sqrt{\sum_{i=1}^{m} (r_{a,i} - \bar{r}_a)^2 \times \sum_{i=1}^{m} (r_{b,i} - \bar{r}_b)^2}} \quad (4)$$

Where $\bar{r}_a$ is the mean rating given by user *a*, $\bar{r}_b$ is the mean rating given by user *b*.

The above formula provides similarity computing between two users. In other words, it can computes which user is reliable as an active user's neighbor.

3.6. Prediction

In this step, by using the results obtained from the previous step as the set of active user's neighbors, predict the rating that the active user a will give to the target item i in the future as the weight average of all neighbors' rating on the same item. For this purpose, equation (5) is applied which is typically used in collaborative filtering [16].

$$\boldsymbol{p_{Ua,i}} = \bar{r}_a + \frac{\sum_{b=1}^{m} (r_{b,i} - \bar{r}_b) \times Sim_{a,b}}{\sum_{u=1}^{n} |Sim_{a,b}|} \quad (5)$$

Where, $p_{Ua,i}$ is the prediction for the active user *a* for item *i*; $Sim_{a,b}$ is the similarity between users *a* and *b* ; *n* is the number of users in the neighborhood.

4. Case Study and Result Analysis

The data used in this case study was proposed by [15]. The item-rating matrix with eight users and nine items is presented, where users described as $\boldsymbol{u}$ = { u_1 ,……, u_8} and items as $\boldsymbol{i}$ = { i_1, i_2 ,……, i_9} and where u_1 is considered as the active user (Table I). Each user has rated some of the items with a rating's scale from 1 to 5 to expresses his/her opinion. 1 star indicates that the user feels unworthy and 5 stars indicates that the user feels excellent. If the cell is empty, then means this item has not been rated or highlighted by anyone. In this case, the empty cells will be considered as 0 value.

TABLE I: User-Item Rating Matrix

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	i_9	***Similarity with u_1***	$r_{u1,i} \cap r_{u2,..u8,i}$
u_1	4	0	0	5	0	0	4	5	0	***1***	
u_2	0	2	4	0	0	3	0	0	5	***Removed***	**0** *no items in common*
u_3	3	0	0	0	4	0	3	0	0	***Invalid***	**2**
u_4	0	1	0	3	0	2	0	0	4	***Removed***	*1 low No. of ratings*
u_5	3	0	4	0	4	0	3	5	0	***1***	**3**
u_6	2	0	0	2	0	0	4	0	0	***-0.5***	**3**
u_7	0	0	3	0	0	0	0	0	2	***Removed***	0 *no items in common*
u_8	2	3	4	0	4	1	3	0	5	***Invalid***	**2**

Most of the cells in user-item matrix are empty cells (unrated items). To improve the accuracy of imputing, we first filter the user-item matrix by removing uncommon users with u_1 and users who have low number of rated items where two rated items should be selected as the minimum. In this specific instance, $\boldsymbol{u_2}$, $\boldsymbol{u_4}$, and $\boldsymbol{u_7}$ will be removed from the matrix illustrated in Table I.

$$\textit{User-Item matrix} = \begin{cases} filter & if\ r_{u1,i} \cap r_{b,i} = \mathbf{\theta} \\ & 0 \le \delta < \mathbf{2} \\ keep & if\ r_{u1,i} \cap r_{b,i} = \mathbf{\theta} \\ & \delta \ge \mathbf{2} \end{cases}$$

In our case $\boldsymbol{K}$ defined as 2, $\theta = 0$ in cases of $\boldsymbol{u_2}$ and $\boldsymbol{u_7}$ and $\theta = 1$ in case of $\boldsymbol{u_4}$.

To compute the different tastes of $\boldsymbol{u_1}$ to other users, we first calculate the distance between the ratings given by $r_{1,i}$ and $r_{2,i}$ $r_{3,i}$ $r_{4,i}$ respectively. Note that unrated items for both the active user and other users will not be involved in the computation. According to equation (1) we can obtain:

$$\boldsymbol{Diff(u1, u6)} = \sum_{i=1}^{m} |r_{1,i} - r_{6,i}| = |4-2| + |5-2| + |4-4| = 5$$

$\boldsymbol{Diff(u1, (u3, u5, and\ u8))}$ **= 1, 1, and 1.5** respectively. Hence, $user_3$ and $user_5$ have the closest distance to $user_1$. The less distance, the close the taste.

