# Graphical Abstract

**Fairness Assessment for Recommender Systems via Situation Test and Item Response Theory**

# Highlights

**Fairness Assessment for Recommender Systems via Situation Test and Item Response Theory**

- A task-agnostic framework for fairness evaluation in recommender systems
- Situation Test Score captures counterfactual fairness robustness
- Item Response Theory models fairness ability, difficulty, and discrimination

# Fairness Assessment for Recommender Systems via Situation Test and Item Response Theory

[a]*RMIT University, Melbourne, Victoria, Australia*

**Abstract**

While existing fairness-aware methods largely focus on group fairness, user-item-side fairness, or multi-sided fairness, they often overlook the diversity of recommendation tasks (e.g., rating prediction, top-N recommendation, or binary recommendation).To address this gap, we introduce SIReA, a framework that conceptualises fairness as the robustness of predictions to changes in sensitive attributes, grounded in Item Response Theory (IRT). The framework first introduces the Situation Test Score (STS), a task-agnostic and context-sensitive metric that captures systematic biases, and then leverages IRT to jointly estimate: 1) the *fairness-related ability* of each recommendation model; 2) the *discrimination* of tasks in exposing fairness disparities; and 3) the *difficulty* of achieving fairness across individual instances. The experiment results on three benchmark datasets with multiple recommendation settings show 1) accuracy does not imply fairness, as models with strong predictive accuracy may still exhibit fairness gaps; 2) rating prediction is more effective than top-N or binary recommendations in exposing fairness disparities. These findings suggest that fairness evaluation requires task- and attribute-specific analysis rather than relying on a single metric or setting. The source code can be found at `https://anonymous.4open.science/r/SIReA-71F3`.

*Keywords:* Fairness, Recommender Systems, Item Response Theory.

## 1. Introduction

Fairness in recommender systems is a key research focus due to the growing influence of algorithmic decision-making in society. Early studies address bias using fairness notions such as demographic parity and equal opportunity [1, 2, 3], often operationalised through group-based metrics over sensitive attributes such as gender or race [4]. Subsequent work extends

these ideas to user-side fairness, emphasising equitable recommendation quality [5], and item-side fairness, which promotes balanced exposure for items or providers [6]. Multi-sided fairness also attracts attention, highlighting the need to balance competing interests among users, producers, and advertisers [7]. Despite these advances, most fairness metrics remain task-specific and rely on aggregate comparisons, which often fail to capture individualised concerns. In particular, the difficulty of generating fair and accurate recommendations varies significantly across user-task instances (e.g., how much a user likes a specific item in rating prediction), yet there is limited understanding of how to quantify this difficulty, which is essential for identifying user–task interactions that are inherently harder to treat fairly and for distinguishing algorithmic shortcomings from intrinsic fairness challenges.

One promising direction lies in the use of Item Response Theory (IRT), a well-established framework from psychometrics that models individual behaviour with respect to user-task characteristics. Building on this idea, an IRT-based framework for implicit recommendation was proposed [8]; however, the framework is restricted to binary outcomes. To address this limitation, based on beta-IRT that offers greater flexibility in modelling continuous responses, [9] introduced Bi-ReIRT to a bi-directional evaluation process on the dataset level. More recently, Fair-IRT [10] was proposed, which applies IRT-based ability estimation to evaluate the fairness of general predictive models. Nevertheless, a key question remains: *how can IRT models be generalised to evaluate fairness in a unified manner across diverse recommendation tasks at the individual user-task level?*

To fill this gap, we first introduce Situation Test Score (STS), then propose Situation test-based IRT for Recommendation fairness Assessment (SIReA) framework to model fairness disparities using beta-IRT. STS supports context-sensitive fairness assessment by comparing model predictions under original and perturbed sensitive attributes, thereby identifying disparities at the individual user-task level. These disparities are then modelled using beta-IRT, which estimates three fairness-relevant parameters: 1) the *ability* of each recommendation model to produce fair outcomes; 2) the *discrimination* of each user-task instance in revealing fairness sensitivity; and 3) the *difficulty* of achieving fairness for specific interactions.

Our main contributions are as follows: 1) a set of STS-based fairness metrics that are adaptable to diverse recommendation tasks; (2) the SIReA framework, which leverages STS and beta-IRT to generate interpretable fairness parameters: model ability, user-task discrimination, and fairness diffi-

culty; 3) a comprehensive analysis showing that accuracy does not necessarily imply fairness, as models with high predictive performance may still show disparities; rating prediction is more effective than top-N or binary tasks in revealing fairness issues.

## 2. Related Work

Fairness in recommendation has been studied from multiple perspectives, including user, item, group, and individual level [11, 2]. Biases may stem from factors such as popularity [12], demographics, or exposure [13, 14, 15]. While several fairness evaluation frameworks have been proposed, most remain task-specific and group-focused. For example, PyCPFair targets group disparities in Top-N recommendation [16], FairRec considers two-sided fairness through pairwise list comparisons [17], and LibRec-AutoFair supports fairness analysis for Top-N and rating prediction using diversity and exposure metrics [18]. However, these approaches lack individual-level interpretability. Although disparity-based metrics offer task-agnostic evaluation, they remain limited in interpretability [19, 20].

