**Study of Recommender Systems, their drawbacks and ways to mitigate them**

**Abstract**

Recommender Systems are a class of problems where the main objective of the problem is to prompt or recommend items or objects to users. Ecommerce websites like Amazon, Flipkart and Big bazar recommends or prompts the users to buy certain items whereas websites like YouTube, Netflix and Amazon Prime show recommended playlists. The revenue of these companies as well as the personalized experience that the users have on visiting these websites depend highly on how well the recommendations are made. Inspite of all these dependencies, the bias that already exist in the available data gets sculpted and magnified and are returned to users as biased recommendations.

Similarly, if a head hunter searches for candidates for a particular position and the ranking algorithm puts someone (XYZ) at the bottom of the list, then that person (XYZ) becomes less visible to recruiters and therefore gets less job offers resulting in less experience and less employability. Hence such unfair ranking algorithms can put people into a vicious loop.

The above two problems cater to the same need of mitigating bias (gender, ethnicity etc.) in recommender/ranking algorithms. Here we try to understand both these algorithms and look into what each one does differently.

**Introduction**

Recommender Systems are used for all sorts of products, fashion, music, videos, movies, books etc. They are omnipresent. Recommender systems can have different modelling approaches such as:

* Collaborative Filtering – Recommend items to users that other users similar to him/her liked in the past.
* Content Based Filtering – Recommend content to users similar to the ones he/she liked in the past.
* Matrix Factorization – Both Users and items are related in some underlying taste which can be uncovered by matrix factorization.
* Hybrid RS – A combination of any of the above.

**Basic Principles of Recommender Systems**

All Recommender systems learn from the preference and ratings of items given by the users. This personalization influences the decisions of users to choose a particular item.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Users | Batman | Superman | Spiderman | Matrix | Titanic |
| Annie | 2.5 | 2 | 3 | 3 | 3.5 |
| Marsh | 3.5 | 3 | 4 | 4.5 | N/A |
| Tim | 2 | 2.5 | 2 | N/A | N/A |
| Steffanie | N/A | 2 | 3 | N/A | N/A |

Ratings given by users can be explicit, that is giving ratings to the items bought, or can be implicit, for example browsing time or the no of clicks of a particular item.

In Collaborative filtering

**Input** : The user-rating matrix that’s shown in fig 1. This matrix is sparse.

**Output**: For a particular user, that row has to be completed

**User Based** - If a user-U likes an item-I, then recommend item-I that was liked by other users like him/her

**Item Based** – If a user-U likes an item-I, then recommend item-J that is similar to item-I

**Experiment with Matrix Factorization on movie-lens dataset:**

* Tested the matrix factorization method on movie-lens small dataset with no of latent features = 20
* Also tried running SVD, KNNMeans and SVDpp(SVDplusplus) on the same
* SVDpp gave the best result with the lowest RMSE
* Though SVDpp gave the best result the time taken was quite a lot
* So tried SVD with crossvalidation for optimum parameters as the difference in rmse of SVD and SVDpp were not very high.

**Comparison of the algorithms**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Algorithm** | **Characteristic** | **RMSE** | **Fit time** | **Test time** |
| KNNWithMeans | Means of the ratings | 0.967581 | 1.630453 | 5.223192 |
| SVD | Explicit ratings | 0.880156 | 17.616324 | 0.960677 |
| SVDpp | Implicit ratings considered | 0.868648 | 1737.273370 | 41.724478 |

**Major drawbacks of Recommender Systems:**

* First one being **data privacy of users**. User’s rating data need to be collected in a server for model training. Keeping in mind user privacy and security, federated recommender system and blockchain recommender systems are being studied upon. In Federated recommender systems with a distributed model, each user’s data is not shared with other users or the server.

