User Polarization Aware Matrix Factorization for Recommendation Systems

Wenlong Sun[a], Olfa Nasraoui[a]

[a]University of Louisville, 2301 S 3rd St, Louisville, Kentucky, USA, 40292

Abstract

User feedback results in different rating patterns due to the users' preferences, cognitive differences, and biases. However, little research has taken into account cognitive biases when building recommender systems. In this paper, we propose novel methods to take into account user polarization into matrix factorization-based recommendation systems, with the hope to produce algorithmic recommendations that are less biased by extreme polarization. Polarization is an emerging social phenomenon with serious consequences in the era of social media communication. Our experimental results show that our proposed methods outperform the widely-used methods while considering both rank-based and value-based evaluation metrics, as well as polarization-aware metrics.

Keywords

Polarization, Matrix Factorization, Recommender System

1. Introduction

Ratings are loaded with biases which are caused by factors related to users, items, and the interaction between users and items. For example, harsh critics might give low ratings compared with the majority of users. Similarly, the popularity of an item generally results in higher ratings for that item [1]. Also the interaction between humans and recommender systems creates a feedback loop which introduces iterated bias leading to biased recommendations [2, 3, 4]. Although, modern recommender systems can make highly personalized predictions, few have considered these biases. Recent studies also emphasized the importance of tackling bias in recommender systems, which can be exacerbated with the propagation of harmful content and intentional spreading of hatred content [5].

Social influence bias is defined as the 'conformity' engendered by answers of other participants characterized by responses similar to the community 'norm' [6, 7, 8]. Social influence bias can yield ratings which are closer to the average, i.e. less diverse and less representative of participants' true interests for items, which may produce biases in an essential component of neighborhood based collaborative filtering, namely assessing the similarity between items and between users. Problems that might happen when biases are introduced include: 1) bias might contaminate the recommender system's inputs, weakening the systems' ability to provide high-quality recommendation; 2) bias can artificially pull the consumer's preference towards displayed recommendations; 3) bias can lead to a distorted view of items' quality.

Anti-social influence bias is defined as the user's tendency to overrate items if they are presented with low average ratings. On Amazon [1] and Netflix [2] for example, users were found to feel the need to increase some ratings and to want the items to receive rewards from their ratings [9]. On the other hand, users may give low ratings if they think that the items do not deserve their ratings, since they want to prevent other users from considering or purchasing those items [9].

Critical users tend to give low ratings to almost all items, while enthusiastic users give high ratings. Some users are very extreme, they either give very high ratings or very low ratings, which leads to a 'U' shape rating distribution. We define this user pattern as 'polarized-users', similar to the notion of 'polarized-items' [10].

OHARS'21: Second Workshop on Online Misinformation- and Harm-Aware Recommender Systems, October 2, 2021, Amsterdam, Netherlands

wenlong.sun@louisville.edu (W. Sun); olfa.nasraoui@louisville.edu (O. Nasraoui)

https://github.com/wenlongsunuofl (W. Sun); http://webmining.spd.louisville.edu/ (O. Nasraoui)

0000-0003-0164-8733 (W. Sun)

[1]www.amazon.com

[2]www.netflix.com

Previous studies in recommendation systems focused on user bias by modeling user bias as Gaussian noise. Shan & Banerjee model user and item bias using a residual approach by removing these biases before applying their latent factor based collaborative filtering approach [1]. Koenigstein et al. identified biases that are related and unrelated to personal features and proposed models to clean signals that are unrelated to personalization purposes, such as item taxonomy, user's rating in a consecutive session, and item's temporal dynamics [11]. In the above studies, researchers assumed that the rating values are ratio scaled even though the true property of ratings might be interval or ordinal scaled.

Some researchers considered cognitive aspects of biases, which reveal different response styles or style biases in rating values of recommendation systems. From these studies, various types of responses were revealed, and researchers attempted to detect and correct bias using computational models [9]. In most recommendation systems, active users tend to choose what they want to rate using explicit ratings. Various types of biases are introduced in this stage caused by user interfaces, user's affection states, or attributes of items. Park at al. proposed a categorization of ratings into six different patterns (see figure 1) [12]. Like-biased (LB) type users may rate items higher. Users might want a movie to get high ratings if they think that it is better than the current average rating. Dislike-biased (DB) type users tend to rate movies they hate, or think are bad and may feel it as their duty to prevent others from watching them. Some users show Bi-Polar (BP) type ratings in which they show both LB and DB style. And users with extremely indecisive responses are categorized as Neutral-Biased (NB) [12]. Normal rating pattern users are denoted as type N, while a vague rating pattern (V) corresponds to users whose ratings have no specific pattern [12]. Chitra et al. studied, using graph-based techniques, how filter bubbles in recommender systems have exacerbated polarization [13]. [14] proposed an approach called PrCP, to depolarize matrix factorization (MF) recommendations where ratings are depolarized before being fed as input to MF.