Beyond group-based measures (on sensitive attributes, e.g. gender), other strands of research have proposed individual-level approaches. Counterfactual fairness ensures predictions remain invariant under hypothetical changes to sensitive attributes [21], while adversarial fairness models enforce invariance by jointly training a predictor and an adversary that attempts to recover sensitive information from learned representations [22, 23]. Both lines of work intervene directly in the training process to enforce fairness, whereas our framework differs by providing a post hoc, task-agnostic evaluation methodology that can be applied to any trained recommender model.

In contrast, IRT has recently gained attention for fairness modelling: Chen et al. [24] introduce IRT to capture difficulty and discrimination in recommendation, and Xu et al [10] extend beta-IRT for assessing fairness in predictive models. Yet, no existing framework unifies fairness assessment across recommendation tasks while offering interpretable insights, a gap this work aims to address.

## 3. The Proposed SIReA

An overview of the proposed SIReA framework is shown in Figure 1. Given a recommendation task with multiple base models, we first define STS-based fairness metrics to evaluate each user-task instance. These fairness

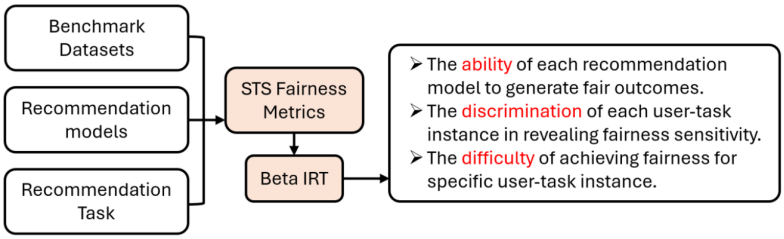



Figure 1: Overview of the proposed SIReA framework.

scores are then formatted into a response matrix suitable for the Beta-IRT model. By training the model on this matrix, we estimate three fairness-related parameters (i.e., ability, discrimination and difficulty). The remainder of this section proceeds as follows: we introduce the STS-based fairness metrics across different recommendation tasks, describe the Beta-IRT model, and then summarise the complete SIReA workflow.

### *3.1. STS-based Fairness Metrics*

Let the dataset be $\mathcal{D} = \{A, X, Y\}$, where $A$ denotes sensitive attributes such as gender, $X$ represents other input features, and $Y$ is the target recommendation output. Recommendation models are trained on $\mathcal{D}$ and evaluated by generating two sets of predictions: $\hat{y} = Y(A, X)$ and $\hat{y}' = Y(A', X)$, where $A'$ is derived by flipping the sensitive attribute. Fairness is assessed by comparing $\hat{y}$ and $\hat{y}'$ using task-specific fairness metrics that reflect how sensitive the model's outputs are to changes in sensitive attributes.

This instance-level fairness assessment is closely related to the concept of the STS, which is originally introduced to assess fairness in job recruitment processes [25]. In this context, applicants are identical in all attributes except for a sensitive one, such as race. STS is used to determine whether the selection process treats applicants differently due to the sensitive attribute, thereby identifying potential biases. This concept is later extended and used as a fairness score in various studies [26, 27, 28, 29]. Following this line of work, we propose three task-specific fairness metrics based on STS, tailored to different recommendation scenarios.

**Definition 1.** The STS for the Rating prediction task is defined as:

$$\mathrm{STS}_{\mathrm{Rating}} = 1 - \frac{|\hat{Y}(A, X) - \hat{Y}(A', X)|}{\hat{Y}(A, X)}, \tag{1}$$

**Definition 2.** The STS for the Top-N recommendation task is defined as:

$$\mathrm{STS}_{\mathrm{Top-N}} = 1 - \left|\mathrm{NDCG}(A) - \mathrm{NDCG}(A')\right|, \tag{2}$$

where $\mathrm{NDCG}(A)$ denotes the normalised discounted cumulative gain calculated based on the ranked list produced by the model when using the original sensitive attribute $A$, and $\mathrm{NDCG}(A')$ is the corresponding value obtained when the attribute is perturbed to $A'$. This metric captures the stability of the recommendation ranking with respect to changes in sensitive attributes.

While NDCG is used as an example, other metrics such as Mean Average Precision, Recall, and MRR can be used similarly. They are computed from model outputs under original and perturbed attributes, and their differences indicate fairness. The choice of metric depends on the task.

**Definition 3.** The STS for Binary recommendation task (e.g. "like" vs. "dislike") is defined as:

$$\mathrm{STS}_{\mathrm{Binary}} = 1 - \left|\hat{P}(A, X) - \hat{P}(A', X)\right|, \tag{3}$$

where $\hat{P}(A, X)$ denotes the predicted probability of a "like" under the original sensitive attribute, and $\hat{P}(A', X)$ is the corresponding probability after perturbation.

Each fairness metric yields a score in the range $[0, 1]$. For a given recommendation task, one metric is selected, and the resulting scores are compiled into a matrix where each row corresponds to an individual user-task instance. This matrix is then used as input for downstream IRT-based analysis.

