

# FaiRIR: Mitigating Exposure Bias from Related Item Recommendations in Two-Sided Platforms

Abhisek Dash, Abhijnan Chakraborty, Saptarshi Ghosh, Animesh Mukherjee, and Krishna P. Gummadi

**Abstract**—Related Item Recommendations (RIRs) are ubiquitous in most online platforms today, including e-commerce and content streaming sites. These recommendations not only help users compare items related to a given item, but also play a major role in bringing traffic to individual items, thus deciding the exposure that different items receive. With a growing number of people depending on such platforms to earn their livelihood, it is important to understand whether different items are receiving their desired exposure. To this end, our experiments on multiple real-world RIR datasets reveal that the existing RIR algorithms often result in very skewed exposure distribution of items, and the quality of items is not a plausible explanation for such skew in exposure. To mitigate this exposure bias, we introduce multiple flexible interventions (*FaiRIR*) in the RIR pipeline. We instantiate these mechanisms with two well-known algorithms for constructing related item recommendations – rating-SVD and item2vec – and show on real-world data that our mechanisms allow for a fine-grained control on the exposure distribution, often at a small or no cost in terms of recommendation quality, measured in terms of relatedness and user satisfaction.

## I. INTRODUCTION

Recommendations are major drivers of traffic (and revenue) on two-sided market platforms, including e-commerce sites like Amazon or Flipkart, and multimedia sites like YouTube, Spotify or Netflix [1]–[3]. There are two primary stakeholders in two-sided platforms: (1) **producers** (or sellers) of items (goods, contents or services) listed on the platforms, and (2) their **consumers** (or users). A few recent works have focused on the *fairness of recommendations* in such platforms, but mostly from the perspective of consumers [4]–[7], such as, whether different groups of consumers experience similar quality of recommendations.

However, as the recommendations help consumers efficiently explore the item space, they also implicitly determine the amount of *exposure* different items get, affecting the revenues of their producers. For example, 30% of Amazon’s traffic originates from recommendations [8]. Similarly, 80% of movies watched on Netflix are driven by recommendations [1]. With a large number of businesses and individuals depending

This research is supported in part by (1) a grant from the Max Planck Society through a Max Planck Partner Group at IIT Kharagpur, and (2) a European Research Council (ERC) Advanced Grant for the project “Foundations for Fair Social Computing”, funded under the European Union’s Horizon 2020 Framework Programme (grant agreement no. 789373). Additionally, A. Dash is supported by a fellowship from Tata Consultancy Services.

A. Dash, S. Ghosh and A. Mukherjee are with Indian Institute of Technology Kharagpur, Kharagpur 721302, India (e-mail: dash.abhi93@iitkgp.ac.in; saptarshi@cse.iitkgp.ac.in; animeshm@cse.iitkgp.ac.in).

A. Chakraborty is with Indian Institute of Technology Delhi, Delhi 110016, India. (e-mail: abhijnan@iitd.ac.in)

K. P. Gummadi is with the Max Planck Institute for Software Systems, 66123 Saarbrücken, Germany. (e-mail: gummadi@mpi-sws.org)

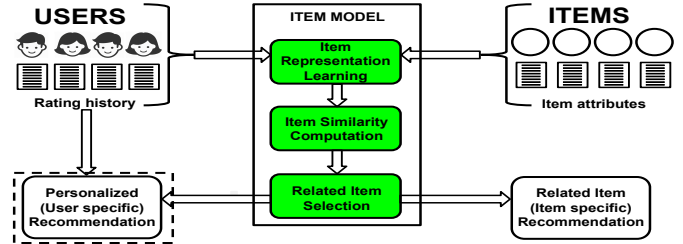


Fig. 1: A generic block diagram explaining item-based recommendation methods. Item models generated in these methods can be utilized for both related item recommendations and personalized recommendations. The latter, enclosed in broken rectangles in the diagram, is beyond the scope of the current work.

on two-sided platforms to earn their livelihood [9], there are growing concerns about the *fairness of these recommendations from the perspective of the producers or sellers* [4], [6], [10]. Even on the legislation front, a recent Indian regulation mandates e-commerce sites to treat their sellers fairly [11]. In this paper, we focus on the producer-side fairness considerations raised by recommendations.

Recommendations in two-sided platforms are primarily of two types (see Figure 1): (i) item-specific **Related Item Recommendations (RIR)**, e.g., ‘customers who viewed this item also viewed the following items’ recommendations on Amazon, or ‘Up next’ video recommendations on YouTube, and (ii) user-specific **Personalized Recommendations**, e.g., ‘Related to items you’ve viewed’, ‘Inspired by your shopping trends’ recommendations on Amazon, ‘Because you watched X’ on Netflix. Notice while personalized recommendations are centered around the past interactions of a specific customer (to whom the recommendations will be shown), related item recommendations are centered in the context of a particular item [12]. The underlying notion of relatedness can be of different types which will be discussed in detail in Section III. Two recent works lately have looked at fairness for the producers [10], [13], but they only consider the user-specific personalized recommendations. To our knowledge, our work here is the first to investigate fairness issues in Related Item Recommendations (RIRs).

**Item exposure bias in RIRs:** As RIR algorithms recommend new items that are ‘related’ or ‘similar’ to the item currently being viewed by a user, there may arise situations where an item gets much more (or less) exposure than what it deserves. For example, a poor-quality item may be recommended as related from a popular good-quality item (say, by virtue of being produced by the same manufacturer) and hence the

poor-quality item may end up getting much more exposure than it deserves. On the other hand, a good-quality item may fail to get the desired exposure, simply because it is not recommended as directly related to other popular items by a RIR algorithm. In fact, our investigation over real-world datasets (Section IV) shows that the relative exposure of items that would be induced by state-of-the-art RIR algorithms is often uncorrelated or disproportionate to the relative quality of the items. We term this discrepancy between the *observed* item exposure (as induced by RIRs) and the *desired* item exposure (e.g., based on item quality) as *exposure bias*.

In this paper, we posit that by solely focusing on ‘relatedness’ between items, RIRs may implicitly bias the exposure distribution of items in a manner that does not reflect a desired producer-fair exposure distribution.<sup>1</sup> We propose mechanisms to quantify and mitigate the exposure bias of RIRs. To formalise the concept of exposure bias, apart from the ‘observed’ exposure of an item, we also need to have a notion of what is the ‘desired’ exposure of that item. Since the notion of desired exposure is highly contextual, it cannot be riveted to a single operational definition. Therefore, we operationalize this notion of ‘desired’ exposure in multiple ways allowing for multiple contextual assumptions. In fact, we develop the rest of our pipeline in such a way that *any new operationalization of the ‘desired’ exposure can be seamlessly plugged in*. We show that our proposed mechanisms can improve producer-side fairness of RIRs, with little or no impact on the utility of the recommendations to consumers.

**Contributions:** We make the following contributions.

- We demonstrate the exposure bias induced due to two popular RIR algorithms – rating-SVD and item2vec – on two real-world datasets – the MovieLens and Amazon product review datasets [15], [16]. The choice of these two datasets is motivated by the two large businesses they represent – the entertainment industry and the e-commerce industry.
- To counter the exposure bias, we propose **FaiRIR**, a novel suit of three algorithms applied at different stages in the RIR pipeline, that can minimize exposure bias while preserving the underlying relatedness of the recommended items to the best possible extent.
  - FaiRIR<sub>rl</sub>, based on *fair representation learning*
  - FaiRIR<sub>sim</sub>, based on *fair similarity computation*
  - FaiRIR<sub>nbr</sub>, based on *fair neighbor selection*
- Extensive offline evaluations on the real-world datasets show that FaiRIR can significantly reduce exposure bias, while preserving the underlying relatedness and utility of the recommendations toward the end-user.
- Finally, we conducted a user survey on Amazon Mechanical Turk, which further demonstrates the efficacy and utility of the proposed FaiRIR algorithms.

We believe that the methodologies discussed in this paper are generic enough to be extended to different setups where related item recommendation algorithms are deployed. For better understandability and reproducibility, we have released

our source codes at: <https://github.com/ad93/FaiRIR>.

## II. BACKGROUND AND RELATED WORK

We review prior works on related item recommendations, followed by the recent works on algorithmic fairness especially in the domain of recommendation systems.

**Related item recommendations:** RIR systems recommend items for a source item (typically, the item a user is consuming at a point in time) based on their relatedness to the source. Evaluating relatedness between a pair of items is central to item-based collaborative filtering recommendations [17], [18]. Multiple prior works have proposed different approaches for identifying relatedness. For example, some approaches use user-rating matrices [17], [19], which may be factorized using matrix decomposition techniques like SVD [20]. Other approaches rely on implicit information such as clicks or co-purchases [21], [22].

RIR algorithms are used in multiple two-sided platforms, with some platform-specific enhancements. For instance, Amazon uses RIRs in its product pages [2], [18], YouTube recommends videos through their ‘Up next’ recommendations [23], Netflix uses RIR for recommending movies [1]. In this work, we attempt to cover different types of relatedness considering each of the above categories, by examining user-item ratings (rating-SVD), and users’ activity information (item2vec).

**Algorithmic bias and fairness:** Recently, there have been extensive research on algorithmic bias and fairness across multiple disciplines [24]. To tackle any inadvertent consequences of algorithmic decisions, many recent studies have considered fairness from mainly two perspectives: (1) *individual fairness*, which requires similarly deserving candidates should be treated similarly [25], [26], and (2) *group fairness*, requiring different social salient groups should be treated similarly [27], [28]. While some studies focus on detecting the discrimination (e.g., [27], [29]), others suggest mitigation strategies by proposing fairness-aware algorithms (e.g., [26], [30], [31]).

