Fairness-Aware Recommendation System

Juhyeon Nam (20204102, harryjhnam@gm.gist.ac.kr)
GIST AI Graduate School, Integrated Course

December 20, 2020

1 Introduction

Since we are living in the era of information overload, technologies dealing with information, such as information storing and retrieval, have become more necessary. Among them, information retrieval is the most critical technology because it greatly influences the human decision-making process. The most representative example of information retrieval is web search. The information retrieval algorithms were the first ones employed for searching the world wide web[9]. Search algorithms at the time were primarily aimed at fetching accurate information about the searched query. However, today's search algorithms have evolved into a personalized form that displays search results with different rankings or completely different content according to the user's preferences. In terms of personalization, modern search algorithms have the same characteristics as recommendation systems.

Recommendation systems provide suggestions for items to the user, such as Netflix, Google search, and Amazon[7]. The information obtained through the recommendation system has a significant influence on people's decision making. For instance, the probability of a user watching a movie on the top of the list on Netflix's main page is higher than that of other movies at the bottom of the page. The modern recommendation systems determine what people will see, so it needs to be unbiased and reflect our values. However, some cases show it is not. In 2018, a book about racial bias in the Google search engine was published[10]. The author of the book researched how we get information through the internet and found out that the search engine algorithm is not giving neutral results. Also, the searching results can reflect society's bias where the code seems to give neutral results. This is because modern search engines and recommendation systems are based on machine learning, and machine learning is known to fit on socially biased datasets.

Furthermore, there is research about bias amplification in recommender systems[5]. The research pointed out the impact of the feedback loop on the popularity bias amplification. It means that recommendation systems are forced to be biased since it learns from the items consumed by the users, which is also suggested by itself. In addition, they showed that the users' taste also changed according to the bias of the recommendation system. This result suggests that the social bias in recommendation systems may be amplified. Moreover, it means that users may have an impact on that bias.

In this paper, we investigate how user bias is

reinforced by deep learning based recommendation systems. We employ the Neural Collaborative Filtering[3] for the recommendation system and use Jester dataset[2] for a user study. The correlation between recommendation systems' bias and users' bias is the key to improve fairness. Thus, two-level bias measures are needed: (1) bias measures for recommendation systems and (2) amplified bias measures for users. We use TCAV[4] for bias measures in the recommendation system. TCAV is a trained model interpretation tool to measure how user-defined feature was critical in the model's decision. For users' bias measure, we conduct a user study. We compare users' consumed items' matrix by Kullback- Leibler divergence.

2 Related Works

In related works, three major related technologies used in this paper are covered. First, we explain how traditional and modern recommendation systems work. Since modern recommendation systems employ big data and machine learning, we also cover empirical examples about AI learning human bias and how to measure such bias. Lastly, we introduce a recent study that shows how recommendation systems amplify bias.

2.1 Traditional Recommendation Systems

In traditional recommendation systems, there are three steps to generate recommendations: (1) Candidate generations, (2) Scoring Systems, and (3) Re-ranking Systems. First, we have a massive pool of items which user might like or dislike. So, the first step, candidate generation, is making a small subset of candidates to recommend to a user. However, we can use various candidate generators for making candidates. To normalize all these candidates from different generators, we can assign scores to each candidate. These scores are calculated by the scoring systems. Finally, a re-ranking system gives a new rank to the candidates with the scoring systems' scores and other additional constraints.

A candidate generation is the core of recommendation systems. There are generally two types of candidate generation systems: Content-based filtering and collaborative filtering.

2.1.1 Content-based filtering

A content-based filtering system[11] recommends items with a similar feature to the user's previously preferred items. Based on some user's set of positively reacted items, we can induct a user's feature vector representing the user's preference. By comparing the similarity between the user's feature vector and items' feature vector, the content-based filtering recommends items with high similarity. In summary, Content-based filtering suggests to users, "Since you liked this item, you might also like these similar items."