### *3.2. Beta-IRT*

IRT models the interaction between an individual and a question by estimating three latent parameters: the individual's ability, the difficulty and the discrimination of the question. Let $p_{ij}$ denote the binary response of individual $i$ to question $j$. In the logistic IRT model, the probability of a correct response ($p_{ij} = 1$) is defined using a logistic function parameterised by $\delta_j$ (difficulty) and $a_j$ (discrimination). The response variable $p_{ij}$ is modelled as a Bernoulli random variable:

$$p_{ij} \sim \mathrm{Bernoulli}(x_{ij}). \tag{4}$$

The expected value of this response, defined by the logistic Item Characteristic Curve (ICC), is:

$$\mathbb{E}[p_{ij} \mid \theta_i, \delta_j, a_j] = x_{ij} = \frac{1}{1 + e^{-a_j(\theta_i - \delta_j)}}, \tag{5}$$

where $\theta_i$ represents the ability of individual $i$; $\delta_j$ controls the location of the ICC, indicating the difficulty of question $j$; and $a_j$ controls the steepness of the curve, indicating the discrimination power of the question. Together, these parameters determine the probability of a correct response.

Classical logistic IRT is restricted to binary outcomes, which limits its applicability in settings where responses are continuous. To address this limitation, the Beta-IRT model [24] extends IRT to continuous-valued responses in the interval $[0, 1]$. In this model, the response $p_{ij}$ is drawn from a Beta distribution with shape parameters $\alpha_{ij}$ and $\beta_{ij}$, defined as:

$$p_{ij} \sim \text{Beta}(\alpha_{ij}, \beta_{ij}),\ \alpha_{ij} = \left(\frac{\theta_i}{\delta_j}\right)^{a_j},\ \beta_{ij} = \left(\frac{1 - \theta_i}{1 - \delta_j}\right)^{a_j}. \tag{6}$$

The parameters $\theta_i$, $\delta_j$, and $a_j$ retain the same interpretations as in the logistic case. However, by modelling continuous responses directly, Beta-IRT avoids discretisation of fairness scores, preserves information, and allows ICCs to assume more flexible forms—including sigmoidal, parabolic, and anti-sigmoidal shapes—that better capture heterogeneous response behaviours. Empirical evidence from prior work [24] further demonstrates that Beta-IRT generalises classical IRT and yields improved predictive performance, which motivates its adoption in this study.

### *3.3. Workflow*

To assess fairness in recommendation systems, we utilise the STS-based fairness metrics and the Beta-IRT model described above. Let $M$ denote the number of user-task instances, and $N$ the number of recommendation models. Each model generates predictions for all $M$ instances in the dataset $D$. For each instance where model $N_i$ is applied to instance $M_i$, we compute the STS using the appropriate metric defined in the previous section. Specifically, we use Equation 1 for rating prediction, Equation 2 for top-N recommendation, and Equation 3 for binary recommendation.

By computing the STS for each $(N_i, M_i)$ pair, we construct an $N \times M$ response matrix. This matrix serves as input to the Beta-IRT, which estimates the IRT parameters for each recommendation model and instance.

The resulting ability, difficulty, and discrimination parameters provide an interpretable assessment of fairness.

## 4. Interpreting Parameters with a Simulated Dataset

To demonstrate the interpretability of the three parameters—discrimination, difficulty, and ability—we constructed a simulated dataset comprising 20 recommendation algorithms and 50 user–item pairs. Algorithm abilities were sampled from 0.1 to 0.9, representing varying levels of fairness capacity. Discrimination values were drawn from –1 to 2, where negative values denote atypical cases and positive values indicate increasing sensitivity. Difficulty values ranged from 0.1 to 1, reflecting how challenging a user–item pair is to treat fairly.

Using these sampled values, we computed the IRT input matrix for each algorithm and user–item pair following Equation 6. The resulting Situation Test Scores (STS) were used to train the model, and the estimated parameters were compared with the ground-truth values to evaluate whether the model recovers the intended behaviours.

### *4.1. Individual Parameters: Discrimination and Difficulty*

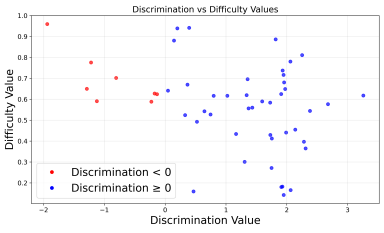
Fig. 2a shows the trained parameters span the same ranges as those in the simulation, confirming that the model successfully recovers the designed variation.

#### *4.1.1. Discrimination*

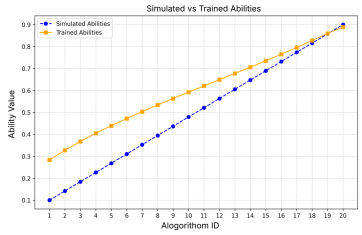
The discrimination parameter $a$ measures how sensitively fairness scores change with model ability. When $a < 0$, it represents atypical inverse behaviour in which fairness decreases as ability increases (red points in Fig. 2a; ICC curves in Fig. 2c). For $0 < a < 1$, discrimination is weak, yielding a gradual, anti-sigmoid curve with only modest changes in fairness (Fig. 2d). By contrast, values of $a > 1$ indicate strong discrimination, producing a sharp, standard S-shaped ICC where fairness scores rise steeply with increasing ability (Fig. 2e).

#### *4.1.2. Difficulty*

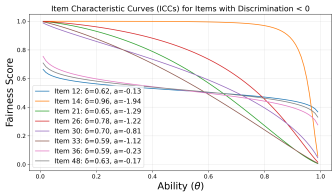
The difficulty parameter $\delta \in [0, 1]$ indicates how challenging it is for a recommendation algorithm to treat a given user–task pair fairly. Low values correspond to easy scenarios, while high values denote more demanding ones.