Few recent works have considered group and individual fairness in *personalized recommendations* [4], [6], [7], [10], [13] where the goal is to ensure that the recommendations do not discriminate against socially salient groups or individuals. To our knowledge, ours is the first attempt to consider *individual fairness in related item recommendations*, where we propose a novel algorithm *FaiRIR*, that attempts to provide the desired level of exposure to different items.

## III. RIR SYSTEMS AND EXPOSURE THEREOF

In this section, we present the notion of relatedness between items and how we instantiate an item model capturing it. We also demonstrate the operationalization of exposure induced by an RIR algorithm with respect to the discussed instantiation.

### A. Relatedness of recommended items

The primary goal of related item recommendations is to maximize the *relatedness of recommended items* to the source item that the consumer has viewed/purchased/liked. Though there is no sacrosanct definition of relatedness, two items can

<sup>1</sup>Note that, exposure bias may get induced due to multiple explicit factors too, such as special relationship of certain items with the platform [14]; however such concerns are beyond the scope of the current work.

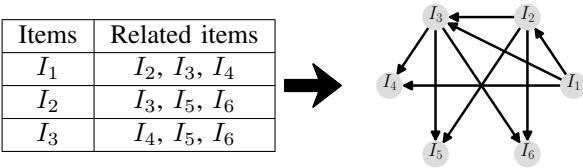


Fig. 2: A sample item model and its corresponding RIN.

be thought of as related over multiple dimensions:

(i) **Content based relatedness**, e.g., movies of the same genre, items from the same producer or brand, etc.,

(ii) **Compatibility**: two items can be related if they are either the substitute or complement of one another, e.g., items that are frequently purchased together: a smartphone and its cover,

(iii) **External feedback on recommendation platforms**: user-actions such as likes and ratings, also define relatedness. For example, items being rated similarly, liked or disliked by a number of common users can be considered as related.

Relatedness, therefore, is subjective, and RIRs are judged based on whether the consumers find the source and the recommended items to be related. Additionally, the metric to measure relatedness between items is often domain-dependent.

The concept of ‘relatedness’ is analogous to ‘accuracy’ or ‘relevance’ in the context of a related item recommendation system – just like classifiers are traditionally designed to optimize for accuracy, RIR systems are traditionally designed to optimize (maximize) relatedness.

### B. Instantiating Item Model by Related Item Network

As shown in Figure 1, both Related Item Recommendations and Personalized Recommendation systems utilize an *Item Model* that captures the relatedness among items. We now discuss an intuitive way to instantiate the Item Model, which was developed in our prior works [14], [32].

We utilize an instantiation of the Item Model of a recommendation system as a *Related Item Network* (RIN). A RIN is a directed network, with each node being analogous to an item in the universe, and a directed edge between two nodes implies that the corresponding source and destination items are related (based on some underlying notion of relatedness). For instance, let us consider an item model as shown in the table in Figure 2, and its corresponding RIN. Since item ‘ $I_2$ ’ is related to item ‘ $I_1$ ’, the corresponding nodes in the RIN are connected via a directed edge (from ‘ $I_1$ ’ to ‘ $I_2$ ’).

Once this instantiation of item model is constructed, a simple way to generate the Related Item Recommendations is as follows. For a particular source item, one can recommend those items to which it links in the RIN. For instance, in Figure 2, the recommendations for source item ‘ $I_1$ ’ are items ‘ $I_2$ ’, ‘ $I_3$ ’, and ‘ $I_4$ ’.

### C. Estimating observed exposure

We define the observed exposure  $E_o(i)$  of an item  $i$  as the exposure it actually gets after the deployment of a RIR algorithm. Ideally, the observed exposure of items should be quantified by click-through rates or other user interaction signals. However, the availability of such comprehensive user-item interactions is seldom possible for third-party researchers

due to the sensitive nature of the information. Counting the number of recommendations received by items (analogous to in-degree of items in RIN) may be a possible work-around in such situations. However, the importance of all recommendations is *not* the same – it varies with the source item, e.g., a recommendation from a popular source item is expected to yield more visibility (for the destination item) than a recommendation from a non-popular item.

Taking such observations into consideration, we use the ‘Random Surfer model’ [33] to estimate the observed exposure. In general, users tend to visit the page of an item and then they start exploring different items recommended on the page. Alternatively, they can also randomly consume any other item thereafter. By simulating such user exploration for a large number of iterations, we take the *steady state visit frequency of a node  $i$*  as its observed exposure  $E_o(i)$  (more details can be found in our prior work [14]). Note that the notion of observed exposure of an item is very similar to PageRank of the corresponding node, in this formulation.

**Considerations during Random Surfer simulation:** While simulating user browsing behavior, we note that different users can have different propensity to follow recommendations. The ‘teleportation probability’  $\alpha \in [0, 1]$  of the Random Surfer model captures such considerations. The surfer chooses to traverse the recommended items with probability  $(1 - \alpha)$ , and teleport to a random item with probability  $\alpha$ . Throughout the paper, we report results for  $(\alpha = 0.15)$  which is the most prevalent value of teleportation in the literature [14], [34]. Finally, we normalize the observed exposure scores of all items such that  $\sum_{i \in \mathbf{I}} E_o(i) = 1$ .

## IV. LIMITATIONS OF EXISTING RIRs

Figure 1 shows a schematic block diagram of the methods involved in building a model of *item-item relatedness*, (also called an *item model*), which is then utilized to make the recommendations at scale [2], [35]. Item models capture the similarity between items based on either user-item interaction logs and/or item attributes [17], [19], [36]. From these logs, first a latent space representation of each item is obtained. Then, the similarity between pairs of items are computed from the latent representations, and  $k$  most similar (related) items to a given item are generated. In case of RIRs, this item model is used to recommend items that are ‘related’ or ‘similar’ to an item. Next, we explore two popular RIR algorithms, and check whether their recommendations induce disproportionate exposure to items.

### A. Two popular RIR algorithms

As a proof of concept, we consider the following two popular algorithms for generating related item recommendations – (i) rating-SVD, and (ii) item2vec (detailed below). We choose these two algorithms because they cover some of the most common techniques for RIRs [12]. These algorithms take as input a *user-item rating matrix*  $M$  whose  $(i, j)$ -th entry gives the rating that the user  $i$  gave to the item  $j$ . First, these algorithms learn a latent space representation of different items (from  $M$ ). To generate recommendations for a seed item  $i$ , the

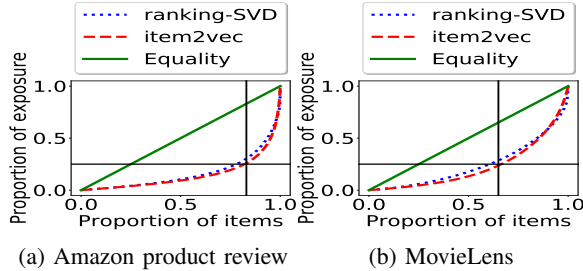


Fig. 3: Lorenz plot showing the skew in observed exposure of items due to two state-of-the-art RIR algorithms over two real world recommendation datasets.

algorithms then generate top- $k$  neighbours of  $i$ , by ranking items based on their similarity with  $i$  in the latent space.

- **rating-SVD** applies Singular Value Decomposition (SVD) [20] to the user-item rating matrix  $M$ , and uses cosine-similarity for similarity evaluation [19]. The underlying notion of relatedness between two items  $i$  and  $j$  can be described as ‘people who liked item  $i$  are likely to like item  $j$ ’. Following the setup in [12], we implement SVD with 128 dimensions. As a pre-processing step, we perform mean subtraction on the input and normalise each row of the final representations to a unit vector.

- **item2vec** [21] is a replication of word2vec [37] representation learning. item2vec substitutes items for words, and tries to find out co-occurrence patterns in user consumption. The underlying notion of relatedness among items can be defined as ‘people who consumed item  $i$  are likely to consume item  $j$  in close temporal proximity’. Following the setup in [12], we train the algorithm for 100 epochs by setting negative sampling to 15 and dimension of the output vector to 128.

### B. Datasets for experiments

We performed our experiments on MovieLens datasets (1M and 10M) [15] and Amazon review dataset [16], that are well-known benchmarks for recommendation tasks.

**MovieLens datasets:** These datasets provide ratings and browsing logs of users; the ratings come from real MovieLens users. The ratings range from  $[0.5, 5]$  in half-star increments. Experiments on both the MovieLens 1M and 10M datasets yield qualitatively similar insights. Hence, we report results only on the MovieLens 10M dataset which contains 10,000,054 ratings from 71,567 different users about 10,677 distinct movies.

**Amazon review dataset:** The Amazon product review dataset released by He *et al.* [16] comprises customer reviews and ratings for different Amazon products. For the purpose of this paper, we used the 5-core cellphone and accessories dataset. This dataset contains 194,439 reviews of 27,879 users for 10,429 different products, where each user and item has at least 5 reviews.

We applied the rating-SVD and item2vec algorithms on these datasets to find the top- $k$  RIRs for each item, and then created the RIN (see Figure 2) to measure the observed exposures for each algorithm.

### C. Skew in exposure of items due to RIRs

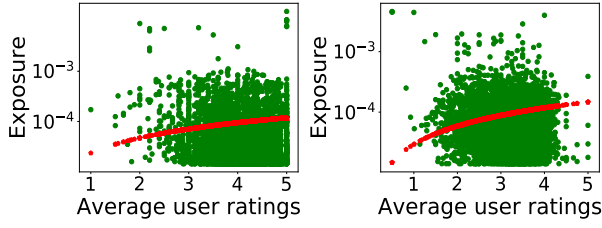
We applied rating-SVD and item2vec algorithms on MovieLens and Amazon product review datasets (described above) to find the top- $k$  recommendations for each item. We experimented with different  $k$  values, and the results were qualitatively similar in all cases. Hence, we report the results for the top-10 related item recommendations.