To compute the average between $user_1$ and $user_2$, $user_3$, $user_4$, respectively, according to equation (2) we can achieve the following:

$$\boldsymbol{AVG\ (u1, u3)} = Diff(u1, u3)/n = 5/3 = 1.666 \cong 1.67$$

By applying the same step on the rest of the users u_5,u_6,,and u_8 we obtained 1, 1.67, and 1.5 respetively. The prediction of unrated items or missing ratings of $\boldsymbol{u_3}$, $\boldsymbol{u_5}$, $\boldsymbol{u_6}$, and $\boldsymbol{u_8}$ can be filled up according to equation (3):

$$\boldsymbol{fill\ (u3, i4 \rightarrow 0)} = |r_{1,4} - 1| = |5-1| = 4$$

Explaining this, we predicted a value to $\boldsymbol{user_3}$ on $\mathbf{item_4}$ by minimizing the rated score given by $\mathbf{user_1}$ on the same item. The rest of the unrated items is predicted using the previous step and is shown in Table II:

TABLE II: Imputed Values and New Similarity Measure

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	i_9	*Similarity with* u_1	$r_{u1,i} \cap r_{u2,\,u3,u5,u6,u8,i}$
u_1	4	0	0	5	0	0	4	5	0	**1**	
u_3	3	0	0	**4**	4	0	3	**4**	0	**1**	**4**
u_5	3	0	4	**4**	4	0	3	5	0	**0.9**	**4**
u_6	2	0	0	2	0	0	4	**3.33**	0	**-0.193**	**4**
u_8	2	3	4	**3.5**	4	1	3	**3.5**	5	**0.816**	**4**

As can be seen above, the neighbors' ratings in common with the active user has increased and led to a change in the similarity measure compared with Table I. The similarity measure has been calculated using equation (4) after applying the proposed method as illustrated below:

$r_{1,i} = 4 + 5 + 4 + 5 = 18$, $\bar{r}_1 = 14/4 = 4.5$, $r_{3,i} = 3 + 4 + 3 + 4 = 14$, and $\bar{r}_3 = 14/4 = 3.5$

$$\boldsymbol{Sim_{U1,U3}} = \frac{(4-4.5)\,(3-3.5) + (5-4.5)(4-3.5) + (4-4.5)(3-3.5) + (5-4.5)(4-3.5)}{\sqrt{((4-4.5)^2(5-4.5)^2(4-4.5)^2(5-4.5)^2) * ((3-3.5)^2(4-3.5)^2}\,(3-3.5)^2(3-4.5)^2} = 1$$

By applying the same steps, we can get the rest of the similarities of other users and the results generated as shown in Table II. The comparison results of the similarity before and after filling the unrated data of the proposed method is described in Figure II (a).

The last stage is to predict the target item rating of user$_1$ to item$_2$ ($r_{1,2}$). By considering $\boldsymbol{u_3}$, $\boldsymbol{u_5}$, $\boldsymbol{u_6}$ and $\boldsymbol{u_8}$ as the neighbors set to the active user ($\boldsymbol{u_1}$), we predict the rating of target item by applying equation (5) to fill in the unrated blank cell of $r_{1,2}$:

Note that, the items not rated by both ($\boldsymbol{u_1}$ and neighbors) will not be predicted according to the proposed technique. The obtained results compared with others is presented in Fig II (b).

$$\boldsymbol{p_{1,2}} = 4.5 + (0-3.5) * 1 + (0-3.75) * 0.9 + (-2.832) * (-0.193) + (3-3) * 0.816/(1 + 0.9 + 0.193 + 0.816) = \mathbf{2.4}$$

TABLE III: The Predicted Ratings of Active User

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	i_9
u_1	4	**2.4**	**3.84**	5	**5**	**2.75**	4	5	**2.88**
u_3	3	0	0	**4**	4	0	3	**4**	0
u_5	3	0	4	**4**	4	0	3	5	0
u_6	2	0	0	2	0	0	4	**3.33**	0
u_8	2	3	4	**3.5**	4	1	3	**3.5**	5

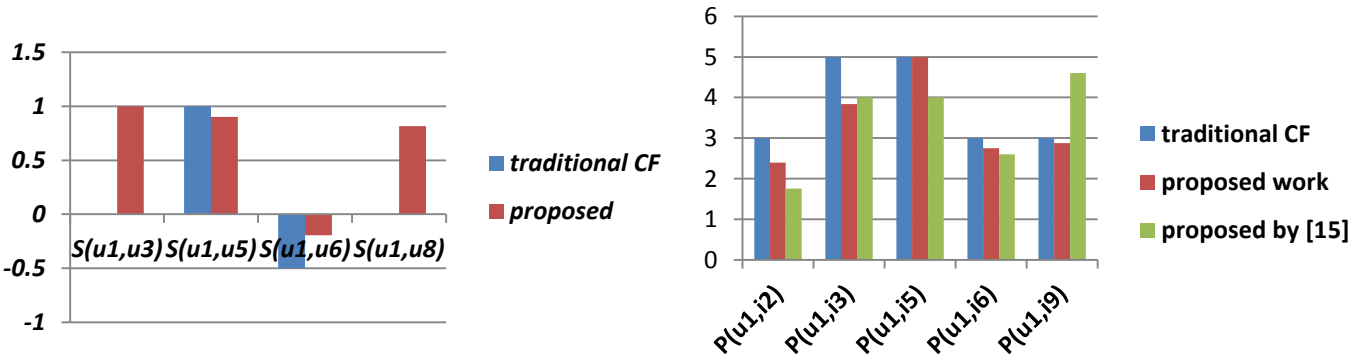

Fig II: (a) Similarity Comparison of Initial Rating and after Proposed Work, and (b) Prediction Comparison of Initial Rating, Proposed Work, and Proposed by [15]

According to the results presented by [15], u_8 was removed from their trust network and compared with our proposed work u_8 had a high impact on the predicted value of u_1 to i_2. In the case of the predicting value of u_1 to i_9, they relied on the ratings given by u_2 and u_4 where the former has no rated items in common with u_1 and the latter has very low rated items with u_1. Logically, $r_{u1,i9}$ should not be predicted as high as 4.6 in case most of th users in common with the active user not rated this item.