2.1.2 Collaborative filtering

A collaborative filtering system[8] recommends items that other users with similar tastes liked. The system makes users' preferences feature vector. The feature vector is used to find out which user has a similar taste to one another. Finally, the system recommends items that similar users liked, but the user had not seen it. Collaborative filtering suggests to users, "I found other users having similar tastes with you, so you might also like their liked items."

2.2 Deep Learning Based Recommendation Systems

Most modern recommendation systems use deep learning. [12] classified the existing deep learning based recommendation models according to employed deep learning techniques: MLP, Autoencoder, CNNs, RNNs, RBM, NADE, Attention, GAN, and DRL. Among them, MLP is the most basic model that borrows most of the concepts in traditional recommendation systems in a straightforward way.

2.3 AI Learns Human Bias

Bias in the system is not only a problem of the recommendation systems. Various systems using big data and machine learning suffer from the same issue.

In 2018, Amazon employed an AI recruiter[1]. An AI recruiter learns from the previous recruiting data and selects some of the most talented applicants after reviewing applicants' resumes. They were building this system since 2014. However, the project was shut down in 2015 since they realized the system was not scoring applicants in a gender-neutral way. Somehow the system taught itself to give higher scores to male applicants than female applicants in technical positions. The AI system became gender discriminative because it learned male dominance across the tech industry in the past decade. In effect, it penalized resumes, including the word "women's."

Academically, numerous studies have reported that models learn social biases from data[6]. To check whether the model has learned social bias, a tool that can test the model for specific features is needed. TCAV[4] is the method to inspect a trained model with a user-define feature (Testing with Concept Activation Feature). For example, users can see how much the "striped" feature contributes to the model to determine if a given image is "zebra." As an application, we can also measure how much the concept of "male" contributes to the model determining whether a given image is "doctor."

3 Methods

To overcome the limitations of existing studies, as mentioned above, we propose a new experimental design. (1)We use deep learning based recommendation system to interpret bias quantitatively by using TCAV[4]. In addition, (2)we use the "Jester" dataset[2], which enables us to get users' feedback in a user study.

3.1 Recommendation Models

We use Neural Collaborative Filtering[3] for the recommendation model. NCF gets the user's one-hot encoded vector and the item's one-hot encoded vector as inputs. From the embedding layer, the latent vectors of the user and the item are generated. These latent vectors are used to calculate the score of the item recommendation to the user by multi-layer perceptron, the neural CF layer.

3.2 JESTER datasets

We use Jester dataset[2] for experiments. Jester dataset consists of 6.5 million anonymous ratings of jokes by users, rated on a scale of -10 to +10. The dataset can be reviewed in a relatively short amount of time by a user during a user study. However, users will be mostly Korean. Therefore, we eliminate long items with words count over 100 to enhance the readability

for the users. Thirty-seven jokes are eliminated from the dataset, and it affects the number of users with no ratings, increased by 55 users. As a result, 68.24% of ratings are remaining from the original dataset. During the recommendation system training, the NCF model splits the ratings given by the same user to the training and testing datasets. Therefore, the user must have given at least two ratings, so we removed users with 0 or 1 ratings. As a result, 6467 users remain in the dataset.

Additionally, users with only positive ratings are also automatically rejected in the recommendation system training process, and the ratings are binarized into like/dislike ratings. At last, the items are encoded with the sentence transformer to sentence embedding rather than one-hot encoding. We use the pre-trained BERT-base model to encode the items.