The existing RIR algorithms perform very well as regards to finding related items for items. However, the exposure that different items get is found to be heavily skewed. Figure 3 shows the cumulative proportion of exposure distribution on the  $y$ -axis, and  $x$ -axis shows the cumulative proportion of items in the increasing order of their exposure (from left to right). The horizontal line (in black) corresponds to 25% of the entire exposure and the corresponding vertical line denotes the percentage of items accounting for it. Figure 3(a) shows that the top 25% of the items with most exposure in the Amazon cellphone and accessories dataset account for 75% of the entire exposure. Thus, a small fraction of the item-set gets very high fraction of the entire exposure; put differently, there is very low item-space coverage.

**Are the top items deserving of the exposure?** Based on the observation of this skewed exposure distribution, a plausible question can be raised: are those 25% items of very high quality? If so, one may argue that such items, having better quality than others, deserve more exposure than others, i.e., gap in quality can explain this skew in the exposure distribution. To further dwell on this particular line of argument, we investigated both the quality distribution and exposure distribution together.

There can be many measures for quality, based on domain experts’ opinion (e.g., critical reviews of movies), public opinion (e.g., ratings given by consumers on e-commerce sites), awards won by movies, and so on. Specifically, in this work, we assume the *average user-rating* of an item as the quantification of its quality. We further normalize the quality distribution so that the quality of all items add up to 1. Note that we understand the limitations of using average user ratings as a quality measure; however, we believe that such ratings in most cases are a reflection of an item’s perceived quality for a given user.

**Observations:** A striking observation we made is the large gap between the quality and exposure of different items. Only 6% – 7% of the total number of items in the Amazon dataset have comparable quality and exposure, with a stark disparity between quality and exposure for a large majority of items. Figure 4 shows scatter plots for ranking-SVD RIRs on Amazon review (Figure 4(a)) and MovieLens (Figure 4(b)) datasets. In these figures, the  $x$ -axis is the average user rating (quality) of different items and  $y$ -axis is the log of their observed exposures. The red curve in each figure is the log plot of the quality distribution, while the green dots show the observed exposure of the corresponding items. Note that many items having low user-ratings (in the range  $[1, 2]$ ) enjoy significant observed exposure, while many high quality items are deprived of the same. Similar trends are found for item2vec on both the datasets (results omitted for brevity).



(a) Amazon product review (b) MovieLens

Fig. 4: (color online) Scatter plot of log of observed exposure and average user rating of different items in Amazon product review, and MovieLens dataset (using rating-SVD). The red curve shows the log plot of the quality distribution.

RIN	High-quality items	Low-quality recommended items
Amazon	Plantronics Windsmart Headset (4.3)	Plantronics Headset Charging Cradle for Voyager (2.2)
	Motorola H385 Bluetooth Headset (5.0)	Motorola Hands-Free Headset (2.9)
MovieLens	Rear Window (8.5)	Meatballs part II (3.7)
	12 Angry Men (8.9)	Stop! Or My Mom Will Shoot (4.2)

TABLE I: Examples of low-quality items that get over-exposed being recommended by high-quality items. Values within () are average user-ratings.

We observe that RIR links often lead to poor-quality movies from high-quality ones, resulting in higher exposure of unworthy items, potentially at the cost of other more worthy items. The reason behind such phenomenon can be formation of recommendation links (due to the underlying relatedness algorithm). Note that if a poor-quality item gets recommended on the web-page of a popular high-quality item, it may be bestowed some exposure simply due to the merit of the high quality item from which it is getting recommended. For instance, many customers will organically end up at the item page of the high quality item and in turn, they will get recommended to the said low quality items increasing their accessibility and visibility. This phenomenon may get reinforced over time as many customers would end up exploring both the items in the same session of exploration [14], [32].

Table I shows a few illustrative examples of the above phenomenon. For example, a high-quality item ‘Motorola H385 Bluetooth Headset’ (avg. user rating 4.9) recommends the relatively low-quality item ‘Motorola Hands-free Headset’ (avg. user rating 2.9) in both rating-SVD and item2vec recommendations (note that there is a quality drop of almost 40%). Similarly, for the MovieLens dataset, several high-quality movies having user-ratings higher than 8.0 recommend movies of much lower user ratings (according to ranking-SVD). Note that, the average user ratings for movies are obtained from their corresponding IMDb pages.

Table II shows a few illustrative examples of the amount of distortion in exposure of some items introduced by RIR algorithms. Many high quality items, e.g., ‘Samsung Galaxy S5 SM-G900H’, etc. have been severely deprived of exposure at the cost of significant exposure of items like ‘Google Nexus Wireless Charger’, etc.

Given the huge inventory size of online platforms, RIR systems are one of the primary tools through which users explore the universe of items. Thus, these systems play a

High ratings, deprived of exposure	Low ratings, with high exposure
<b>Amazon Cellphones and Accessories</b>	
Samsung 3.5mm Stereo Headset for Galaxy S4 (4.2, $\frac{1}{10}$ times)	Google Nexus Wireless Charger from Google (1.2, 10 times)
TechPro Galaxy S4 ShatterProof Premium Tempered Glass Screen Protector (4.83, $\frac{1}{8}$ times)	Sony Ericsson Liveview watch micro display for Android Devices (1.4, 9 times)
Samsung Galaxy S5 SM-G900H (4.8, $\frac{1}{7}$ times)	HTC One Screen Protector, Spigen Steinheil GLAS.t SLIM Premium Tempered Glass (1.35, 9 times)
<b>MovieLens</b>	
Bittersweet Motel (8.0, $\frac{1}{11}$ times)	Diebinnen (5.5, 17 times)
Men With Guns (7.6, $\frac{1}{10}$ times)	Slaughterhouse Rock (3.9, 10 times)
Braveheart (8.3, $\frac{1}{12}$ times)	Get Over it (5.7, 9 times)

TABLE II: High-quality (low-quality) items that are differently exposed. Values within () denote average user-ratings and the ratio of observed exposure to normalized quality. significant role in deciding how much *exposure* (or visibility) different items receive. Hence, an unjust exposure distribution can have detrimental repercussions for the producers of these items and their livelihood. Being keyed to relatedness, these algorithms inadvertently overlook these important aspects of different stakeholders in the business cycle. In section VI we suggest three different intervention mechanisms (**FaiRIR**) to appropriately adjust the skew in the exposure of items.

## V. DESIRED EXPOSURE AND EXPOSURE BIAS

Next, we discuss how exposure can be *fairly distributed* among a set of items, by motivating it through the lens of distributive justice [38]. We then define ‘Exposure bias’, given the desired and observed distributions of exposure.

### A. Desired exposure of items

Exposure in an online platform is a beneficial commodity, hence the producers of items would prefer having more of it (than having less). In such a scenario, an intuitive notion of fairness would be **equality of exposure**, i.e., the exposure should be uniformly distributed among all the items (by recommended them uniformly). However, the characteristics of the items should also be taken into account, since these characteristics may provide prima facie grounds for a departure from equality [38]. For instance, all items are probably *not* of similar merit or intrinsic quality. This difference in ‘merit’ or ‘quality’ can be a justified reason for departure from equality. Thereby, the ‘desired exposure’ of an individual (item) can be determined by its ‘deservingness’ (merit)<sup>2</sup>. This departure from equality is well established through the notion of **meritocratic fairness** and the related literature on meritocracy [39], [40]. For instance, a high-quality item is considered more deserving of user attention than a low-quality item.

Alternatively, the desired exposures of various items can also be driven by a broader idea of societal welfare. For instance, YouTube ‘Up next’ related video recommendation has recently been criticised for leading users to far-right echo chamber and extremist content [41], potentially influencing elections (e.g., the Brazilian presidential election [42]). In response, YouTube tweaked its ‘Up next’ algorithm, and started recommending Fox News videos from far-right conspiracy

<sup>2</sup>Desiredness should not be confused with deservingness, i.e., desiredness  $\neq$  deservingness in general. Deservingness, in contrast, is an extreme case of desiredness.

theory videos, instead of other videos from the same channels [43]. Clearly, YouTube deemed some videos *unworthy* of the exposure they were getting earlier and decided to nudge users to follow other videos. In some scenarios, it might be legally required to provide each item with some minimum amount of exposure, regardless of the item attributes [10], [13].

**Desiredness as a control knob for fairness:** Note that, we do *not* argue for any particular notion of desired exposure distribution; rather, the formulation and algorithms given in the subsequent sections are *agnostic* to any measure of desiredness. Rather than advocating for any specific desired exposure, we perceive desiredness as a necessary *controllable knob* in our framework to ensure fairness in the final outcomes. Hence, if some legislation or a particular platform has a sacrosanct quantification of the desiredness of each item, the same can be easily plugged into our proposed fairness interventions.