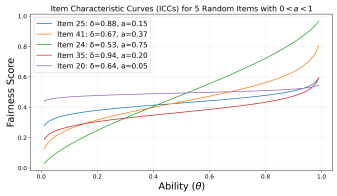

(a) Discrimination vs. difficulty



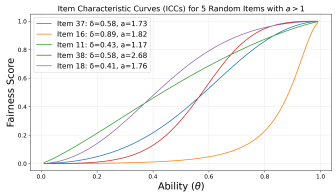

(b) Ability distributions per algorithm




(c) $a_j < 0$




(d) $0 < a_j < 1$




(e) $a_j > 1$

Figure 2: Effects of discrimination parameter values in the SIReA framework, with additional ability distributions per algorithm.

As shown in Figure 2e, tasks with $\delta$ near 1 produce low fairness scores across most of the ICC curve, even for high-ability algorithms, whereas tasks with $\delta$ near 0 sustain high scores regardless of model ability. These results closely mirror the values assigned in the simulation, confirming that the estimated difficulty parameter faithfully represents the intended fairness structure.

### *4.2. Predictive Model Parameter: Ability*

The ability parameter $\theta$ quantifies the fairness capacity of each model. Figure 2b compares the simulated and estimated values, showing close agreement and preserving the relative ranking of algorithms. Minor deviations appear for some cases, but overall the estimated abilities reliably approximate the simulated ground truth, demonstrating that $\theta$ is an effective indicator of fairness capability in this setting.

## 5. Experiments

### *5.1. Experimental Setup*

We evaluate fairness across three common recommendation tasks using the proposed **SIReA** framework: rating prediction, top-N recommendation,

Table 1: Execution times (s) for training recommendation models and the Beta-IRT model across datasets.

| Dataset | Task | Env. | Training size | IRT matrix size | Time (s) |
|---|---|---|---|---|---|
| Book-Crossing | Rec Models | GPU (T4) | 628,508 | – | 315 |
| | Beta-IRT | CPU (i7-8565U) | – | $157,127 \times 5$ | 324 |
| Last.fm (LSM) | Rec Models | GPU (T4) | 809,232 | – | 293 |
| | Beta-IRT | CPU (i7-8565U) | – | $202,308 \times 5$ | 314 |
| MovieLens-100K | Rec Models | CPU (i7-8565U) | 80,000 | – | 128 |
| | Beta-IRT | CPU (i7-8565U) | – | $20,000 \times 5$ | 273 |

*Note.* Rec Models time is the average training time across models. For Beta-IRT, we report rating prediction training time (binary is similar, Top-N is faster). Experiments were run on a local CPU (Intel i7-8565U, 16 GB RAM) and on Google Colab Pro with an NVIDIA Tesla T4 GPU (15 GB GPU RAM, 12.7 GB system RAM).

and binary recommendation. Experiments are conducted on three benchmark datasets: **MovieLens-100K (ML)**, **Last.fm (LSM)**, and **Book-Crossing (BK)**. The ML dataset contains explicit ratings on a 1–5 scale for user–movie pairs, along with demographic attributes such as gender, age, and occupation. The LSM dataset provides implicit feedback through user play counts on music tracks, as well as demographic information, including gender and country. Since explicit ratings are not available, user preferences are inferred from play count frequencies. For the binary recommendation task, a track is considered *liked* if it appears among a user's top three most-played items. The BK dataset contains ratings on a 1–10 scale for user–book pairs, together with demographic information such as age. All datasets are partitioned into 80% training and 20% testing splits. In our experiments, we consider gender and age as sensitive attributes. For the age-based analysis, users are categorised as *young* or *old* based on a predefined age threshold.

To evaluate model fairness, we select five knowledge-aware recommendation models: CFKG [30], CKE [31], KGAT [32], KGCN [33], and KGIN [34]. These models were deliberately chosen because our fairness experiments involve demographic attributes (e.g., gender, age, country), which can only be meaningfully incorporated in knowledge-aware algorithms that integrate side information into the recommendation process. Together, they represent the main categories of knowledge-aware approaches: embedding-based methods (CFKG, CKE), graph-based neural models (KGAT, KGCN), and a hybrid wide-and-deep architecture (KGIN). They are also well-established, widely cited, and frequently used as baselines in recommender system research, ensuring both state-of-the-art performance coverage and comparability with prior work. Each model is applied to the three tasks described above, and its fairness is evaluated using the corresponding STS-based metrics.

Table 2: Recommendation vs. fairness ability across tasks under **gender-based** (MovieLens = ML, Last.fm = LSM) and **age-based** (MovieLens = ML, Book = BK) evaluation. LSM omitted for age due to missing attribute.

