Bias in Recommender Systems Cheat Sheet

Part 1 - Introduction

Recommendation is an AI task that has many differences from traditional tasks (classification, regression). In the recommendation setting, we have **users** and **items**, and the goal is to recommend new items to each user. We do so by using past **ratings** of items by users (known as **interactions**).

We will often have the data (ratings) displayed within an **interaction matrix**. This matrix contains the rating given by a user to an item, where relevant, if there are no ratings we often use NaN or 0. An example with 5 users, 6 items and ratings from 1 to 5 is shown below.

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 4 & 3 \\ 0 & 0 & 2 & 0 & 4 & 3 \\ 0 & 1 & 2 & 5 & 4 & 3 \\ 0 & 0 & 2 & 0 & 1 & 3 \\ 5 & 0 & 0 & 3 & 4 & 5 \end{bmatrix}$$

The goal in recommendation is usually to first **predict** unknown ratings (those that are 0 in above matrix). These predicted ratings are then used to recommend new items to users, choosing those that have a predicted high rating.

Part 2 - Notions of Fairness

Taxonomy

- User Fairness
  - Group User Fairness
  - Individual User Fairness
- Item Fairness
  - Provider Item Fairness
  - Individual Item Fairness

Group User Fairness attempts to equalize outcomes for subgroups of users (e.g., male / female).

Indivisual User Fairness attempts to ensure that similar users get similar items.

Provider Item Fairness attempts to equalize outecomes of different Providers of items (e.g., Kellogs / Nestle).

Individual Item Fairness attempts to reduce differences in exposures of different items.

Part 3 - Measuring Bias

A very useful notion used in many bias metrics is the **exposure distribution**. The exposure distribution can be computed for subgroups or the overall population. It is a vector over items with each entry being the proportion of recommendations coming from this item. The exposure distribution of a group A can be computed as $q_A = N_A(i)/N_A$. Where $N_A(i)$ is the number of times item i is recommended to group A and N_A is the total number of recommendations to group A.

For Group User Fairness, it is common to measure bias by comparing the exposure distribution of two groups, known as **exposure bias**. If A and B are two subgroups, and q_A, q_B are their respective exposure distributions, we can create metrics by looking at the distance between these. E.g. The total variation distance or the KL divergence.

For individual item fairness, we often compute the total exposure distribution, and measure it's inequality known as **popularity bias**. If the total exposure distribution is q, we often will plot it (the long tail plot) to get an idea of the inequality between popular and niche items. E.g. The Gini Index.

See following for more metrics and references. [1] Deldjoo et al, 2022, A Survey of Research on Fair Recommender Systems.

The **holisticai library** can help in measuring bias. To install the library in a notebook, run the following line of code.

```
In [ ]: %pip install holisticai
```

Once the library is installed, the documentation can be found here (https://holisticai.readthedocs.io/en/latest/). To import all bias metrics, one can run the following line.

In [ ]:

```


from holisticai.bias.metrics import *
```

Part 4 - Mitigating Bias

Taxonomy

- Pre-processing
- In-processing
- Post-processing

Pre-processing refers to techniques where mitigation is applied to the data.

In-processing refers to techniques where the mitigation is included in the model and it's training.

Post-processing refers to techniques where the mitigation happens after training, directly on the outcomes.

Collaborative filtering is a common approach to recommendation using the heuristic that similar users will like similar items. A popular collaborative filtering approach to recommendation is **matrix factorization**.

Whithin the slides, we give an overview of a individual item bias mitigation strategy ([2]) and a group user bias mitigation strategy ([3]), both of which are based on changing the matrix factorization objective.

[2] Sun et al, 2019, Debiasing the Human-Recommender System Feedback Loop in Collaborative Filtering

[3] Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, Jun Sakuma, 2018, Recommendation Independence

The **holisticai library** can help in mitigating bias. To install the library in a notebook, run the following line of code.

In [ ]:

```
%pip install holisticai
```

Once the library is installed, the documentation can be found here (https://holisticai.readthedocs.io/en/latest/). To import all bias mitigation strategies, one can run the following line.

In [ ]:

```
from holisticai.bias.mitigation import *
```