### B. Estimating desired exposure

We denote the desired exposure of item  $i$  as  $E_d(i)$ , and the desired exposure distribution over all items as  $E_d$ . In this work, as a proof of concept, we consider a generic formulation to accommodate multiple types of desired exposure distributions. We consider, a fraction  $\beta \in [0, 1]$  of the total exposure is equally distributed among all items. This fraction of the exposure takes care of the minimum exposure of all items (and their producers). It is meant to provide all items with some minimum exposure to satisfy the basic needs of the items and their producers (as argued in [13]). The remaining  $(1-\beta)$  fraction of the total exposure is distributed proportional to the quality or merit of individual items, thus advocating *meritocratic fairness* [39], [40]. Notice, the above formulation of  $E_d$  reduces to purely meritocratic distribution of exposure for  $\beta = 0$ , and to uniform distribution of exposure for  $\beta = 1$ . The exposure distributions are normalized so that the total exposure of all items in the item-set sum up to 1, i.e.,  $\sum_{i \in \mathbf{I}} E_d(i) = 1$ .

As mentioned earlier, in this work, we assume the *average user-rating* of an item as the quantification of its merit / quality. The importance that we attach to an item's merit to obtain its desired exposure is controlled by the parameter  $\beta$ .

**A potential limitation of user-ratings:** One potential concern about using average user-ratings as a quality metric, might be that the number of ratings an item gets is partly driven by the existing recommendation algorithms. However, we believe that, although a user may have been led to an item via some recommendation, her rating would reflect the inherent quality of the item as perceived by her. Further, we also considered a slightly different quality measure – average user-rating of an item, weighted by its number of ratings. The qualitative results of the analyses remained similar in this setting too. Hence, for simplicity and completeness we consider the average user-rating score to be the indicator of quality throughout this paper.

### C. Defining exposure bias

According to our formulation, a RIR system would be fair (unbiased), if it gives every item an observed exposure that is proportional to its desired exposure. Since  $E_o(i)$  and  $E_d(i)$

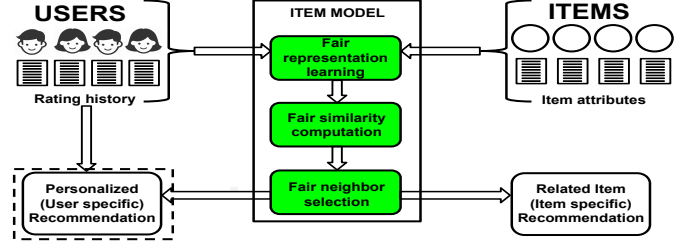


Fig. 5: Recommendation pipeline shown earlier in Figure 1, now shown with FaiRIR suit of fairness interventions.

denotes the *observed and desired exposures* of item  $i$ , mathematically, a RIR system is fair if  $\frac{E_o(i)}{E_d(i)} = \frac{E_o(j)}{E_d(j)} \forall i, j \in \mathbf{I}$ . As discussed in the previous section, a RIR system  $R$  may lead to items getting different observed exposures than what is desired. *Exposure Bias (ExpBias)* is the deviation caused due to  $R$  between the desired and observed exposure of items. Following the set up in our prior work [14], we measure *ExpBias* by KL divergence [44] between the observed exposure distribution  $E_o = \{E_o(i) \forall i \in \mathbf{I}\}$  and the desired exposure distribution  $E_d = \{E_d(i) \forall i \in \mathbf{I}\}$ :

$$ExpBias(R) = D_{KL}(E_o || E_d) = \sum_i E_o(i) \log \left( \frac{E_o(i)}{E_d(i)} \right) \quad (1)$$

**Categorization of items:** Based on the observed and desired exposure of items, we categorize items in three different classes based on how closely the observed exposure replicates their desired exposure.

- (a) **Under-exposed:** item  $i$  is *under-exposed* if  $1 - \epsilon \leq \frac{E_o(i)}{E_d(i)}$ ,
- (b) **Over-exposed:** item  $i$  is *over-exposed* if  $\frac{E_o(i)}{E_d(i)} \geq 1 + \epsilon$ ,
- (c) **Adequately-exposed:** item  $i$  is *Adequately-exposed* if  $1 - \epsilon \leq \frac{E_o(i)}{E_d(i)} \leq 1 + \epsilon$ ,

While this threshold ( $\epsilon$ ) can be chosen based on prior context and established regulations, in this paper, we use  $\epsilon = 0.2$ . Note that similar thresholds have been used in multiple prior works too [14], [45], [46].

## VI. MITIGATING EXPOSURE BIAS

In this section, we propose multiple interventions (**FaiRIR**) in the recommendation pipeline (shown in Figure 5), that can reduce exposure bias, by making exposure of item  $i$  proportional to its desired exposure  $E_d(i)$ , while maintaining the relatedness of recommendations.

(I) **FaiRIR<sub>rl</sub> (Fair representation learning):** Change the latent space representation of items such that items with similar desired exposure come closer in the latent space.

(II) **FaiRIR<sub>sim</sub> (Fair similarity computation):** Introduce desired exposure based similarity into the metric to compute similarity of items.

(III) **FaiRIR<sub>nbr</sub> (Fair neighbor selection):** For a particular source item, instead of selecting the most similar items, select items considering both similarity and desired exposure.

Next, we will elaborate these three procedures.

### A. FaiRIR<sub>rl</sub>: Fair representation learning

Our first approach tries to learn fair representations of the items from the latent space representations obtained from vanilla rating-SVD or item2vec, to mitigate the induced exposure bias. This approach is inspired by recent works attempting to incorporate individual fairness in various algorithms [25],

[26], [47]. Similar to [26], we learn representation that preserves the similarity of desired exposure between items while minimizing the loss in relatedness, and, thereby, reconciling both desired exposure and relatedness.

**Input:** Let us assume there are  $M$  items, and for each item, we have  $N$  latent attributes from the representations learned by the existing RIR algorithms (e.g., rating-SVD, item2vec), resulting in a  $M \times N$  matrix  $X$ . We denote the record for the  $i$ -th item as vector  $x_i$  and the value  $x_{ir}$  denotes the  $r$ -th attribute of  $x_i$ . Note that, this representation space only captures the relatedness between items.

**Considering desired exposure:** We design a *desiredness graph*  $D_G(V, E)$ , where every node represents an item, i.e.,  $V = \mathbf{I}$ , and every item is connected to  $k$  other items who have similar desired exposure, i.e.,  $E = \{(i, j) : E_d(i) \simeq E_d(j)\}$ . We obtain the node representation  $x_i^*$  by using a network representation learning algorithm node2vec [48] for each item  $i$ . Note that, representation  $x_i^*$  of an item captures similarity of the items based on their desired exposure only.

**Goal:** Our goal is to learn fair representation  $\tilde{x}_i$  for item  $i$ , which preserves both relatedness and similarity in desired exposure. Analogous to the input, the output is again a matrix  $\tilde{X}$  with dimension  $M \times N$  where  $i$ -th row represents the fair representation of item  $i$ .

**A simple approach to achieve fair representations:** Given the two sets of representations  $x_i$  and  $x_i^*$ , the most intuitive way to generate a fair representation is to concatenate both the sets of representations. We denote such an approach as  $\text{FaiRIR}_{concat}$ . Such a simple approach may be effective because while  $x_i$  encodes the information regarding relatedness,  $x_i^*$  encodes the information regarding desiredness. However, as we shall show later in Section VII-A, the performance of such a simple approach is not stable across different datasets and vanilla RIR algorithms. Hence, we proceed to learn the fair representations by optimizing a loss function which reconciles between relatedness loss and desiredness loss in the final learnt representations (described next).

**Probabilistic clustering:** Following prior works [25], [26], our framework treats the goal of computing fair representation as the formal problem of probabilistic clustering. The aim is to learn  $\mathbb{K}$  prototype vectors  $v_h$  ( $\forall h \in \{1, 2, \dots, \mathbb{K}\}$ ), such that item  $i$  ( $\forall i \in \{1, 2, \dots, M\}$ ) is assigned to clusters in a probabilistic manner, such that the probabilities encode the distance between item  $i$  and the prototype  $v_h$ . Given the distance function  $d$  in an  $N$ -dimensional latent space, the probability that item  $i$  belongs to cluster with prototype  $v_h$  is given as:  $u_{ih} = \frac{\exp(-d(x_i, v_h))}{\sum_{t=1}^{\mathbb{K}} \exp(-d(x_i, v_t))}$ . Notice such probabilistic clustering based set up can be viewed as a low-rank representation of the input matrix  $X$  with  $\mathbb{K} < M$ , so that we are able to reduce the attribute values into a more compact form.

**Output representation:** The fair representation  $\tilde{X}$ , a matrix of dimension  $M \times N$  of fair output vectors  $\tilde{x}_i$  ordered row-wise, includes (a)  $\mathbb{K} < M$  prototype vectors  $v_h$  of  $N$  dimensions. (b) A probability distribution  $u_i$ , of  $\mathbb{K}$  dimensions, for each item  $i$ .  $u_{ih}$  represents the probability that item  $i$  belongs to the cluster with prototype  $v_h$ . Mathematically,

$\tilde{x}_i = \sum_{h \in \{1, \dots, \mathbb{K}\}} u_{ih} * x_j$ , where  $u_{ih}$  is as defined earlier.

**Loss function:** Next, we present the loss function which optimizes for the reconstruction loss between the input representation  $X$  and the fair output representation  $\tilde{X}$  while preserving the desiredness of the products.

$$L = \lambda * \underbrace{\sum_{i=1}^M \sum_{r=1}^N (x_{ir} - \tilde{x}_{ir})^2}_{\text{Relatedness loss}} + \mu * \overbrace{\sum_{i, j \in \{1, \dots, M\}} [d(\tilde{x}_i, \tilde{x}_j) - d(x_i^*, x_j^*)]^2}_{\text{Desired exposure based similarity loss}} \quad (2)$$

This loss  $L$  has two separate parts: (i) **Relatedness loss:** the sum of the squared errors between the input representation matrix  $X$  and the (low dimensional) output representation matrix  $\tilde{X}$ , and (ii) **Desired exposure based similarity loss:**  $d()$  captures the distance between desired exposure of two items, computed as **Euclidean distance** between their vector representations. Hyper-parameters  $\lambda$  and  $\mu$  decide the importance we want to associate with these two losses.