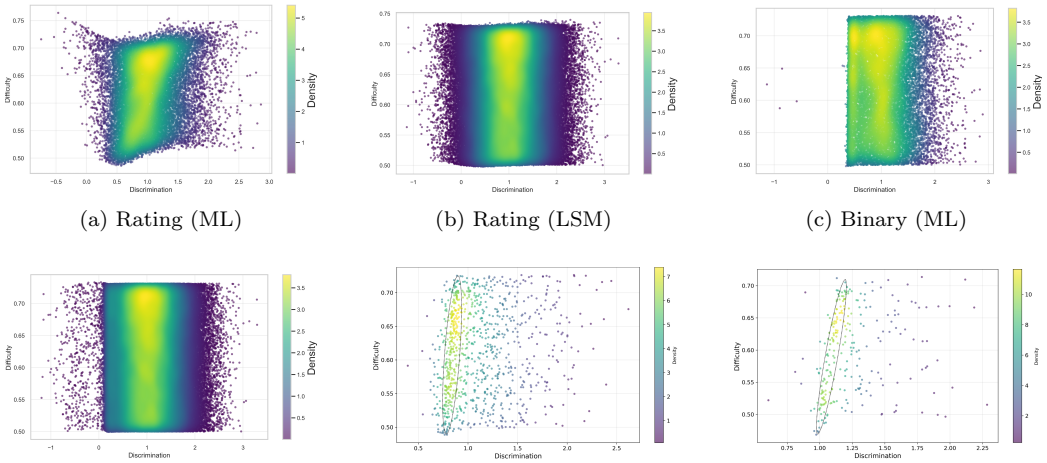
| | **Rating Prediction** | | | | | | | | **Top-N Recommendation** | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Gender-based | | | | Age-based | | | | Gender-based | | | | Age-based | | | |
| **Model** | RMSE | | Ability | | RMSE | | Ability | | NDCG | | Ability | | NDCG | | Ability | |
| | ML | LSM | ML | LSM | ML | BK | ML | BK | ML | LSM | ML | LSM | ML | BK | ML | BK |
| CFKG | 0.941 | 1.356 | 0.845 | 0.887 | 0.945 | 3.669 | 0.866 | 0.867 | 0.758 | 0.630 | 0.997 | 0.994 | 0.935 | 0.604 | 0.997 | 0.999 |
| CKE | 0.940 | 1.441 | 0.846 | 0.835 | 0.940 | 3.973 | 0.770 | 0.744 | 0.759 | 0.632 | 0.996 | 0.994 | 0.889 | 0.586 | 0.996 | 0.999 |
| KGAT | 0.941 | 1.433 | 0.825 | 0.841 | 0.943 | 3.935 | 0.816 | 0.672 | 0.758 | 0.603 | 0.998 | 0.993 | 0.936 | 0.587 | 0.998 | 0.999 |
| KGCN | 0.939 | 1.466 | 0.781 | 0.884 | 0.949 | 3.578 | 0.689 | 0.714 | 0.761 | 0.600 | 0.996 | 0.993 | 0.926 | 0.587 | 0.996 | 0.999 |
| KGIN | 0.957 | 1.353 | 0.983 | 0.869 | 0.946 | 3.972 | 0.842 | 0.744 | 0.756 | 0.634 | 0.995 | 0.993 | 0.937 | 0.587 | 0.995 | 0.999 |

| | **Binary Recommendation** | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **Gender-based** | | | | | | **Age-based** | | | | | |
| **Model** | Precision | | Recall | | Ability | | Precision | | Recall | | Ability | |
| | ML | LSM | ML | LSM | ML | LSM | ML | BK | ML | BK | ML | BK |
| CFKG | 0.736 | 0.302 | 0.735 | 0.252 | 0.999 | 0.999 | 0.736 | 0.479 | 0.735 | 0.527 | 0.999 | 0.999 |
| CKE | 0.725 | 0.321 | 0.770 | 0.243 | 0.999 | 1.000 | 0.725 | 0.482 | 0.770 | 0.472 | 0.999 | 0.999 |
| KGAT | 0.729 | 0.346 | 0.749 | 0.214 | 0.999 | 0.999 | 0.729 | 0.497 | 0.749 | 0.465 | 0.999 | 0.999 |
| KGCN | 0.715 | 0.333 | 0.785 | 0.211 | 0.999 | 0.999 | 0.715 | 0.491 | 0.785 | 0.516 | 0.999 | 0.999 |
| KGIN | 0.706 | 0.299 | 0.812 | 0.436 | 0.999 | 0.999 | 0.706 | 0.483 | 0.812 | 0.562 | 0.999 | 0.999 |

In addition to fairness performance, we also measured execution times to examine the computational efficiency of SIReA. Table 1 summarises the average training times for the recommendation models and for the Beta-IRT model across the three datasets.

### *5.2. Fairness Assessment across Tasks and Sensitive Attributes*

We begin by evaluating the fairness performance of the five recommendation models across three different tasks. As shown in Table 2, the following observations are made: (1) *Accuracy does not imply fairness*: A model's recommendation accuracy does not necessarily reflect its fairness ability. For example, KGCN achieves strong accuracy on the rating prediction task (RMSE = 0.939), yet exhibits relatively low fairness (ability = 0.781) on the ML dataset. This indicates that high predictive accuracy does not guarantee fair treatment of individual users or groups. (2) *Rating prediction reveals fairness disparities more effectively*: Among the three tasks, rating prediction appears more effective in differentiating the fairness capabilities of various models. Specifically, on ML, fairness ability scores for rating prediction range from 0.781 to 0.983, and from 0.835 to 0.887 on LSM. In contrast, the scores for binary and top-N recommendation tasks are highly concentrated: between 0.999 and 1.0 for binary recommendation, and around 0.997 for top-N recommendation.(3) *Fairness ability depends on the sensitive attribute*: The ranking of models by fairness ability is not consistent across different sensi-




(a) Rating (ML) (b) Rating (LSM) (c) Binary (ML)