Few papers related to them are as follows:

***Federated Recommender Systems:***

[***https://ising.cse.ust.hk/files/Federated\_Recommendation\_Systems.pdf***](https://ising.cse.ust.hk/files/Federated_Recommendation_Systems.pdf)

***FedRec++:***

[***https://ojs.aaai.org/index.php/AAAI/article/view/16546***](https://ojs.aaai.org/index.php/AAAI/article/view/16546)

***BlockChain In Recommender Systems:***

[***https://www.sciencedirect.com/science/article/pii/S1574013721000769***](https://www.sciencedirect.com/science/article/pii/S1574013721000769)

* Second being **Gender Bias** in Recommender systems. Many experiments are being done to mitigate gender bias in Recommender systems.

One of the papers related to mitigating gender bias in Recommender systems is as follows :

***Exploring and mitigating gender bias in Recommender systems with Explicit feedback :***

[***https://arxiv.org/pdf/2112.02530.pdf***](https://arxiv.org/pdf/2112.02530.pdf)

* Third being **unfair ranking** in search leading to unfair ranking in recommendations.

One of the papers ***FA\*IR: A Fair Top-k Ranking Algorithm*** deals with fair ranking in search

[***https://arxiv.org/pdf/1706.06368.pdf***](https://arxiv.org/pdf/1706.06368.pdf)

**Federated Recommender Systems:**

To protect the privacy and security of user data in recommender systems, a data decentralized architecture of Federated Recommender Systems has been adopted. Here users keep their data locally and updated parameters/knowledge are allowed to be shared between users. Hence the data privacy is maintained. Federated Recommender Systems are categorized into Horizontal FedRec, Vertical FedRec and Transfer FedRec.

Horizontal FedRec : Here the items are shared, but users are different between parties.

Vertical FedRec: Here Users are shared but items are different between parties.

Transfer FedRec : Here neither the users nor the items are shared between parties.

Federated recommendation methods either bias the model training or do not protect the users’ rating behaviors well.

Some works on Federated RS choose to sample out some items for each user which are not rated and give certain virtual ratings to them, so that the server cannot identify the scores and the set of rated items. But, these virtual ratings definitely introduce some noise to the model training process, which then cause loss in the recommendation performance.

**FedRec++:**

This is a novel lossless federated recommendation method (FedRec++) wherein some denoising clients are allocated(i.e., users) to eliminate the noise in a privacy-aware manner.

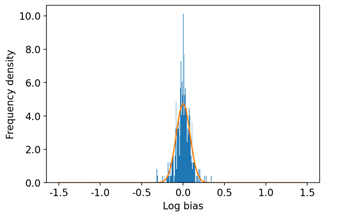
* Firstly, each user’s original rating records are always kept locally in the client which ensures the security of the original data.
* Secondly, FedRec++ assigns a virtual rating to each randomly sampled unrated item, which protects the users’ rating behaviors.
* Thirdly, each client transfers the gradients of the sampled unrated items to a denoising client which again does not expose the user’s rating behaviors.
* Fourthly, as the gradient does not contain the userID, the denoising client cannot identify the sender of the gradient, and hence the anonymity is maintained
* Finally, even if the server intends to collaborate with the denoising clients, the server cannot obtain the rating behaviors because of the anonymity of the clients.

**Experiment with Mitigating Gender Bias In Recommender Systems on Book Crossing and Amazon Book Review dataset:**

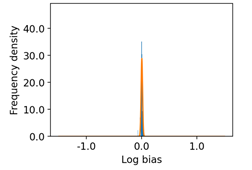
**Model :** This study is to eliminate bias against items that are dealt with a particular gender. There are pre-processing, in-processing and post-processing methods for bias prevention and this study deals with a hybrid model of pre and post processing methods.

This model mitigates biases by initially finding the log bias of each individual and then debiasing an individual’s ratings. This debiasing ensures that the exising bias doesn’t propagate any further in the system. These ratings are then fed to the recommender algorithms to produce the recommendations. Although these debiased ratings assure that the biases of an individual do not affect another, there is a possibility of accuracy loss in the process. A new step called preference correction is introduced here such that an individual’s preference is injected into the debiased recommendation to maintain the accuracy.