### B. $\text{FaiRIR}_{sim}$ : Fair similarity computation

As discussed in section IV, the existing cosine similarity between two item representations,  $sim(x_i, x_j) = \frac{x_i \cdot x_j}{\|x_i\| \|x_j\|}$ , accounts for relatedness; however, it does *not* account for the relative gap between their desired exposure. In this intervention, we propose to incorporate similarity of desired exposure between items along with relatedness. If  $E_d(i)$  and  $E_d(j)$  denote the desired exposure of items  $i$  and  $j$  respectively, then the new similarity measure is defined as:

$$sim(x_i, x_j) = \underbrace{\exp(-|E_d(i) - E_d(j)|)}_{\text{Desired exposure based similarity}} * \overbrace{\frac{x_i \cdot x_j}{\|x_i\| \|x_j\|}}^{\text{Relatedness similarity}} \quad (3)$$

Using the similarity metric mentioned in equation 3, we promote items having higher desired exposure based similarity and relatedness by giving them higher similarity score.

### C. $\text{FaiRIR}_{nbr}$ : Fair neighbour selection

Next, instead of changing the representation or the similarity metric, we change the way the items are selected for recommendation. In practice, against every item, an equal number (say,  $k$ ) of items are recommended; these  $k$  items are usually most similar to the source item, based on some similarity metric. However, the number of recommendations each item will end up receiving is not controlled. We propose a fair way of selecting the  $k$  neighbours such that the likelihood of an item being selected is proportional to its desired exposure. That is, the recommendation would be fair if the likelihood of a highly desired item being recommended is greater than that of a less desirable item. If  $R_t$  denotes the list of items recommended for item  $t$ , then for all item-pairs  $(i, j)$ ,  $\forall t \in \{1, 2, \dots, M\}$ ,  $P(i \in R_t) \geq P(j \in R_t) | E_d(i) \geq E_d(j)$ . If an RIR algorithm follows the above equation, while preserving the notion of relatedness, it is likely to mitigate exposure bias.

Algorithm 1 details our proposed algorithm. Through the *Desired* dictionary, it ensures that the likelihood of recommendation among different items follow the similar distribution as defined by their desired exposure. Effectively, for any source item, we have two ranked list of items according to their relatedness and desired exposure. In order to reconcile between

---

**Algorithm 1** Fair neighbor selection
 

---

**Input:** *Desired*: number of recommendations desired by each item, *k*: number of recommendations per item and similarities among all items *Sim*

**Output:** *Recommendation*

```

function FIND_NEIGHBOR(u, k, Desired, Sim)
  Similarity = Ranked list of items based on their similarity to item u
  Desiredness = Ranked list of items based on their Desired number of recom-
  mendations
  Final = Aggregated ranked list using Similarity and Desiredness
  Return top-k items based on Final ranked list
end function

```

```

procedure RECOMMENDATION(k, Desired, Sim)
  Recommendation =  $\phi$ 
  for all item u in the item space do
    RelatedItems = FIND_NEIGHBOR(u, k, Desired, Sim)
    for all item v in RelatedItems do
      Recommendation = Recommendation  $\cup$  (u, v)
      Desired[v] = Desired[v] - 1
      if Desired[v] = 0 then
        remove v from Desired
      end if
    end for
  end for
  Return Recommendation
end procedure

```

---

these two rankings, we use a well known rank aggregation method, based on borda count [49]. Intuitively, any item having higher rank in both the ranked lists will be considered the most suitable related item to be recommended.

**Practical applicability of FaiRIR:** There are two key questions regarding the practical applicability of FaiRIR algorithms in real-world setting – (i) does the deployment need a knowledge of the internal details of the RIR algorithm being used?, and (ii) can the FaiRIR algorithms adapt to a dynamic setting, e.g., when new items emerge intermittently.

FaiRIR<sub>rl</sub> and FaiRIR<sub>sim</sub> can be used only when internal details of the RIR algorithm are known (e.g., user-item logs, similarity metric etc.). Also, given that these interventions are applied in the earlier stage of the recommendation pipeline, they are more suitable to adapt to a dynamic setting. FaiRIR<sub>nbr</sub>, on the other hand, can be used even by considering the related item recommendation algorithm as a black box, provided we have the recommendation outputs and the desired exposure of all items. For example, in case of an existing on-line platform, we do not exactly know the RIR algorithm that is used. However, since we can capture the recommendation outputs in the form of an RIN, we can use any network level similarity as a proxy for relatedness and rewire the RIN to reduce the exposure bias.

To adapt to the dynamic setting, in case of FaiRIR<sub>nbr</sub>, one can start recommending the newly emerging items from items which already exist in the network and are similar to them (where the similarity can be based on some metadata or content-based measures, such as the genre or actors of movies); thereby mitigating the requirement of re-wiring the entire RIN.

## VII. EXPERIMENTAL EVALUATION

We evaluate our interventions (FaiRIR algorithms) on MovieLens and Amazon Cell Phone and Accesories datasets. The algorithms are evaluated based on: (1) their effectiveness in mitigating exposure bias (Sec VII-A), (2) the relatedness of their recommendations (Sec VII-B), and (3) the overall utility of the recommendations to the end-users (Sec VII-C).

Algorithm		Over	Adequate	Under	<i>ExpBias</i>
rating-SVD	Vanilla	25.89 %	10.95 %	63.16 %	0.71
	FaiRIR <sub>concat</sub>	19.77 %	9.71 %	70.52 %	0.67
	FaiRIR <sub>rl</sub>	34.69 %	27.4 %	37.91 %	0.15
	FaiRIR <sub>sim</sub>	24.3 %	15.00 %	60.7 %	0.39
	FaiRIR <sub>nbr</sub>	0.07 %	98.9 %	0.03 %	0.003
item2vec	Vanilla	26.75 %	12.38 %	60.87 %	0.56
	FaiRIR <sub>concat</sub>	22.95 %	19.65 %	57.4 %	0.35
	FaiRIR <sub>rl</sub>	31.06 %	34.19 %	34.75 %	0.10
	FaiRIR <sub>sim</sub>	31.74 %	23.43 %	44.83 %	0.21
	FaiRIR <sub>nbr</sub>	0.04 %	99.94 %	0.02 %	0.002

TABLE III: % of movies that are over, adequately and under-exposed in MovieLens dataset (vanilla and intervened rating-SVD & item2vec) with desired distribution proportional to quality of the movies (i.e.,  $\beta = 0.0$ ).

### A. Mitigation of exposure bias

**FaiRIR<sub>rl</sub>:** We applied FaiRIR<sub>rl</sub> to the learnt representations of items (from the vanilla rating-SVD and item2vec algorithms) over the MovieLens and Amazon datasets, with the following parameter settings.

**Parameter setting:** We initialize the prototype vectors  $v_j$  to random values from uniform distribution in  $(0, 1)$ . To account for the variations due to the initialization, we report the results obtained from the best of three runs. For the hyperparameters  $\lambda, \mu$  in Eqn. 2, we performed a grid search over the set  $\{0.01, 0.1, 1.0, 10, 100\}$ . For  $K$ , we performed a grid search over the set  $\{10, 20, 30\}$ . We found the best performance (argmin for the loss function in eqn 2) for  $\lambda = 1, \mu = 0.01$  and  $K = 20$ .

**Results:** The effectiveness of FaiRIR<sub>rl</sub> is shown in Table III (for MovieLens) and Table IV (for Amazon) over both rating-SVD and item2vec algorithms. Compared to the original algorithms, the exposure bias has decreased significantly, with an increase in fraction of items being adequately exposed. E.g., for rating-SVD, percentage of items adequately exposed has increased from 06.62% to 23.71%, and the exposure bias has reduced from 1.28 to 0.18 for the Amazon dataset.

**FaiRIR<sub>rl</sub> outperforms FaiRIR<sub>concat</sub>:** Recall in Section VI, we discussed two potential ways for fair representation learning. While one was the aforementioned optimization framework, the other was simple concatenation of representations learnt from vanilla RIR ( $x_i$ ) algorithm and desiredness graph ( $x_i^*$ ). In Tables III and IV, we show the efficacy of the approach in mitigating exposure bias for  $\beta = 0.0$ . While using FaiRIR<sub>concat</sub> we see reasonable improvement on both the datasets for representations learnt from Item2Vec approach; the performance was not so great for representations learnt from SVD approach. In either case, the proposed optimization based representation learning (FaiRIR<sub>rl</sub>) outperforms the concatenation approach and its performance is more robust across datasets and across RIR algorithms. Hence, in the remainder of the paper, we shall consider FaiRIR<sub>rl</sub> to be the fair representation learning based mitigation approach and compare it with interventions at other stages of the pipeline.

**FaiRIR<sub>sim</sub>:** The effect of FaiRIR<sub>sim</sub> is also shown in Tables III and IV. In all cases, the *ExpBias* has decreased, with an increase in percentage of items being adequately exposed.