3.3 Fairness measure

We use a TCAV for fairness measure(Testing on Concept Activation Vector). TCAV is not the ready-to-use API; it's a comprehensive analysis method for trained models. First, we have a trained recommendation system model to be analyzed and will analyze this model with a concept, "gender."

$$\begin{aligned} S_{c,j,l}(x) &= \nabla h_{l,k}(f_l(x)) \cdot v_c^l \\ &= \lim_{\epsilon \to 0} \frac{h_{l,j}(f_l(x) + \epsilon v_c^l) - h_{l,j}(f_l(x))}{\epsilon} \end{aligned} \tag{1}$$

Then we can measure the model's sensitiveness about the given concept by the equation 1. $f_l(x)$ is the bottleneck vector from the lth NeuMF layer of the model, where x is the input. Then collect these bottlenecks where the input words are related to the given concept C, or not. Then a linear classifier is trained to classify concept related words against non-related words. The concept activation vector v_c^l is the normal vector of the decision plane of the linear classifier. With this concept activation vector, we can get a directional derivative of h: a mapping from the bottleneck vector to the prediction score. Then we can get the sensitiveness score $S_{c,j,l}$ of the model's prediction with given input x.

$$TCAV_{Q_{c,j,l}} = \frac{\{x \in X_j : S_{c,j,l}(x) > 0\}}{|X_j|} \tag{2}$$

The original paper[4] calculate the TCAV score as the ratio of positive $S_{c,j,l}$ over the number of all inputs as the equation 2 shows.

$$TCAV_{c,j,l} = \frac{\sum_{x \in X_j} S_{c,j,l}(x)}{|X_j|} \tag{3}$$

However, we modified the measure to the mean of $S_{c,j,l}$ among the ratings to the given item, shown in equation 3. Therefore we can measure the gender-sensitiveness of a single item.

3.4 Fairness-aware recommendations

The fairness-aware recommendation is generated by sorting the recommendations in reverse order according to the TCAV score. Detailed procedure of generating fairness-aware recommended items is consists of three steps: (1) Cache twice the number of items required, (2) sort the items in reverse by TCAV score, and (3) recommend the top items from the sorted items.

4 User study

The user study consists of two stages: (1) preliminary user study and (2) main user study. First, users received ten items randomly selected from the dataset, and users suppose to give binary ratings(Like/Dislike) on items. Then the ratings are added to the JESTER datasets to train the recommendation system. Users will receive ten items recommended from the trained recommendation system, which now learned each user's preference. At this point, users in the control group get the original items recommended from the recommendation system. On the other hand, the experiment group gets the fairness-aware recommended items modified from the original items based on their TCAV scores.

We conduct a user study with four users. Two of them are assigned to the control group, and the others are assigned to the experiment group. We use two criteria to measure the fairness-aware recommendation system: (1) ratio of user's positive ratings about recommended items, and (2) TCAV measure to the recommendation system about a particular fairness-aware concept. We tested our model on TCAV with the concept "gender."

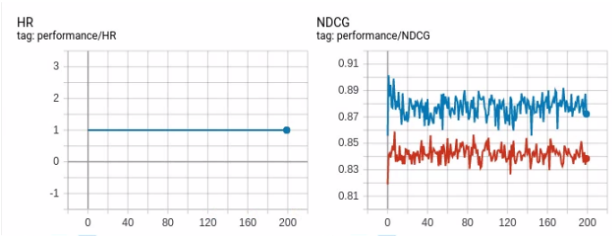

Figure 1: Training results of recommendation system with user study ratings. Blue line is the recommendation system with pre-trained BERT-base encoding and the red line is with the BERT-large encoding.

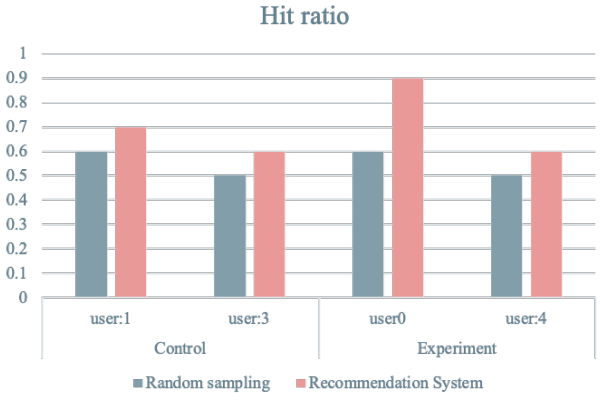

Figure 2: Recommendation system's performance measure during the user study with hit ratio.