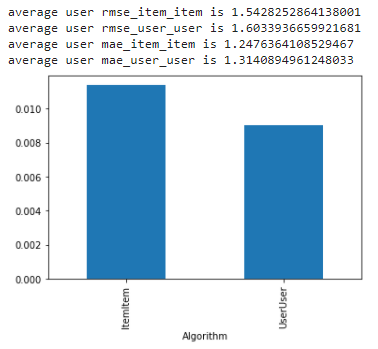
This model thus reduces bias while not deteriorating the accuracy.



Input Log Bias in the Book Crossing dataset



Output Log Bias for Book Crossing dataset



Book Crossing Dataset

**FA\*IR Top-k Ranking Algorithm**

Unfair ranking can put a candidate at the bottom of a search list. That particular candidate thus becomes less visible to recruiters, gets less job offers, becomes less experienced and becomes less employable. This loop becomes really vicious for candidates. Similarly when searching for the top 40 and top 10 candidates the proportion of protected and non protected groups should ideally be the same. But that doesn’t happen if the ranking is unfair.

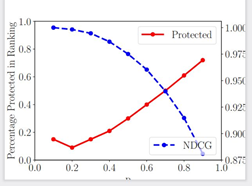
This FAIR ranking algorithm is a method to ensure fairness with a minimum loss of utility. *Individuals are ranked decreasingly by their utility, and the ranked list is built incrementally. At each position, the individual with the highest utility is inserted, among those that would not violate the fairness constraint*

A FAIR ranking is one in which the number of protected candidates do not fall below a given minimum proportion at any given time in the ranking and the utility is maintained as well.

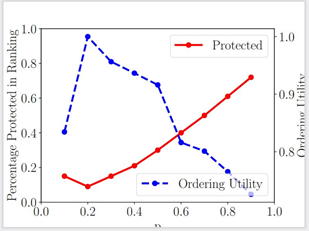
The two types of utility that needs to be maintained are selection and ordering utility.

Selection Utility – This utility is one in which every candidate who are present in the top k are more qualified than the ones who are not present in the top-k

Ordering Utility – In this utility for every pair of candidates in the top-k the more qualified candidate is ranked above the other one.



Protected Vs NDCG



Protected Vs Ordering Utility

**Conclusion and future work:** Despite the drawbacks in Recommender systems they are the most commonly used algorithms for prompting users to select certain items based on their preferences. The Gender Bias model proposes to mitigate the bias with regard to gender in recommender systems keeping the accuracy intact. The FA\*IR algorithm proposes to do a fair ranking given the length of the ranking, the two groups of protected and non-protected items and the minimum proportion of people to be achieved.

The datasets used in FA-IR algorithm are:

* Sat - *\*Scholastic Assessment Test\**

(<https://secure-media.collegeboard.org/digitalServices/pdf/sat/sat-percentile-ranks->composite-crit-reading-math-writing-2014.pdf): a standardized test used in the US for university admissions

* Compass - *\*Correctional Offender Management Profiling for Alternative Sanctions\**

(https://github.com/propublica/compas-analysis): a survey used in some US states for alternative sanctions such as parole

* Germancredit – German Credit Scores

https://archive.ics.uci.edu/ml/datasets/Statlog+(German+Credit+Data)) (SCHUFA) dataset

* Xing - A dataset collected from the professional social network Xing in Jan/Feb 2017 |

These datasets can be used in the Gender Bias model to compare both these algorithms in terms of utility loss and accuracy.

References: [***https://ising.cse.ust.hk/files/Federated\_Recommendation\_Systems.pdf***](https://ising.cse.ust.hk/files/Federated_Recommendation_Systems.pdf)

[***https://ojs.aaai.org/index.php/AAAI/article/view/16546***](https://ojs.aaai.org/index.php/AAAI/article/view/16546)

[***https://www.sciencedirect.com/science/article/pii/S1574013721000769***](https://www.sciencedirect.com/science/article/pii/S1574013721000769)

[***https://arxiv.org/pdf/2112.02530.pdf***](https://arxiv.org/pdf/2112.02530.pdf)

[***https://arxiv.org/pdf/1706.06368.pdf***](https://arxiv.org/pdf/1706.06368.pdf)