In this paper we modify matrix factorization-based collaborative filtering with the aim of producing algorithmic recommendations that are less biased by extreme polarization. The rest of the paper is organized as follows: We start with a presentation of our proposed methodology in Section 2, then follow with our evaluation experiments in Section 3, and end with our conclusions in Section 4.

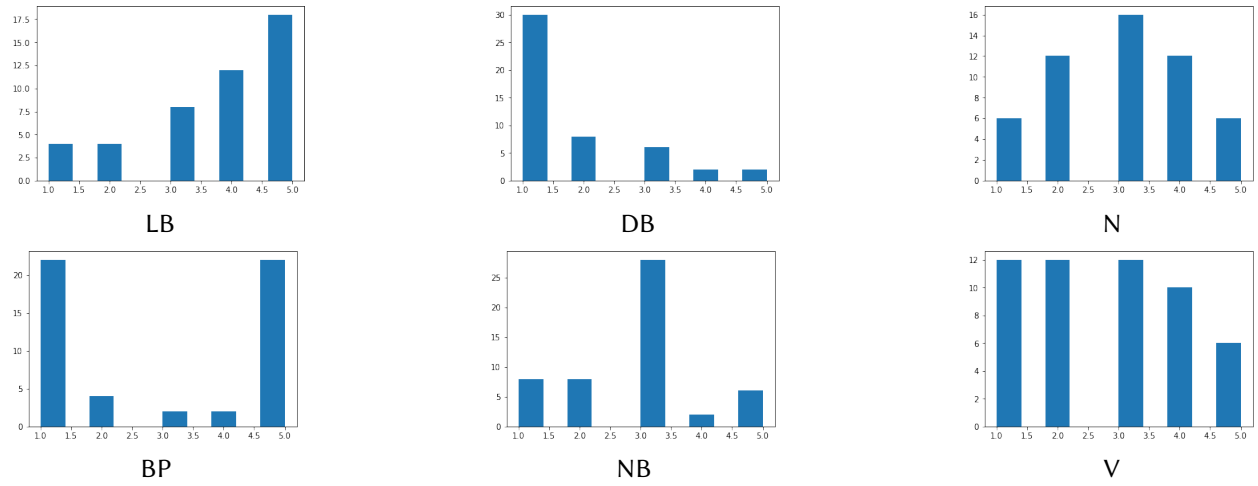

Figure 1: Six different rating patterns

2. Methodology

2.1. Polarization Detector

Our idea is inspired by the polarization phenomena studied in [10, 14] which aimed to counteract the polarized *item* problem in a recommender systems. [10] first constructed a feature set from a user's rating histogram; and trained a classifier on labeled data. There, polarization was defined based only on the ratings of an *item* by all users. In this work we further consider polarization that is based on the ratings given by a *user* to all items. Given a recommender environment $G = (U; I; R)$, where user $u \in \mathbb{R}^{1\times n}$ had rated item $i \in \mathbb{R}^{m\times 1}$ with rating $r_{ui} \in \mathbb{R}^{m\times n}$ on a scale of x to y, User u's *polarization score* ϕ_u is defined as the spread of their ratings r_u. Because [10]'s polarization detector was shown to work better than existing polarization detection

algorithms (see details in [10]), we adopt its approach to build a user polarization detector to calculate a *user* polarization score instead of an item polarization score.

2.2. User-polarization-aware Recommendation system

In this section, we exploit a user polarization detector in depolarizing a recommendation system. We first transform the original ratings to account for polarization. The proposed solution tries to counteract polarization by making the training data set less polarized, and employs a stochastic mapping function as defined below: $f : (U, I\, R) \rightarrow (U, I, R^{'})$.

The transformation function aims to transform a user-item rating based on the rating itself, polarization score, and user's current rating standard deviation, as follows

$$\begin{array}{ll} r^{'}_{ui} = r_{ui} + \Phi_u \bigtimes \frac{1}{1+e^{-std(u)}} & if\, r_{ui} < \delta \\ r^{'}_{ui} = r_{ui} - \Phi_u \bigtimes \frac{1}{1+e^{-std(u)}} & if\, r_{ui} \geqslant \delta \end{array} , \tag{1}$$

where r_{ui} is the original rating and δ is a threshold on ratings for user u defined as the rating mid-range. ϕ_u is the polarization score of user u. When the polarization score is 0, there is no change to the original ratings. Finally, $std(u)$ is the standard deviation of the user's ratings.