5. Conclusion

As acknowledged in collaborative filtering, the greater the number of rated items, the more accurate the prediction we get. To increase the number of rated items in the sparse user-item rating matrix where some of the active user's neighbors did not rate some items that the active user already rated and missing values need to be predicted, a new technique was proposed that improves the performance of collaborative filtering by imputing predicted values without changing the original data (rated items) and no additional data is required. The predicted values are obtained from identifying the differences in behavior between active the user and his/her neighbors. By comparing our results with that obtained from traditional collaborative filtering and the one proposed by other work on the same, we found that our technique improved the first and outperformed the second where some of the active user's predicted ratings should be low or high according to the behavior of his/her neighbors. This therefore indicates that the predicted ratings rely on the active user's neighbors. Finally, the proposed technique is very simple and can be used for many kinds of recommender systems under any domain. The ease of its implementation enables interested people to apply it on their systems.

6. References

[1] P. Wang, Q. Qian, Z. Shang, and J. Li, "An recommendation algorithm based on weighted Slope one algorithm and user-based collaborative filtering," in *Control and Decision Conference (CCDC), 2016 Chinese*, 2016, pp. 2431-2434.

[2] M. Hongwei, Z. Guangwei, and L. Peng, "Survey of collaborative filtering algorithms," *Journal of Chinese Computer Systems*, vol. 30, pp. 1282-1288, 2009.

[3] F. G. Davoodi and O. Fatemi, "Tag based recommender system for social bookmarking sites," in *Advances in Social Networks Analysis and Mining (ASONAM), 2012 IEEE/ACM International Conference on*, 2012, pp. 934-940.

[4] J. A. Konstan, B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon, and J. Riedl, "GroupLens: applying collaborative filtering to Usenet news," *Communications of the ACM*, vol. 40, pp. 77-87, 1997.

[5] B. Smyth and P. Cotter. (2001) Personalized Electronic Programme Guides. *Artificial Intelligence Magazine 21(2)*.

[6] J. L. Schafer, *Analysis of incomplete multivariate data*: CRC press, 1997.

[7] E. Negre, "Recommender Systems," *Information and Recommender Systems*, pp. 7-27, 2015.

[8] P. Melville, R. J. Mooney, and R. Nagarajan, "Content-boosted collaborative filtering for improved recommendations," in *Aaai/iaai*, 2002, pp. 187-192.

[9] O. Fatukasi, J. Kittler, and N. Poh, "Estimation of missing values in multimodal biometric fusion," in *Biometrics: Theory, Applications and Systems, 2008. BTAS 2008. 2nd IEEE International Conference on*, 2008, pp. 1-6.

[10] R. Vera-Rodriguez, P. Tome, J. Fierrez, and J. Ortega-Garcia, "Fusion of footsteps and face biometrics on an unsupervised and uncontrolled environment," in *SPIE Defense, Security, and Sensing*, 2012, pp. 83711U-83711U-8.

[11] M. Aste, M. Boninsegna, A. Freno, and E. Trentin, "Techniques for dealing with incomplete data: a tutorial and survey," *Pattern Analysis and Applications*, vol. 18, pp. 1-29, 2015.

[12] X. Su, T. M. Khoshgoftaar, and R. Greiner, "Imputed neighborhood based collaborative filtering," in *Web Intelligence and Intelligent Agent Technology, 2008. WI-IAT'08. IEEE/WIC/ACM International Conference on*, 2008, pp. 633-639.

[13] X. Su, T. M. Khoshgoftaar, X. Zhu, and R. Greiner, "Imputation-boosted collaborative filtering using machine learning classifiers," in *Proceedings of the 2008 ACM symposium on Applied computing*, 2008, pp. 949-950.

[14] S. Ghazarian and M. A. Nematbakhsh, "Enhancing memory-based collaborative filtering for group recommender systems," *Expert systems with applications*, vol. 42, pp. 3801-3812, 2015.

[15] M. M. Azadjalal, P. Moradi, A. Abdollahpouri, and M. Jalili, "A trust-aware recommendation method based on Pareto dominance and confidence concepts," *Knowledge-Based Systems*, vol. 116, pp. 130-143, 2017.

[16] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "GroupLens: an open architecture for collaborative filtering of netnews," in *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, 1994, pp. 175-186.

[17] N. Rastin and M. Z. Jahromi, "Using content features to enhance performance of user-based collaborative filtering performance of user-based collaborative filtering," *arXiv preprint arXiv:1402.2145*, 2014.