**FaiRIR<sub>nbr</sub>:** The effect of FaiRIR<sub>nbr</sub> is also shown in Tables III and IV. In all cases, the exposure bias has decreased substantially (almost reduced to zero), with a significant increase in



Algorithm		Over	Adequate	Under	$ExpBias$
$\beta = 0.0$					
rating-SVD	Vanilla	18.04 %	06.62 %	75.34 %	1.28
	FaiRIR <sub>conca</sub>	20.14 %	12.78 %	67.1 %	0.75
	FaiRIR <sub>rl</sub>	33.01 %	23.71 %	43.28 %	0.18
	FaiRIR <sub>sim</sub>	14.60 %	8.70 %	76.70 %	1.2
	FaiRIR <sub>nbr</sub>	0.0 %	100.0 %	0.0 %	0.002
item2vec	Vanilla	15.73 %	05.67 %	78.60 %	1.22
	FaiRIR <sub>conca</sub>	31.26 %	19.41 %	49.33 %	0.32
	FaiRIR <sub>rl</sub>	35.17 %	26.71 %	38.12 %	0.14
	FaiRIR <sub>sim</sub>	20.65 %	16.77 %	62.57 %	0.81
	FaiRIR <sub>nbr</sub>	0.0 %	100.0 %	0.0 %	0.002
$\beta = 0.25$					
item2vec	Vanilla	15.66 %	05.66 %	78.68 %	1.21
	FaiRIR <sub>rl</sub>	32.43 %	22.66 %	44.91 %	0.18
	FaiRIR <sub>sim</sub>	18.90 %	7.67 %	73.43 %	0.91
	FaiRIR <sub>nbr</sub>	0.0 %	100.0 %	0.0 %	0.001
$\beta = 0.75$					
item2vec	Vanilla	15.46 %	05.68 %	78.86 %	1.21
	FaiRIR <sub>rl</sub>	32.89 %	24.28 %	42.83 %	0.16
	FaiRIR <sub>sim</sub>	15.96 %	6.67 %	77.57 %	1.14
	FaiRIR <sub>nbr</sub>	0.0 %	100.0 %	0.0 %	0.0002
$\beta = 1.00$					
item2vec	Vanilla	15.41 %	05.78 %	78.81 %	1.21
	FaiRIR <sub>rl</sub>	32.12 %	30.37 %	37.51 %	0.12
	FaiRIR <sub>sim</sub>	15.41 %	05.78 %	78.81 %	1.21
	FaiRIR <sub>nbr</sub>	0.0 %	100.0 %	0.0 %	0.0

TABLE IV: % of items that are over, adequately and under-exposed in Amazon review dataset (vanilla and intervened item2vec. For rating-SVD representative results shown for  $\beta = 0.0$ ; other results are similar to item2vec and omitted).

percentage of items being adequately exposed.

Figure 6 shows scatter plots for the three interventions on the MovieLens dataset; each plot shows log of observed exposure on  $y$ -axis and desired exposure ( $\beta = 0.0$ ) on  $x$ -axis. From these figures as well, it is evident that the distribution of observed exposure is closest to that of desired exposure for FaiRIR<sub>nbr</sub> (Figure 6(c)).

**Analysis on multiple desired exposure distribution:** We also analyze our proposed interventions with different desired exposure distributions, by varying  $\beta$  in the range  $[0, 1]$ . The results for Amazon dataset are shown in Table IV (similar results are obtained for the MovieLens dataset, not shown for brevity). We see that irrespective of the  $\beta$  values, FaiRIR<sub>rl</sub> and FaiRIR<sub>nbr</sub> are very effective in mitigating the exposure bias. However, with increase in  $\beta$ , the effectiveness of FaiRIR<sub>sim</sub> reduces. The reason being, as  $\beta$  approaches 1.0 (i.e., uniform desired exposure distribution), the ‘desired exposure based similarity’ part in equation 3 reduces to 1.0 and  $sim(x_i, x_j)$  becomes only the relatedness similarity (cosine similarity).

Overall, FaiRIR<sub>nbr</sub> is seen to be most effective in reducing exposure bias, probably due to the following reason. While FaiRIR<sub>rl</sub> and FaiRIR<sub>sim</sub> attempt to reduce exposure bias indirectly by altering the representation learning / similarity computation in the latent space, FaiRIR<sub>nbr</sub> directly controls the number of other items from which a particular item  $i$  is recommended (the desired number of recommendations for  $i$ ), which specifically ensures that  $i$  gets an exposure close to its desired exposure.

**Summary:** We have shown that the proposed methods are successful in mitigating the induced exposure bias significantly. While the performance of FaiRIR<sub>rl</sub> and FaiRIR<sub>nbr</sub> show significant improvement across both the datasets and RIR algorithms, the improvement is less stable for FaiRIR<sub>sim</sub>.

The next natural question to investigate is whether this

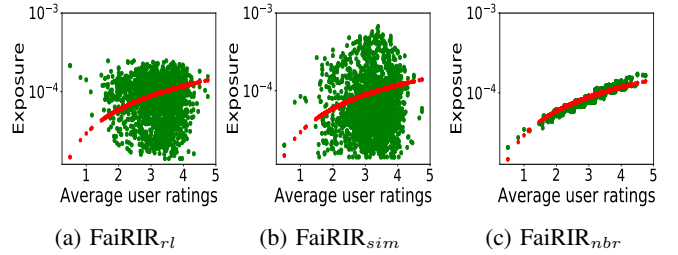


Fig. 6: (color online) Scatter plot of log of observed exposure and average user rating of movies in MovieLens dataset after different interventions on the rating-SVD. The red scatter shows their desired exposure with  $\beta = 0.0$ .

Algorithm		Genre overlap	Like overlap	Relevance score
rating-SVD	Vanilla	0.53	0.64	3.73
	FaiRIR <sub>rl</sub>	0.34	0.56	2.69
	FaiRIR <sub>sim</sub>	0.44	0.64	3.63
	FaiRIR <sub>nbr</sub>	0.44	0.63	3.62
item2vec	Vanilla	0.37	0.56	2.79
	FaiRIR <sub>rl</sub>	0.27	0.52	2.54
	FaiRIR <sub>sim</sub>	0.35	0.59	2.77
	FaiRIR <sub>nbr</sub>	0.34	0.59	2.76

TABLE V: Relatedness of recommendations generated over the MovieLens dataset: (i) mean genre overlap of recommended & source item, (ii) Like overlap between users who liked the source & recommended items, (iii) relevance score according to AMT survey.

control of exposure bias comes at a cost of loss in relatedness of the recommendations.

### B. Preserving recommendation relatedness

Mitigating bias due to algorithms usually incur an associated cost in terms of drop in performance (e.g., drop in accuracy in fair classifiers [30]). In the context of RIRs, the relatedness of the recommendation is the prime objective of the algorithm. Hence, in this experiment we check the incurred loss in relatedness due to our proposed FaiRIR interventions.

**Genre / Category based similarity:** Intuitively, relatedness of recommendations will be high if the recommended items are ‘similar’/‘related’ to the source item. The measure of relatedness is often domain-dependent. For instance, in the domain of movies, every movie has a set of one or more genres. We measure the similarity of a source movie and a recommended movie by their *genre overlap* with respect to the source movie, i.e., by the fraction of genres preserved by the recommended movie as compared to that of the source movie. For instance, let the movie *Gladiator* be recommended from the movie *Avatar* by an algorithm. As per the IMDb website, *Avatar* has genres {Action, Adventure, Fantasy} and *Gladiator* has genres {Action, Adventure, Drama}. Thus, the similarity between the movies is  $\frac{2}{3}$ . Similarly, every item in the Amazon dataset has an associated set of *categories*, which we used in an identical fashion to compute relatedness of two Amazon items. Now, we compute the relatedness of RIRs generated by an algorithm as follows. For each pair of items  $(i, j)$  where  $j$  has been recommended for  $i$  by the algorithm, we compute the genre (category) overlap between  $i$  and  $j$ , and then take the mean across all such pairs.

**Results:** The ‘Genre overlap’ column of Table V and Table VI show the genre/category overlap for the various RIR algo-

Algorithm		$\beta = 0.0$	$\beta = 0.25$	$\beta = 0.75$	$\beta = 1.0$
rating-SVD	Vanilla	0.46	0.46	0.46	0.46
	FaiRIR <sub>rl</sub>	0.41	0.40	0.41	0.40
	FaiRIR <sub>sim</sub>	0.45	0.45	0.46	0.46
	FaiRIR <sub>nbr</sub>	0.44	0.44	0.45	0.45
item2vec	Vanilla	0.50	0.50	0.50	0.50
	FaiRIR <sub>rl</sub>	0.41	0.41	0.41	0.40
	FaiRIR <sub>sim</sub>	0.48	0.48	0.49	0.50
	FaiRIR <sub>nbr</sub>	0.46	0.46	0.47	0.47

TABLE VI: **Relatedness of recommendations (mean category overlap between source and recommended items) over Amazon dataset, shown for different desired distributions.**

rithms, for the MovieLens and Amazon datasets respectively. We observe decrements in the average genre overlap for all the interventions (as compared to the original algorithms), which is the expected cost of minimizing exposure bias. The reduction is severe in case of FaiRIR<sub>rl</sub> only, whereas it is not so severe in case of FaiRIR<sub>sim</sub> and FaiRIR<sub>nbr</sub>. Note that, FaiRIR<sub>rl</sub> is an *application-agnostic* methodology, where the main objective is to learn fair representations, while the other two interventions are *application-specific*. Hence, the foregoing observation is in line with the findings in fairness literature, wherein application-agnostic approaches tend to incur higher losses in performance [26].

**Analysis on multiple desired exposure distribution:** Table VI shows the category overlap for various desired exposure distributions (various values of  $\beta$ ) for the Amazon product review dataset. The interpretation of the results is the same as discussed above. The performance of all the FaiRIR algorithms in preserving relatedness is pretty much stable. However, since FaiRIR<sub>sim</sub> reduces to vanilla RIR algorithms as  $\beta$  approaches 1.0, it does not incur any additional loss in relatedness.