The purpose of this transformation is to reduce the intensity of polarization of the ratings of the user. Figure 3 shows the effect of this transformation. The yellow bar represents all the ratings of user 16 for 100 jokes from the Jester dataset (which we use because it is a complete rating data set), while the red bar shows the transformed ratings. We also compare the polarization score before and after transformation.

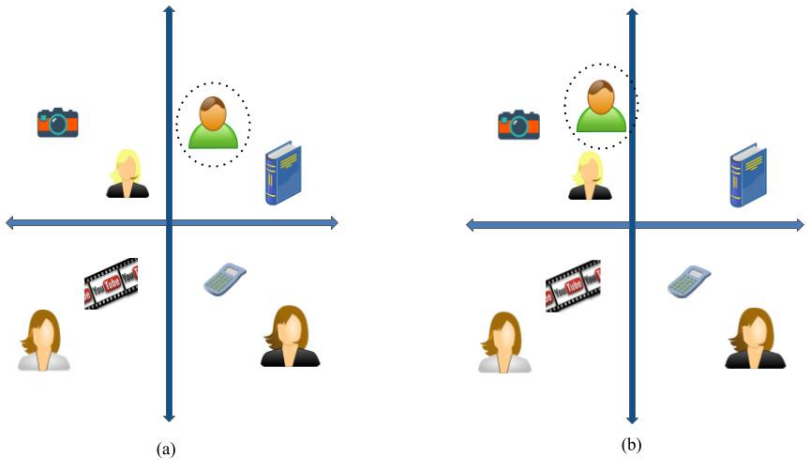

Figure 2: How the transformation affects the user-item polarization in the latent feature space.

Considering collaborative filtering systems, the intuition behind transforming the original ratings is to bring users closer in latent space after the transformation. Figure 2 shows the intuition of counter-polarization in matrix factorization. Sub-figure 2(a) shows the original data in latent space, while the sub-figure 2(b) shows the data after transformation. By moving a polarized user closer to other users in latent space, we obtain less polarized recommendations.

Matrix Factorization (MF)-based recommender systems have shown good performance by combining good scalability with predictive accuracy [9]. Therefore, we use the Non-negative Matrix Factorization (NMF) recommendation algorithm to test the effect of our proposed transformation. Recall that the general concept of NMF is to decompose a non-negative data matrix R with size $m \times n$ into two positive element matrix factors P and Q, with size $m \times k$ and $k \times n$, respectively, such that $k << min(n, m)$ is a positive integer representing the rank of matrices P and Q. A classical NMF predicts the overall rating r_{ij} by minimizing $||r_{ij} - p_i.q_j||^2$.

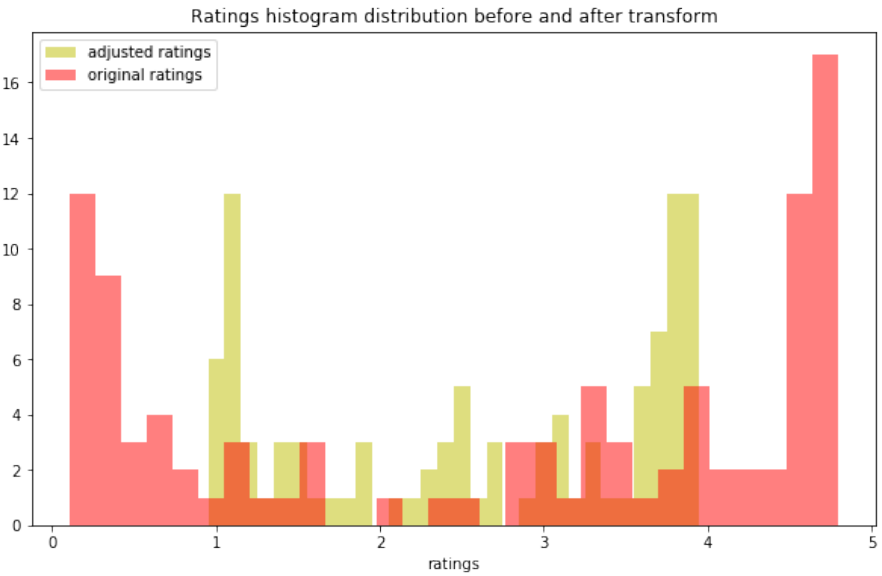

Figure 3: Effect of proposed transformation on original ratings

Bringing the transformed ratings into the objective, the final objective function for User-polarization-aware matrix factorization (UpaMF) is:

$$\min_{p_u,q_i} \sum_{u\in U}\sum_{i\in I} ||r'_{ui} - p_u.q_i||^2 + \beta(||p_u||^2 + ||q_i||^2) \tag{2}$$