(d) Binary (LSM) (e) Top-N (ML) (f) Top-N (LSM)

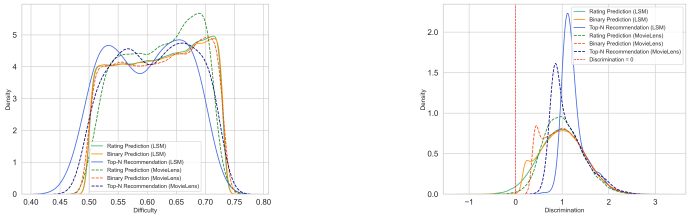
Figure 3: Density plots of discrimination versus difficulty.

tive attributes. For instance, in the ML dataset, KGIN achieves the highest fairness ability under gender-based evaluation, whereas CFKG performs best under age-based evaluation. Similarly, other models also shift in their relative rankings depending on whether gender or age is considered as the sensitive attribute. This finding highlights that fairness assessments are highly sensitive to the choice of protected attribute, underscoring the importance of evaluating models against multiple demographic dimensions to obtain a more comprehensive view of fairness.

Taken together, these findings suggest that while top-N recommendation is widely adopted in practice due to its relevance to user consumption patterns, its simpler output structure may obscure fairness disparities. In contrast, the finer granularity of rating prediction outputs provides a more sensitive lens through which to assess fairness. Moreover, the results demonstrate that fairness is not solely determined by accuracy and is highly contingent on the choice of sensitive attribute.

### *5.3. Difficulty vs Discrimination of Instances*

We further examine the difficulty and discrimination of instances across the three recommendation tasks. Figure 3 presents scatter plots of difficulty versus discrimination for each task-dataset combination, while Figure 4 shows the corresponding distributions. The analysis yields the following insights: (1) *Variation across tasks*: The relationship between difficulty and discrimi-



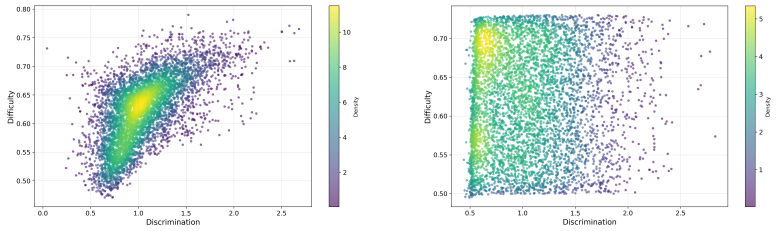

(a) Difficulty (b) Discrimination

Figure 4: Distributions of difficulty and discrimination.

nation differs significantly by task. In rating prediction, instances are widely dispersed, suggesting that fairness challenges are diverse. Binary recommendation shows a flatter pattern, and top-N recommendation reveals a tighter, more compact distribution. By utilising the Kolmogorov-Smirnov (KS) test, we find statistically significant differences between all task pairs in terms of their difficulty and discrimination. For instance, on the ML dataset, the KS test comparing difficulty distributions between rating prediction and binary recommendation yields D = 0.055 with p ¡ 0.0001, confirming a significant difference. (2) *Difficulty varies more than discrimination*: Difficulty values span a wider range than discrimination values. This reflects the presence of varied fairness challenges across instances. In contrast, discrimination values tend to follow a normal-like distribution, which may be attributed to assumptions made during the IRT training process. (3) *Negative discrimination in binary recommendation*: A considerable number of instances in binary recommendation exhibit negative discrimination values. This indicates that higher predictive ability may be associated with less fair treatment in certain instances. One possible explanation is the limited expressiveness of binary feedback, which lacks the detailed preference information available in rating or ranking-based tasks. (4) *Clustering in top-N recommendation*: In the top-N task, we observe a notable cluster (indicated as dotted ovals in Fig. 3e and 3f) of instances with discrimination values around 0.75 and difficulty values between 0.48 and 0.75 on ML. This cluster, consisting of around 27% of the instances, does not appear in the other tasks and suggests distinct fairness characteristics unique to top-N recommendation.

### *5.4. Analysis of Clustered Instances in Top-N Recommendation*

As shown in Figure 3e and Figure 3f, a narrow cluster of instances appears in the Top-N recommendation task, with discrimination values concen-




(a) Rating Prediction (b) Binary Recommendation

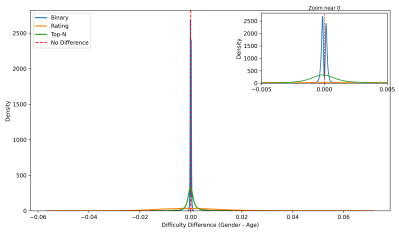
Figure 5: Comparison of discrimination and difficulty for clustered instances across the rating prediction and binary recommendation tasks.

trated around 0.7. This suggests consistent model sensitivity across items for these users. To further investigate, we isolate these users and evaluate their fairness characteristics in the rating prediction and binary recommendation tasks (Figure 5a and Figure 5b) on the ML dataset. We observe: (1) In rating prediction, the users form a broader and more dispersed cluster; although discrimination remains similar, difficulty values vary more, indicating increased fairness complexity. (2) In binary recommendation, these users are more scattered along the discrimination axis, with a small vertical cluster still present, suggesting slightly greater model sensitivity than Top-N recommendation but less than rating prediction. Overall, the results confirm that rating prediction better reveals fairness disparities than binary or Top-N tasks.