### C. Judging overall utility of recommendations

Considering that the ultimate objective of RIR algorithms is to satisfy human users, we conduct the following two evaluations for the proposed algorithms – (1) We measure the mean overlap of common users who liked both a source item and a recommended item, and (2) We conduct a user survey for judging the utility (relevance) of the recommendations generated by various algorithms on the MovieLens dataset.

**(1) Users’ propensity to like the recommended items:** When a source item  $i$  recommends an item  $j$ , intuitively, the utility of the recommendation is high if the recommended item  $j$  is ‘liked’ by most users who had also liked the source item  $i$ . To this end, we consider a user to have liked a given movie / item if (s)he gives a rating/score of more than 3.5 (out of 5) i.e., a score of more than 70%. Our choice is influenced by studies claiming 3.5–4.5 (out of 5) being the sweet spot [50], [51] in five star rating systems. Note that, since all users who had liked  $i$  need not consume or rate item  $j$ , we need to consider only those users who liked  $i$  and rated the recommended item  $j$ . For the item  $i$ , let  $R_i$  and  $L_i$  be the set of users who rated and liked the item respectively ( $L_i$  is a subset of  $R_i$ ). For a source item  $i$  and a recommended item  $j$  (from  $i$ ), we evaluate the **like overlap** as  $\frac{|L_i \cap L_j|}{|L_i \cap R_j|}$ , i.e., out of all users who liked item  $i$  and rated item  $j$ , what fraction of users actually liked (rated highly)  $j$ . Specifically, for each pair of items  $(i, j)$  where  $j$  has been recommended for  $i$ , we compute the like overlap between  $i$  and  $j$ , and then take the mean across all such pairs.

**Results:** Table V (‘Like overlap’ column) and Table VII (all columns, for different  $\beta$  values) show the mean *like overlap* for the various RIR algorithms, for the MovieLens and Amazon datasets respectively. For the MovieLens data (Table V last column), while using rating-SVD, the average *like overlap* reduces for FaiRIR<sub>rl</sub> (as compared to the original algorithm); however, there is practically no reduction in like overlap for FaiRIR<sub>sim</sub> and FaiRIR<sub>nbr</sub>. Interestingly, while using item2vec, *FaiRIR<sub>sim</sub>* and *FaiRIR<sub>nbr</sub>* (*like overlap* = 0.59) *have outperformed even the original algorithm (like overlap = 0.56)*. For Amazon dataset (Table VII), for both rating-SVD and item2vec algorithms, all the FaiRIR algorithms have performed comparably or even better than the original algorithms. We repeated this experiment with 4 and 4.5 thresholds for *like overlap* and found qualitatively similar observations (omitted for brevity). These results show the efficacy of the proposed approaches to preserve utility of the recommendation to end users. However, one can argue that such high scores might be an artifact of the underlying desiredness assumption while designing the algorithm esp. when  $\beta = 0$  i.e., a completely meritocratic distribution. Hence, analysis on different  $\beta$  values are of utmost importance.

**Analysis on multiple desired exposure distribution:** Table VII shows the *like overlap* for various desired exposure distributions (i.e., for various values of  $\beta$ ) for the Amazon dataset. We see that the performance of all the FaiRIR interventions in preserving utility toward users is very stable and is agnostic of the underlying desired distribution (similar results for MovieLens dataset are omitted for brevity).

**(2) A user survey to judge recommendations:** We recruited human workers to assess the relevance of recommendations, via the Amazon Mechanical Turk (AMT) platform. We used ‘AMT master workers’ who are known to perform such tasks meticulously. To ensure reliable judgments, it is important that the annotators are likely to be familiar with the items whose recommendations they are being tasked to judge. Hence, we chose to perform this survey with the MovieLens dataset, since workers are much more likely to be familiar with popular movies, than with cellphones and accessories available on Amazon. We considered movies in the top 5% most popular movies (based on the number of ratings) in MovieLens dataset as our source movies. To further guarantee the reliability of the judgments, we also provided the annotators with the link to the IMDb information page of each movie, which contains all metadata about the movie along with a snippet and its trailer. The annotators were asked to browse through the IMDb page for a movie if they were not familiar with it.

For a particular source movie  $x$ , we generated top-5 recommendations using various algorithms on the MovieLens dataset. For each movie  $y$  recommended for  $x$  (in top-5) by some algorithm, we asked a worker – “If your friend likes the movie  $x$ , how likely are you to recommend movie  $y$ ?”. A worker could answer this question on a Likert scale of [1, 5] with response 1 representing “very unlikely” and response 5 representing “very likely”. The recommendations were anonymized, i.e., the workers were *not* mentioned about the source algorithm of each recommendations.

Algorithm		$\beta = 0.0$	$\beta = 0.25$	$\beta = 0.75$	$\beta = 1.0$
rating-SVD	Vanilla	0.91	0.91	0.91	0.91
	FaiRIR <sub>rl</sub>	0.90	0.89	0.90	0.89
	FaiRIR <sub>sim</sub>	0.91	0.91	0.91	0.91
	FaiRIR <sub>nbr</sub>	0.92	0.91	0.91	0.91
item2vec	Vanilla	0.81	0.81	0.81	0.81
	FaiRIR <sub>rl</sub>	0.87	0.84	0.83	0.86
	FaiRIR <sub>sim</sub>	0.85	0.84	0.82	0.81
	FaiRIR <sub>nbr</sub>	0.82	0.81	0.81	0.81

TABLE VII: Mean like overlap (overlap of users who liked the recommended & source item) of the recommendations generated by different algorithms over Amazon dataset, shown for different desired exposure distributions.

We collected responses for 100 different source movies and approximately 1,550 distinct pairs of source and recommended pairs of movies. Each pair was evaluated by at least 10 AMT workers, and we considered the average score over all these AMT workers as the utility/relevance score for this pair. The mean relevance for a RIR algorithm is computed as the average relevance over all the recommended pairs generated by it that were evaluated.

**Results:** Each of the RIR algorithms scored based on the mean of the relevance scores of different items it recommended for a given item. Table V (Relevance Score column) shows the mean relevance scores for different RIR algorithms.

For both rating-SVD and item2vec, FaiRIR<sub>sim</sub> and FaiRIR<sub>nbr</sub> preserve a high degree of relevance in their recommendations. For instance, the mean relevance score of the original rating-SVD algorithm is 3.73 (out of 5.0), while that of FaiRIR<sub>sim</sub> and FaiRIR<sub>nbr</sub> are 3.63 and 3.62 respectively. The drop in mean relevance score is even lesser in case of item2vec. To further substantiate the results, we performed *Student's T-test* on the samples of mean relevance scores, as obtained from the user survey. We found the drop in mean relevance score to be statistically significant only for FaiRIR<sub>rl</sub> (as compared to the original algorithm); for FaiRIR<sub>sim</sub> and FaiRIR<sub>nbr</sub>, the drop was *not* statistically significant. In fact, for almost 40% of the recommendations, the FaiRIR variants have *higher* relevance scores than the vanilla algorithms.

Note that, we also attempted to combine all the interventions, i.e., first learn fair representations, then fairly compute similarity, and finally select the neighbors as shown in Algorithm 1. The results for the combined approach were dominated by the effects of FaiRIR<sub>nbr</sub>. Hence, we have not reported results of the combined approach for brevity.

**Summary:** We investigated whether the mitigation of exposure bias by FairIR comes at the cost of degradation in the relatedness of the recommendations. Specifically we evaluated – (i) the preservation of relatedness of the recommendation through *genre / category overlap*, and (ii) the utility of the recommendation through *liking overlap* metric and a user-survey on AMT. Across all evaluations, we consistently observe that our proposed mechanisms (especially FaiRIR<sub>sim</sub> and FaiRIR<sub>nbr</sub>) successfully mitigate exposure bias without sacrificing much on the relatedness of recommendations.

## VIII. CONCLUSION

In this paper, we considered the impact of related item recommendations (RIRs) on the exposure of items. We show

that existing RIRs induce exposure bias by not considering any notion of the desired exposure of items. Although there can be alternate ways to estimate the exposure of different items, we believe that the qualitative results would remain unchanged. We further proposed a novel suit of algorithms (**FaiRIR**), which can reduce the exposure bias, while maintaining the effectiveness of recommendations. Note that, though the experiments in this paper are conducted on two datasets (from movie and e-commerce domains) for proof of concept, the proposed algorithms are applicable to any other domain, including job recommendation sites, and others.

In this work, we considered RIRs that only utilize the relatedness with respect to one source item. Many platforms provide personalized recommendations to users, where multiple items viewed by a target user are considered. We plan to study the effects of personalization on exposure bias in future.