Here r'_{ui} is the transformed rating, while p_u and q_i are the latent feature vectors for users and items, respectively. Using stochastic gradient descent, we obtain the following updating rule to minimize the objective function:

$$\begin{aligned} p_u &\leftarrow p_u + \eta(2e_{ui}q_i - 2\beta p_u) \\ q_i &\leftarrow q_i + \eta(2e_{ui}p_u - 2\beta q_i) \end{aligned} \tag{3}$$

Here $e_{ui} = \hat{r}'_{ui} - p_u q_i$, and η is the learning rate for gradient descent.

Algorithm 1: User-polarization-aware Matrix Factorization (UpaMF)

Data: Transformed rating matrix R'_{ui}, β, η
Result: Latent features P_u and Q_i
Initialize P_u and Q_i;
Calculate error using Eq. 2;
while *Stopping criteria is not met* **do**
  Update P_u using $p_u \leftarrow p_u + \eta(2e_{ui}q_i - 2\beta p_u)$;
  Update Q_i using $q_i \leftarrow q_i + \eta(2e_{ui}p_u - 2\beta q_i)$;
end

2.3. Weighted User-polarization-aware Recommendation system

The above User-polarization-aware Recommendation System takes advantage of the polarization score obtained from the polarization detector. However, the original ratings matrix information is completely ignored. We next propose a *weighted* user-polarization-aware recommendation system, to allow a flexible weighting of the original ratings. The objective of the weighted user-polarization-aware matrix factorization (WUpaMF) is defined as:

$$\begin{aligned} \min_{p_u,q_i} \sum_{u\in U}\sum_{i\in I} \Phi_u||r'_{ui} - p_u q_i||^2 + (1-\Phi_u)||r_{ui} - p_u q_i||^2 \\ +\beta(||p_u||^2 + ||q_i||^2) \end{aligned} \tag{4}$$

where Φ_u weights the influence of the original rating factorization, and is also the user polarization score. When $\Phi_u = 0$, the objective reduces to regular matrix factorization, while it reduces to UpaMF when $\Phi = 1$. The updating rule, using gradient descent, for this new objective is:

$$\begin{aligned} p_u &\leftarrow p_u + \eta(2((1-\Phi_u)e_{ui} + \Phi_u e'_{ui})q_i - 2\beta p_u) \\ q_i &\leftarrow q_i + \eta(2((1-\Phi_u)ue_{ui} + \Phi_u e'_{ui})p_u - 2\beta q_i) \end{aligned} \tag{5}$$

Here $e'_{ui} = r'_{ui} - p_u q_i$, $e_{ui} = r_{ui} - p_u q_i$, and η is the learning rate for gradient descent.

Algorithm 2: Weighted User-polarization-aware Matrix Factorization (WUpaMF)

Data: Original rating R_{ui},Transformed rating matrix R'_{ui}, β, η, Φ_u
Result: Latent features P_u and Q_i
Initialize P_u and Q_i;
Calculate error using Eq. 4;
while *Stopping criteria is not met* **do**
  Update P_u using $p_u \leftarrow p_u + \eta(2((1-\Phi_u)e_{ui} + \Phi_u e'_{ui})q_i - 2\beta p_u)$ Update Q_i using $q_i \leftarrow q_i + \eta(2((1-\Phi_u)ue_{ui} + \Phi_u e'_{ui})p_u - 2\beta q_i)$;
end

3. EXPERIMENTAL RESULTS

We used the Jester dataset, which contains 4.1 million ratings by 73,421 users on 100 jokes, collected between April 1999 and May 2003 [15]. Jester uses continuous ratings in the range of $[-10, 10]$ with 99 meaning unavailable ratings, and has a rating density of 40%. In our experiment, in order to discern clear patterns of the user's rating we use the top 23,983 users who rated at least 36 jokes out of the total 100.

We first built a rating histogram of each user's ratings, and labeled the ratings into Polarized or Non-Polarized. Finally, we obtained 517 polarized users. In order to have a balanced dataset, we then randomly selected 517 non-polarized users to form a combined dataset (517 polarized users and 517 non-polarized users). We randomly left out 205 users (20% of the final data) for testing the polarization detector. All in all, we ended up with a rating matrix of 1034 users by 100 items. Note that the density of those ratings is around 94%, which indicates that most of the users rated all 100 items. We then randomly split all ratings into a training (90%) and testing set (10%). For both UpaMF and WUpaWF, we set $iteration = 200$, $\beta = 0.02$ and $\eta = 0.002$ for training in Eq. 3 and Eq. 5.