5 Results

The recommendation system achieved the best performance with perfect hit ratio and the 0.9 NDCG.

The user study results are shown in figure 2. The first four columns are the hit ratio of the control group, and the last four columns are the hit ratio of the experiment group. Gray bar is the hit ratio with random sampling, the results from stage 1 in the user study. And the red bar is the hit ratio with the trained recommendation system from stage 2. The recommendation system consistently improves the quality of recommendations than the randomly selected items. The trained recommendation system improves the mean hit rate by 0.1, while the fairness-aware recommendation system improves it by 0.2.

	Original recommendation system (Control)	Fairness-aware recommendation system (Experiment)
Mean TCAV scores of recommended items	-1.3692 E+10	-2.2848 E+10

Table 1: Mean TCAV scores of the original recommendation system and the fairness-aware recommendation system.

The table 1 shows that the fairness-aware recommendation system haw lower TCAV score than the original recommendation system. It means that the fairness-aware recommendation system improves the fairness of the recommended items.

6 Conclusion

In this paper, we show that the TCAV can measure the fairness of text items in the user-defined way. Moreover, the proposed fairness-aware recommendation system improves the fairness of recommended items while preserving the recommendation system's performance. However, the number of users in the user study was relatively small. We need to conduct more user studies with a larger number of users and various concepts other than the "gender".

References

[1] J. Dastin. (2018) Amazon scraps secret ai recruiting tool that showed bias against women. [Online]. Available: https://www.reuters.com/article/us-amazon-com-jobs-automation-insight/amazon-scraps-secret-ai-recruiting-tool-that-showed-bias-against-women-idUSKCN1MK08G

[2] K. Goldberg, T. Roeder, D. Gupta, and C. Perkins, "Eigentaste: A constant time collaborative filtering algorithm," *information retrieval*, vol. 4, no. 2, pp. 133–151, 2001.

[3] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proceedings of the 26th international conference on world wide web*, 2017, pp. 173–182.

[4] B. Kim, M. Wattenberg, J. Gilmer, C. Cai, J. Wexler, F. Viegas, *et al.*, "Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav)," in *International conference on machine learning*. PMLR, 2018, pp. 2668–2677.

[5] M. Mansoury, H. Abdollahpouri, M. Pechenizkiy, B. Mobasher, and R. Burke, "Feedback loop and bias amplification in recommender systems," in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 2020, pp. 2145–2148.

[6] E. Ntoutsi, P. Fafalios, U. Gadiraju, V. Iosifidis, W. Nejdl, M.-E. Vidal, S. Ruggieri, F. Turini, S. Papadopoulos, E. Krasanakis, *et al.*, "Bias in data-driven artificial intelligence systems—an introductory survey," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 10, no. 3, p. e1356, 2020.

[7] F. Ricci, L. Rokach, and B. Shapira, "Introduction to recommender systems hand-

book," in *Recommender systems handbook*. Springer, 2011, pp. 1–35.

[8] U. Shardanand and P. Maes, "Social information filtering: algorithms for automating "word of mouth"," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, 1995, pp. 210–217.

[9] A. Singhal *et al.*, "Modern information retrieval: A brief overview," *IEEE Data Eng. Bull.*, vol. 24, no. 4, pp. 35–43, 2001.

[10] J. Snow. (2018) Bias already exists in search engine results, and it's only going to get worse. [Online]. Available: https://www.technologyreview.com/2018/02/26/3299/meet-the-woman-who-searches-out-search-engines-bias-against-women-and-minorities/

[11] R. Van Meteren and M. Van Someren, "Using content-based filtering for recommendation," in *Proceedings of the Machine Learning in the New Information Age: MLnet/ECML2000 Workshop*, vol. 30, 2000, pp. 47–56.

[12] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep learning based recommender system: A survey and new perspectives," *ACM Computing Surveys (CSUR)*, vol. 52, no. 1, pp. 1–38, 2019.