## REFERENCES

- [1] C. A. Gomez-Urbe and N. Hunt, “The netflix recommender system: Algorithms, business value, and innovation,” *ACM TMIS*, 2016.
- [2] B. Smith and G. Linden, “Two decades of recommender systems at amazon.com,” *IEEE internet computing*, 2017.
- [3] Rejoinder, “The amazon recommendations secret to selling more online,” <http://rejoinder.com/resources/amazon-recommendations-secret-selling-online>, 2018.
- [4] S. Yao and B. Huang, “Beyond parity: Fairness objectives for collaborative filtering,” in *NeurIPS*, 2017, pp. 2921–2930.
- [5] Z. Zhu, X. Hu, and J. Caverlee, “Fairness-aware tensor-based recommendation,” in *ACM CIKM*, 2018.
- [6] B. Edizel, F. Bonchi, S. Hajian, A. Panisson, and T. Tassa, “Fairecsys: mitigating algorithmic bias in recommender systems,” *International Journal of Data Science and Analytics*, pp. 1–17, 2019.
- [7] S. C. Geyik, S. Ambler, and K. Kenthapadi, “Fairness-aware ranking in search & recommendation systems with application to linkedin talent search,” 2019.
- [8] A. Sharma, J. M. Hofman, and D. J. Watts, “Estimating the causal impact of recommendation systems from observational data,” in *ACM EC*, 2015.
- [9] A. Chakraborty, A. Hannak, A. J. Biega, and K. P. Gummadi, “Fair sharing for sharing economy platforms,” in *FATRec*, 2017.
- [10] G. K. Patro, A. Biswas, N. Ganguly, K. P. Gummadi, and A. Chakraborty, “Fairrec: Two-sided fairness for personalized recommendations in two-sided platforms,” in *WWW*, 2020.
- [11] P. I. B. . G. of India, “Review of policy on foreign direct investment (fdi) in e-commerce,” <http://pib.gov.in/newsite/PrintRelease.aspx?relid=186804>, 2018.
- [12] Y. Yao and F. M. Harper, “Judging similarity: a user-centric study of related item recommendations,” in *ACM RecSys*, 2018.
- [13] G. K. Patro, A. Chakraborty, N. Ganguly, and K. P. Gummadi, “Incremental fairness in two-sided market platforms: On smoothly updating recommendations,” in *AAAI*, 2020.
- [14] A. Dash, A. Chakraborty, S. Ghosh, A. Mukherjee, and K. P. Gummadi, “When the umpire is also a player: Bias in private label product recommendations on e-commerce marketplaces,” in *ACM FAccT*, 2021.
- [15] F. M. Harper and J. A. Konstan, “The movielens datasets: History and context,” *ACM TIS*, vol. 5, no. 4, 2016.
- [16] R. He and J. McAuley, “Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering,” in *WWW*, 2016.
- [17] C. Desrosiers and G. Karypis, “A comprehensive survey of neighborhood-based recommendation methods,” in *Recommender systems handbook*. Springer, 2011.
- [18] G. Linden, B. Smith, and J. York, “Amazon.com recommendations: Item-to-item collaborative filtering,” *IEEE Internet computing*, no. 1, 2003.
- [19] B. M. Sarwar, G. Karypis, J. A. Konstan, J. Riedl *et al.*, “Item-based collaborative filtering recommendation algorithms,” *WWW*, vol. 1, 2001.
- [20] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, “Application of dimensionality reduction in recommender system-a case study,” Minnesota Univ Minneapolis Dept of Computer Science, Tech. Rep., 2000.
- [21] O. Barkan and N. Koenigstein, “Item2vec: neural item embedding for collaborative filtering,” in *IEEE MLSP*, 2016.

- [22] Y. Hu, Y. Koren, and C. Volinsky, "Collaborative filtering for implicit feedback datasets," in *IEEE ICDM*, 2008.
- [23] P. Covington, J. Adams, and E. Sargin, "Deep neural networks for youtube recommendations," in *ACM RecSys*, 2016.
- [24] S. Barocas and A. D. Selbst, "Big data's disparate impact," *Calif. L. Rev.*, vol. 104, p. 671, 2016.
- [25] R. Zemel, Y. Wu, K. Swersky, T. Pitassi, and C. Dwork, "Learning fair representations," in *ICML*, 2013.
- [26] P. Lahoti, K. P. Gummadi, and G. Weikum, "ifair: Learning individually fair data representations for algorithmic decision making," in *ICDE*, 2019.
- [27] D. Pedreschi, S. Ruggieri, and F. Turini, "Discrimination-aware data mining," in *ACM SIGKDD*, 2008.
- [28] D. Pedreschi, S. Ruggieri, and F. Turini, "Measuring discrimination in socially-sensitive decision records," in *SDM*. SIAM, 2009.
- [29] J. Angwin, J. Larson, S. Mattu, and L. Kirchner, "Machine bias," *ProPublica*, May, vol. 23, 2016.
- [30] M. B. Zafar, I. Valera, M. Gomez Rodriguez, and K. P. Gummadi, "Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment," in *WWW*, 2017.
- [31] M. Zehlike, F. Bonchi, C. Castillo, S. Hajian, M. Megahed, and R. Baeza-Yates, "Fa\*ir: A fair top-k ranking algorithm," in *ACM CIKM*, 2017.
- [32] A. Dash, A. Mukherjee, and S. Ghosh, "A network-centric framework for auditing recommendation systems," in *IEEE INFOCOM*, 2019.
- [33] A. Blum, T.-H. H. Chan, and M. R. Rwebangira, "A random-surfer web-graph model," in *Proc. ANALCO*, 2006.
- [34] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," *COMPUTER NETWORKS AND ISDN SYSTEMS*, vol. 30, 1998.
- [35] A. N. Nikolakopoulos and G. Karypis, "Recwalk: Nearly uncoupled random walks for top-n recommendation," in *ACM WSDM*, 2019.
- [36] X. Ning and G. Karypis, "Slim: Sparse linear methods for top-n recommender systems," in *IEEE ICDM*, 2011.
- [37] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *NeurIPS*, 2013.
- [38] M. E. Yaari and M. Bar-Hillel, "On dividing justly," *Springer Social choice and welfare*, 1984.
- [39] M. Joseph, M. Kearns, J. Morgenstern, S. Neel, and A. Roth, "Meritocratic fairness for infinite and contextual bandits," in *AIES*, 2018.
- [40] M. Joseph, M. Kearns, J. H. Morgenstern, and A. Roth, "Fairness in learning: Classic and contextual bandits," in *NeurIPS*, 2016.
- [41] K. Roose, "The making of a youtube radical," <https://nyti.ms/3LwiuDC>, 2019.
- [42] M. Fisher and A. Taub, "How youtube radicalized brazil," <https://nyti.ms/3J8Cqx9>, 2019.
- [43] J. Cook and S. K. Will, "Youtube's conspiracy theory crackdown has one big winner: Fox news," <https://bit.ly/3LMtkpd>, 2019.
- [44] S.-H. Cha, "Comprehensive survey on distance/similarity measures between probability density functions," *City*, vol. 1, no. 2, 2007.
- [45] A. Chakraborty, J. Messias, F. Benevenuto, S. Ghosh, N. Ganguly, and K. Gummadi, "Who makes trends? understanding demographic biases in crowdsourced recommendations," in *AAAI ICWSM*, 2017.
- [46] A. Dash, A. Shandilya, A. Biswas, K. Ghosh, S. Ghosh, and A. Chakraborty, "Summarizing user-generated textual content: Motivation and methods for fairness in algorithmic summaries," in *ACM CSCW*, 2019.
- [47] P. Lahoti, K. P. Gummadi, and G. Weikum, "Operationalizing individual fairness with pairwise fair representations," *arXiv preprint arXiv:1907.01439*, 2019.
- [48] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *ACM SIGKDD*, 2016.
- [49] J. d. Borda, "Mémoire sur les élections au scrutin," *Histoire de l'Académie Royale des Sciences pour 1781 (Paris, 1784)*, 1784.
- [50] Womply, "How online reviews impact small business revenue," <https://bit.ly/3IWWAro>, 2021.
- [51] Metacritic, "How we create the metacore magic," <https://www.metacritic.com/about-metascors>, 2021.



**Abhisek Dash** did his B.Tech. from IIT Bhubaneswar in 2015, M.Tech. from IEST Shibpur in 2017 and currently pursuing Ph.D. in IIT Kharagpur. His research interests lie in investigating unfairness, bias related concerns in online socio-technical systems.



**Abhijnan Chakraborty** is currently an Assistant Professor at Indian Institute of Technology (IIT) Delhi, India. He received M.Tech and Ph.D. from IIT Kharagpur, India. He was a Post-Doctoral Researcher with the Max Planck Institute for Software Systems (MPI-SWS), Germany. His research interests include Social Computing, Information Retrieval, AI & Law, and Responsible AI. Dr. Chakraborty's research has won the Best Paper Award at ASONAM'16 and the Best Poster Award at ECIR'19.



**Saptarshi Ghosh** is currently an Assistant Professor with the Department of Computer Science and Engineering, IIT Kharagpur and the head of a Max Planck Partner Group at IIT Kharagpur. He received his Ph.D. from IIT Kharagpur, India, in 2013. He was a Humboldt Post-Doctoral Fellow with the Max Planck Institute for Software Systems, Saarbrücken, Germany. His research interests include Social network analysis, Legal analytics, and Algorithmic bias and fairness.



**Animesh Mukherjee** is an Associate Professor and A K Singh Chair in the Department of Computer Science and Engineering, Indian Institute of Technology, Kharagpur. His main research interests center around (a) investigation of hate and abusive content on social media platforms, (b) fairness and bias in information retrieval systems, (c) media bias, and (d) quality monitoring of Wikipedia articles. Some of the notable awards that he has received are INAE Young Engineering Award, INSA Medal for Young Scientist, IBM Faculty Award, Facebook AI and Ethics Research Award, Google Tensorflow award, GYTI Award, Humboldt Fellowship for Experienced Researchers.



**Krishna P. Gummadi** is the Head of the Networked Systems Research Group, Max Planck Institute for Software Systems (MPI-SWS), and a Professor with the University of Saarland, Germany. His current research interests include enhancing fairness, accountability, transparency, and explainability of automated (particularly, data-driven and learning-based) decision-making systems. Dr. Gummadi is a recipient of numerous awards, including the ACM SIGCOMM and ICWSM Test-of-Time Awards.