3.1. Polarization detection

We trained our polarization detector using the Random Forest Algorithm [10], and report the 5-fold cross validation results. As seen in the ROC curve shown in Figure 4, the polarization detector can precisely detect the polarization of the users' rating patterns.

Figure 4 shows that the constructed feature set can capture the polarization trends of user ratings. Figure 5 shows that the user polarization scores before the transformation are very unbalanced, and more users have very high polarization scores. On the other hand, the polarization scores after transformation are less extreme.

3.2. Counter polarization recommendation system results

We evaluated the performance of our proposed method in terms of rating prediction accuracy, i.e., the Mean Average Error (MAE) [9]. We also compared the NDCG@10 scores for rank-based evaluation [16], as well as the diversity and novelty scores to measure the diversity of recommendations [17]. The novelty score measures how novel the top-N recommendations are. We use the popularity to measure the novelty, i.e.,

$$Novelty(i) = -log_2 p(i), \tag{6}$$

where $p(i)$ is the popularity of item i in the top-N recommended item list, i.e., $p(i) = \frac{\#\ of\ users\ who\ rated\ i}{\#\ of\ total\ users}$. We then average these values across all the test users. For the diversity, we use the distance-based item novelty [17]. For a recommended item i, the diversity is calculated using

$$Diversity(i) = min_{j \in Recs, j \neq i} d(i, j), \tag{7}$$

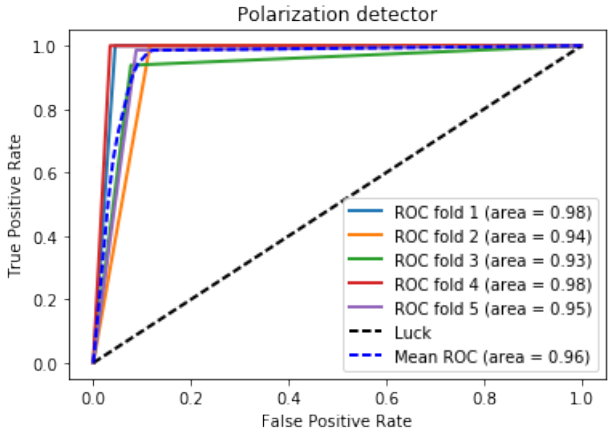

Figure 4: ROC of polarization detector

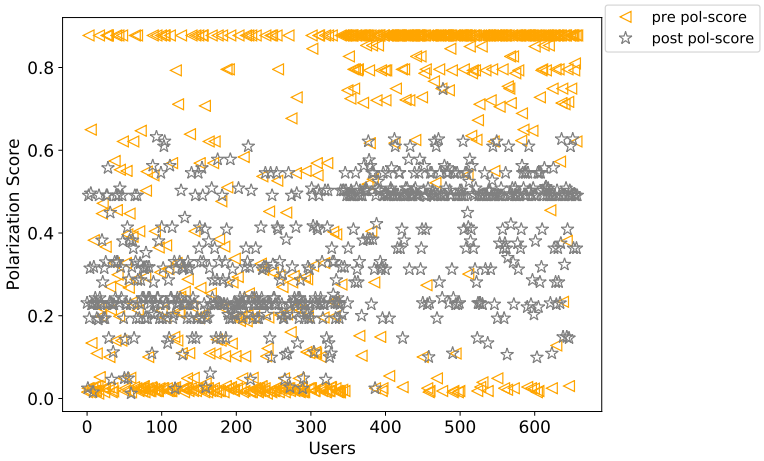

Figure 5: Polarization score before and after transformation

where $Recs$ is the set of top-N recommended items for a user and $d(i, j)$ is the distance between item i and j calculated using the cosine similarity based on the rating vector. Then, we average over all the recommended items for the user.