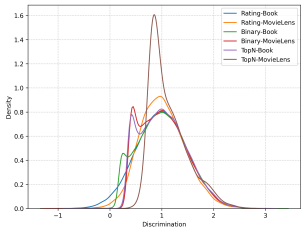
### *5.5. Comparative Analysis of Sensitive Attributes and Datasets*

First, we examined how recommendation tasks are influenced by different sensitive attributes within the same dataset (MovieLens), comparing age and gender. The results show that rating prediction is most affected by attribute changes (Fig. 6a): while both rating prediction and top-N recommendation display comparable disparities in discrimination, rating prediction exhibits the largest disparity in difficulty. This indicates that fairness in rating prediction is particularly sensitive to the choice of sensitive attribute, making it a more discriminative setting for fairness evaluation than binary or top-N tasks. Kolmogorov-Smirnov (KS) tests confirm that these disparities are statistically significant.

Next, we investigated how recommendation tasks are affected by the same sensitive attribute across different datasets by comparing MovieLens and Book. The results reveal that top-N recommendation exhibits the highest




(a) Difficulty difference (Gender – Age) in MovieLens.




(b) Discrimination distributions across MovieLens and Book datasets.

Figure 6: Comparison of fairness across tasks, attributes, and datasets. (a) shows density differences in MovieLens (gender vs. age). (b) shows discrimination distribution comparisons across MovieLens and Book datasets.

disparity across datasets (Fig. 6b): both difficulty and discrimination vary most strongly in the top-N setting, whereas rating prediction shows moderate sensitivity and binary recommendation appears least affected. KS tests again indicate statistically significant differences.

Overall, these findings demonstrate that fairness evaluations are jointly shaped by the sensitive attribute, the dataset, and the recommendation task. Rating prediction is most vulnerable to attribute changes, while top-N recommendation is most sensitive to dataset shifts, underscoring the importance of conducting fairness assessments across multiple attributes and datasets.

## 6. Conclusion

In this paper, we present SIReA, a novel fairness assessment framework for recommender systems that combines Situation Test Scores with beta-IRT modelling. By evaluating model predictions under original and perturbed sensitive attributes, SIReA quantifies fairness at the individual user–task level and estimates three interpretable parameters: model ability, user–task discrimination, and fairness difficulty. Our experiments across diverse recommendation tasks and real-world datasets demonstrate that SIReA effectively captures nuanced fairness disparities that conventional metrics often overlook, though it currently focuses on single-attribute fairness in a post hoc setting and has been primarily evaluated on knowledge-aware models due to data availability. The framework thus offers a task-agnostic approach to fairness assessment, providing interpretable insights for both fairness diagnostics and the design of fairer recommender systems.

## References


[1] T. Kamishima, S. Akaho, H. Asoh, and J. Sakuma, "Fairness-aware classifier with prejudice remover regularizer," in *Machine Learning and Knowledge Discovery in Databases*, pp. 35–50, Springer, 2011.

[2] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. Zemel, "Fairness through awareness," in *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, pp. 214–226, ACM, 2012.

[3] M. Hardt, E. Price, and N. Srebro, "Equality of opportunity in supervised learning," in *Advances in Neural Information Processing Systems*, vol. 29, pp. 3315–3323, 2016.

[4] R. Burke, "Multisided fairness for recommendation," in *Proceedings of the 2017 Workshop on Fairness, Accountability, and Transparency in Machine Learning (FATML)*, (Halifax, Canada), 2017. arXiv:1707.00093.

[5] H. Yoo, Z. Zeng, J. Kang, R. Qiu, D. Zhou, Z. Liu, F. Wang, C. Xu, E. Chan, and H. Tong, "Ensuring user-side fairness in dynamic recommender systems," *arXiv preprint arXiv:2308.15651*, 2023.

[6] H. Wu, B. Mitra, C. Ma, F. Diaz, and X. Liu, "Joint multisided exposure fairness for recommendation," in *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1311–1321, ACM, 2022.

[7] H. Abdollahpouri and R. Burke, "Multi-stakeholder recommendation and its connection to multi-sided fairness," *arXiv preprint arXiv:1907.13158*, 2019.

[8] Y. Liu, A. Medlar, and D. Głowacka, "What we evaluate when we evaluate recommender systems: Understanding recommender systems' performance using item response theory," in *Proceedings of the 17th ACM Conference on Recommender Systems (RecSys '23)*, (Singapore, Singapore), pp. 658–670, ACM, 2023.

[9] Z. Xu, C. Ma, Y. Ren, J. Chan, W. Shao, and F. Xia, "Towards better evaluation of recommendation algorithms with bi-directional item

response theory," in *Companion Proceedings of the ACM on Web Conference 2025*, (New York, NY, USA), p. 1455–1459, 2025.

[10] Z. Xu, S. Kandanaarachchi, C. S. Ong, and E. Ntoutsi, "Fairness evaluation with item response theory," in *Proceedings of the ACM on Web Conference 2025*, pp. 2276–2288, 2025.

[11] R. Burke, G. Adomavicius, T. Bogers, T. Di Noia, D. Kowald, J. Neidhardt, Ö. Özgöbek, M. S. Pera, N. Tintarev, and J. Ziegler, "De-centering the (traditional) user: Multistakeholder evaluation of recommender systems," *Preprint submitted to Elsevier*, 2025. arXiv:2501.05170.