Another way to measure the debiasing effectiveness is to check the **blind spot size** (see Eq. 8). The blind spot size is defined as the number of items with predicted ratings $\hat{R_{u,i}}$ lower than a given threshold δ, i.e., $\mathbf{D_\delta^u} = \{\mathbf{i} \in I \mid \hat{R}_{u,i} < \delta\}$. Note that because each user has their own blind spot, we define a threshold for each user. The threshold is found based on a percentile cut-off for each user, set to 95%. Therefore, We define the blind spot for user u as

$$\mathbf{D_\epsilon^u} = \{\mathbf{i} \in I \mid \hat{R}_{u,i} < max(pred) * \epsilon\}, \tag{8}$$

where ϵ is threshold which controls the cutoff. Modern recommendation systems suffer from two important issues: filter bubbles [18], and blind spots [2, 19, 20, 3, 21] (see Fig. 6). Generally, Recommender Systems (RSs) should aim to avoid the filter bubble, as well as reduce the size of the blind spot. If the blind spot size is too big, users will only have a limited ability to discover possibly interesting items. If the filter bubble size is too extreme, users are limited to discover only certain types of items.

We check how many times an item falls into the blind spot based on the predicted ratings across all users who have not rated or seen this item yet. We use $\mathcal{R}_u = \{i \in I | user \quad u \quad has \quad rated \quad item \quad i\}$ as the observed ratings for user u. Therefore the BL score for an an item i is computed as follows:

$$BL_{score}^{(i)} = \frac{\sum\limits_{u \in U} |i \in D_\epsilon^u \quad and \quad i \notin \mathcal{R}_u|}{\sum\limits_{u \in U} |i \notin \mathcal{R}_u|.} \tag{9}$$

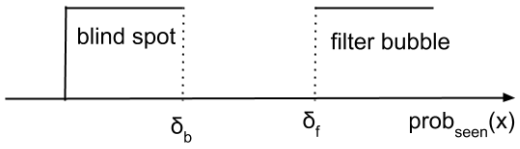

Figure 6: The blind spot and filter bubble in a recommender system. Here, δ_b represents the threshold up to which items that have a lower probability of being seen will not be discovered by users. On the other hand, items with probability of being seen higher than δ_f will likely be discovered by users.

The numerator is the total number of users who has not rated item i and have item i in their blind spot based on the prediction. The denominator is the total number of users who have not rated item i. For example, if an item falls into the blind spot 100 times based on Eq. 8 across all 400 users who have not rated this item, then the BL score for this item will be $100/400 = 0.25$.

We applied the polarization detector to the testing set to obtain the polarization score, then we transformed the rating using Equation 1. We also computed the Mean Average Precision (MAP) on the top-N recommendation list [17]. Five algorithms were applied on the transformed rating matrix or the original rating matrix:

1. Regular Matrix Factorization (Reg MF) on riginal ratings;
2. Bias-considered matrix factorization (Bias considered MF) on original ratings (in which biases from users, items and the overall rating are added) [9];
3. User-polarization-aware matrix factorization (Upa MF) on transformed ratings;
4. Bias-considered User polarization- aware matrix factorization on transformed ratings (Upa bias Considered MF);
5. Weighted user-polarization-aware matrix factorization (WUpa MF).

For 5), we need to compare all the above metrics on the original ratings and transformed ratings separately. We repeated the experiments 10 times and report the averaged scores.

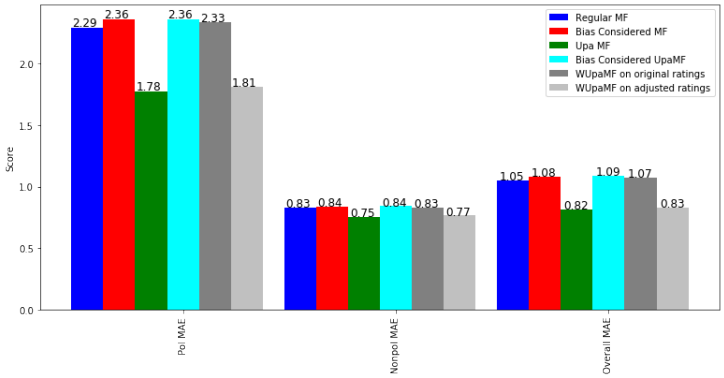

Figure 7: MAE of MF and different methods to handle user polarization

We report all the metrics above for both polarized and non-polarized users. The WUpa MF in Eq. 4 involves both the transformed and original ratings; therefore we compare all the metrics above based on both the original ratings and transformed ratings. Figure 7 shows the MAE on both polarized users, non-polarized users and all users. Our proposed Upa MF has the lowest MAE in all groups. Figure 8 shows that our proposed Upa MF algorithm has the highest NDCG@10 score, while Bias considered Upa MF has the lowest NDCG@10 score.

We report the diversity and novelty results in Figure 9 and Figure 10. As shown in Figure 9,the Non-polarized user group has higher diversity score using Bias considered MF; but overall, Upa MF results in the highest diversity score.

We also report the (BL) score distribution for all five compared algorithms in Figure 11. Bias considered Upa has the highest BL score, indicating that this algorithm boost only a few items. On the other hand, Upa MF achieves the lowest BL score, promoting several unpopular items to be explored by users. Finally figure 10

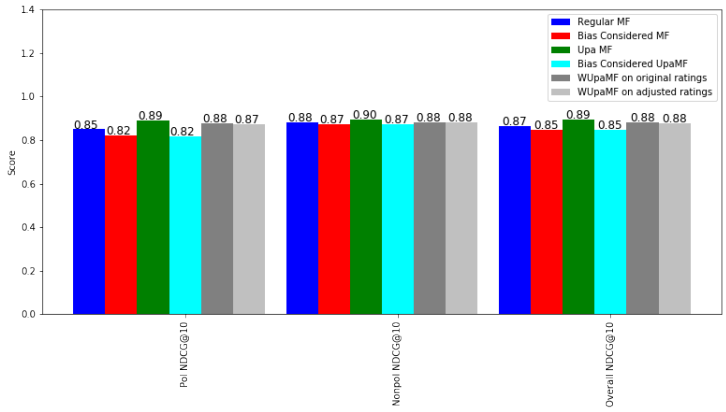

Figure 8: NDCG@10 of MF and different methods to handle user polarization

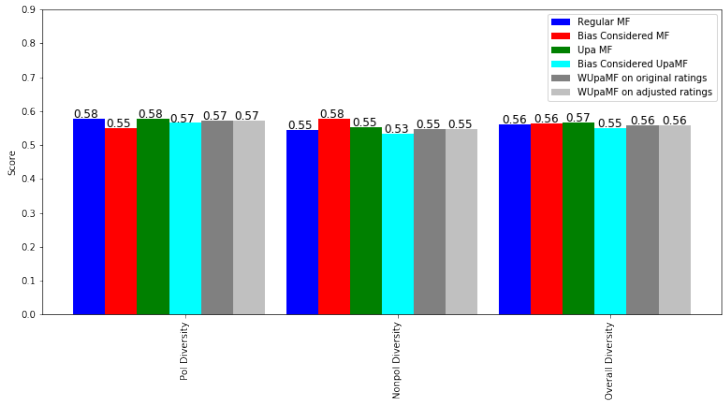

Figure 9: Diversity@10 of of MF and different methods to handle user polarization

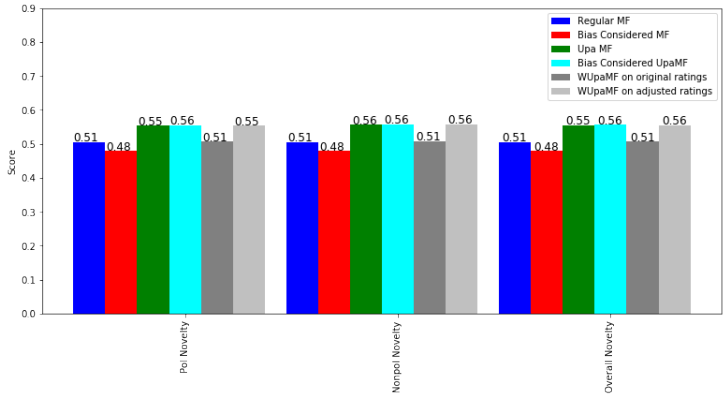

Figure 10: Novelty@10 of MF and different methods to handle user polarization

shows that Upa MF and Bias considered MF have comparable novelty scores and they both exceed the novelty of baseline MF and bias-considered MF.

4. Conclusion

Users have been fond to exhibit different rating patterns in previous studies. However, few studies have taken into account the users' *polarized* rating pattern. In this paper, we first adapted our previous item polarization detection technique to obtain a polarization score for each user, which was then used in a novel User-polarization-aware matrix factorization (UpaMF) algorithm to obtain less polarized recommendations. We also proposed a weighted User-polarization-aware MF algorithm. Our results show that our proposed

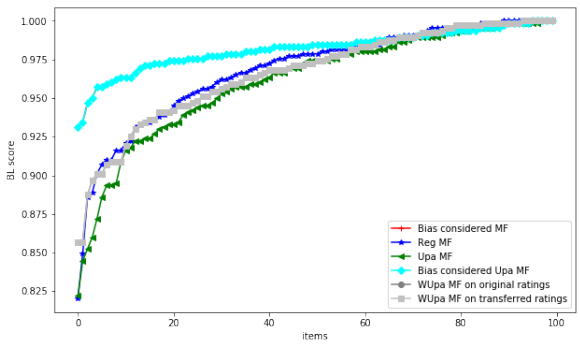

Figure 11: Blind spot score of MF and different methods to handle user polarization. The x-axis is all the items sorted in ascending order of BL score, and the y-axis is the BL score.

techniques outperformed the naive MF baseline and the bias-aware Matrix Factorization in terms of both rank-based and value-based evaluation metrics, while improving the recommendation lists' diversity and reducing the blind spots induced by the recommendations.