[12] A. Klimashevskaia, D. Jannach, M. Elahi, and C. Trattner, "A survey on popularity bias in recommender systems," *User Modeling and User-Adapted Interaction*, vol. 34, no. 3, pp. 1777–1834, 2024.

[13] J. Chen, H. Dong, X. Wang, F. Feng, M. Wang, and X. He, "Bias and debias in recommender system: A survey and future directions," *ACM Transactions on Information Systems*, vol. 1, pp. 1–38, Dec. 2020.

[14] J. Li, K. Deng, J. Li, and Y. Ren, "Session-oriented fairness-aware recommendation via dual temporal convolutional networks," *IEEE Trans. Knowl. Data Eng.*, vol. 37, no. 2, pp. 923–935, 2025.

[15] J. Li, Y. Ren, and K. Deng, "Fairgan: Gans-based fairness-aware learning for recommendations with implicit feedback," in *Proceedings of the ACM Web Conference 2022*, WWW '22, (New York, NY, USA), p. 297–307, Association for Computing Machinery, 2022.

[16] M. Naghiaei, H. A. Rahmani, and Y. Deldjoo, "Pycpfair: A framework for consumer and producer fairness in recommender systems," *Software Impacts*, vol. 13, p. 100382, 2022.

[17] 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 *Proceedings of The Web Conference 2020 (WWW '20)*, (Taipei, Taiwan), pp. 1194–1204, ACM, 2020.

[18] N. Sonboli, R. Burke, Z. Liu, and M. Mansoury, "Fairness-aware recommendation with librec-auto," in *Proceedings of the 14th ACM Conference on Recommender Systems (RecSys '20)*, pp. 594–596, ACM, 2020.

[19] S. Yao and B. Huang, "Beyond parity: Fairness objectives for collaborative filtering," in *Advances in Neural Information Processing Systems (NeurIPS 2017)*, pp. 2921–2930, 2017.

[20] Y. Li, H. Chen, Z. Fu, Y. Ge, and Y. Zhang, "User-oriented fairness in recommendation," in *Proceedings of the Web Conference 2021 (WWW '21)*, pp. 490–500, ACM, 2021.

[21] M. J. Kusner, J. Loftus, C. Russell, and R. Silva, "Counterfactual fairness," in *Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS)*, pp. 4066–4076, Curran Associates Inc., 2017.

[22] H. Edwards and A. Storkey, "Censoring representations with an adversary," in *Proceedings of the International Conference on Learning Representations (ICLR) Workshop*, 2016.

[23] B. H. Zhang, B. Lemoine, and M. Mitchell, "Mitigating unwanted biases with adversarial learning," in *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pp. 335–340, ACM, 2018.

[24] Y. Chen, T. Silva Filho, R. B. C. Prudêncio, T. Diethe, and P. Flach, "3-irt: A new item response model and its applications," in *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS)*, vol. 89 of *Proceedings of Machine Learning Research*, pp. 991–1000, PMLR, 2019.

[25] M. Bendick, "Situation testing for employment discrimination in the united states of america," *Horizons stratégiques*, vol. 5, no. 3, pp. 17–39, 2007.

[26] L. Zhang, Y. Wu, and X. Wu, "Situation testing-based discrimination discovery: A causal inference approach," in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI 2016)*, pp. 2718–2724, IJCAI/AAAI Press, 2016.

[27] J. M. Álvarez and S. Ruggieri, "Counterfactual situation testing: Uncovering discrimination under fairness given the difference," in *Proceedings of the 3rd ACM Conference on Equity and Access in Algorithms, Mechanisms, and Optimization (EAAMO '23)*, pp. 2:1–2:11, 2023.

[28] L. T. Binh, S. Ruggieri, and F. Turini, "k-nn as an implementation of situation testing for discrimination discovery and prevention," in *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '11)*, pp. 502–510, ACM, 2011.

[29] Z. Xu, Z. Xu, J. Liu, D. Cheng, J. Li, L. Liu, and K. Wang, "Assessing classifier fairness with collider bias," in *Advances in Knowledge Discovery and Data Mining – 26th Pacific-Asia Conference, PAKDD 2022*, vol. 13281 of *Lecture Notes in Computer Science*, pp. 262–276, Springer, 2022.

[30] F. Zhang, N. J. Yuan, D. Lian, X. Xie, and W.-Y. Ma, "Collaborative knowledge base embedding for recommender systems," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 353–362, ACM, 2016.

[31] F. Zhang, N. J. Yuan, D. Lian, X. Xie, and W.-Y. Ma, "Collaborative knowledge base embedding for recommender systems," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 353–362, ACM, 2016.

[32] X. Wang, X. He, Y. Cao, M. Liu, and T.-S. Chua, "Kgat: Knowledge graph attention network for recommendation," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '19)*, pp. 950–958, ACM, 2019.

[33] H. Wang, M. Zhao, X. Xie, W. Li, and M. Guo, "Knowledge graph convolutional networks for recommender systems," in *Proceedings of the 2019 World Wide Web Conference (WWW '19)*, pp. 3307–3313, ACM, 2019.

[34] H.-T. Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, W. Chai, M. Ispir, *et al.*, "Wide & deep learning for recommender systems," *arXiv preprint arXiv:1606.07792*, 2016.