5. Acknowledgement

This work was supported by National Science Foundation grant NSF-1549981.

References

[1] H. Shan, A. Banerjee, Generalized probabilistic matrix factorizations for collaborative filtering, in: Data Mining (ICDM), 2010 IEEE 10th International Conference on, IEEE, 2010, pp. 1025–1030.

[2] W. Sun, O. Nasraoui, P. Shafto, Iterated algorithmic bias in the interactive machine learning process of information filtering., in: KDIR, 2018, pp. 108–116.

[3] W. Sun, S. Khenissi, O. Nasraoui, P. Shafto, Debiasing the human-recommender system feedback loop in collaborative filtering, in: Companion Proceedings of The 2019 World Wide Web Conference, 2019, pp. 645–651.

[4] S. Khenissi, B. Mariem, O. Nasraoui, Theoretical modeling of the iterative properties of user discovery in a collaborative filtering recommender system, in: Fourteenth ACM Conference on Recommender Systems, 2020, pp. 348–357.

[5] A. Tommasel, D. Godoy, A. Zubiaga, Workshop on online misinformation-and harm-aware recommender systems, in: Fourteenth ACM Conference on Recommender Systems, 2020, pp. 638–639.

[6] S. Moscovici, C. Faucheux, Social influence, conformity bias, and the study of active minorities, Advances in experimental social psychology 6 (1972) 149–202.

[7] W. Wood, Attitude change: Persuasion and social influence, Annual review of psychology 51 (2000) 539–570.

[8] P. M. DeMarzo, D. Vayanos, J. Zwiebel, Persuasion bias, social influence, and unidimensional opinions, The Quarterly Journal of Economics 118 (2003) 909–968.

[9] Y. Koren, R. Bell, C. Volinsky, Matrix factorization techniques for recommender systems, Computer 42 (2009).

[10] M. Badami, O. Nasraoui, W. Sun, P. Shafto, Detecting polarization in ratings: An automated pipeline and a preliminary quantification on several benchmark data sets, in: 2017 IEEE International Conference on Big Data (Big Data), IEEE, 2017, pp. 2682–2690.

[11] N. Koenigstein, G. Dror, Y. Koren, Yahoo! music recommendations: modeling music ratings with temporal dynamics and item taxonomy, in: Proceedings of the fifth ACM conference on Recommender systems, ACM, 2011, pp. 165–172.

[12] K.-W. Park, B.-H. Kim, T.-S. Park, B.-T. Zhang, Uncovering response biases in recommendation., in: MPREF@ AAAI, 2014.

[13] U. Chitra, C. Musco, Analyzing the impact of filter bubbles on social network polarization, in: Proceedings of the 13th International Conference on Web Search and Data Mining, 2020, pp. 115–123.
[14] M. Badami, O. Nasraoui, P. Shafto, Prcp: Pre-recommendation counter-polarization., in: KDIR, 2018, pp. 280–287.
[15] K. Goldberg, T. Roeder, D. Gupta, C. Perkins, Eigentaste: A constant time collaborative filtering algorithm, Information Retrieval 4 (2001) 133–151.
[16] G. Shani, A. Gunawardana, Evaluating recommendation systems, Recommender systems handbook (2011) 257–297.
[17] P. Castells, S. Vargas, J. Wang, Novelty and diversity metrics for recommender systems: choice, discovery and relevance (2011).
[18] T. T. Nguyen, P.-M. Hui, F. M. Harper, L. Terveen, J. A. Konstan, Exploring the filter bubble: the effect of using recommender systems on content diversity, in: Proceedings of the 23rd international conference on World wide web, ACM, 2014, pp. 677–686.
[19] O. Nasraoui, P. Shafto, Human-algorithm interaction biases in the big data cycle: A markov chain iterated learning framework, arXiv preprint arXiv:1608.07895 (2016).
[20] P. Shafto, O. Nasraoui, Human-recommender systems: From benchmark data to benchmark cognitive models, in: Proceedings of the 10th ACM Conference on Recommender Systems, ACM, 2016, pp. 127–130.
[21] W. Sun, O. Nasraoui, P. Shafto, Evolution and impact of bias in human and machine learning algorithm interaction, Plos one 15 (2020) e0235502.