Human-Centric Intelligent Systems (2022) 2:81–94
https://doi.org/10.1007/s44230-022-00008-w

RESEARCH ARTICLE

Achieving User-Side Fairness in Contextual Bandits

Wen Huang[1] · **Kevin Labille**[1] · **Xintao Wu**[1] · **Dongwon Lee**[2] · **Neil Heffernan**[3]

Received: 15 March 2022 / Accepted: 30 August 2022 / Published online: 14 September 2022

Abstract
Personalized recommendation based on multi-arm bandit (MAB) algorithms has shown to lead to high utility and efficiency as it can dynamically adapt the recommendation strategy based on feedback. However, unfairness could incur in personalized recommendation. In this paper, we study how to achieve user-side fairness in personalized recommendation. We formulate our fair personalized recommendation as a modified contextual bandit and focus on achieving fairness on the individual whom is being recommended an item as opposed to achieving fairness on the items that are being recommended. We introduce and define a metric that captures the fairness in terms of rewards received for both the privileged and protected groups. We develop a fair contextual bandit algorithm, Fair-LinUCB, that improves upon the traditional LinUCB algorithm to achieve group-level fairness of users. Our algorithm detects and monitors unfairness while it learns to recommend personalized videos to students to achieve high efficiency. We provide a theoretical regret analysis and show that our algorithm has a slightly higher regret bound than LinUCB. We conduct numerous experimental evaluations to compare the performances of our fair contextual bandit to that of LinUCB and show that our approach achieves group-level fairness while maintaining a high utility.

Keywords Contextual bandit · Fairness · Online recommendation · Personalized recommendation

Abbreviation

MAB	Multi-arm Bandit
UCB	Upper confidence bound
LinUCB	Upper confidence bound bandit with linear payoff function
Fair-LinUCB	Fair upper confidence bound bandit with linear payoff function

1 Introduction

Personalized recommendation based on multi-arm bandit (MAB) algorithms has become a popular topic of research and shown to lead to high utility and efficiency [2] as it dynamically adapts the recommendation strategy based on feedback. However, it is also known that such personalization could incur biases or even discrimination that can influence decisions and opinions [12, 13]. Recently researchers have started taking fairness and discrimination into consideration in the design of MAB based personalized recommendation algorithms [4, 30, 44]. However, they focused on the fairness of the recommended items (e.g., services provided by small or large companies) instead of the customers who received those items. For example, [30] focused on individual fairness, i.e., "treating similar individuals similarly," and considered the individual as an arm with the aim of ensuring the probability of selecting an arm is equal to the probability with which the arm has the best quality realization. [4] aimed to achieve group fairness over items by ensuring the probability distribution from which items are sampled satisfies certain fairness constraints at all time steps. In this paper, we aim to develop novel algorithms to ensure fair and ethical treatment of customers with different profile attributes (e.g., gender, race, education, disability,

Wen Huang and Kevin Labille contributed equally to this work.

✉ Xintao Wu
xintaowu@uark.edu

Wen Huang
wenhuang@uark.edu

Kevin Labille
kclabill@uark.edu

Dongwon Lee
dongwon@psu.edu

Neil Heffernan
nth@wpi.edu

[1] University of Arkansas, Fayetteville, AR, USA

[2] Penn State University, State College, PA, USA

[3] Worcester Polytechnic Institute, Worcester, MA, USA

and economic conditions) in a contextual bandit based personalized recommendation.

Consider the personalized educational video recommendation in Table 1c as an illustrative example. Table 1a shows two students, Alice and Bob, having the same profiles except for the gender. Table 1b shows potential videos and Table 1c shows recommendations by a personalized recommendation algorithm. Focusing on the fairness of the video would ensure that videos featuring female speakers have similar chances of being recommended as those featuring male speakers. However, one group of students could benefit more from the recommended videos than the other group, therefore yielding to an unequal improvement of the learning performances. In our work, rather than focusing on the fairness of the item being recommended, i.e., the video, we focus on the user-side fairness in terms of the reward, i.e., the improvement of student's learning performance after watching the recommended video. We want to ensure that both male students and female students who share similar profiles will receive a similar reward regardless of the video being recommended, such that they both benefit from the video recommendations and improve their learning performance equally.

We study how to achieve the user-side fairness in the classic contextual bandit algorithm. The contextual bandit framework [26], which is used to sequentially recommend items to a customer based on her contextual information, is able to fit user preferences, address the cold-start problem by balancing the exploration and exploitation trade-off in recommendation systems, and simultaneously adapt the recommendation strategy based on feedback to maximize the customer's learning performance. However, such a personalized recommendation system could induce an unfair treatment of certain customers which could lead to discrimination. We develop a novel fairness aware contextual bandit algorithm such that customers will be treated fairly in personalized learning.

We train our fair contextual bandit algorithm to detect discrimination, that is, whether or not a group of customers is being privileged in terms of reward received. Our fair contextual bandit algorithm then measures to what degree each of the items (arms in bandits) contributes to the discrimination. Furthermore, in order to counter the discrimination, if any, we introduce a fairness penalty factor. The goal of this penalty factor is to maintain a balance between fairness and utility, by ensuring that the arm picking strategy will not incur discrimination whilst achieving good utility. Finally, we compare our algorithm against the traditional LinUCB both theoretically and empirically and we show that our approach not only achieves group-level fairness in terms of reward, but also yields comparable effectiveness.

Overall, our contributions are twofold. First, we develop a fairness aware contextual bandit algorithm that achieves user-side fairness in terms of reward and is robust against factors that would otherwise increase or incur discrimination. Secondly, we provide a theoretical regret analysis to show that our algorithm has a regret bound higher than LinUCB up to only an additive constant.

2 Related Work

2.1 Bandits Based Recommendation

Many bandits based algorithms have been developed to suggest recommendations for products and services. Contextual bandit [26] is an extension of the classic multi-armed bandit (MAB) algorithm [24]. The MAB chooses an action from a fixed set of choices to maximize the expected gain where each choice's properties are only partially known at the time of choice and the gain of a choice will be observed only after the action is taken. In other words, the MAB simultaneously attempts to acquire new information (exploration) and optimize decisions based on existing knowledge (exploitation). Compared to the traditional content-based recommendation approaches, the MAB is able to fit dynamic-changed user preferences over time and address the cold-start problem by balancing the exploration and exploitation trade-off in the recommendation system. However, the MAB does not use any information about the state of the environment. The contextual bandit model extends the MAB model by making the recommendation conditional on the state of the environment. Other variations include stochastic [1], Bayesian [14], adversarial [35], and non-stationary [16] bandits. In this paper, we focus on the contextual bandit model because it is posed to help identify which items work for whom. The contextual information is the customer's features and the features of the items under exploration, and the reward is derived from purchase record or customer feedback.

2.2 Fairness-Aware Machine Learning

Fairness aware machine learning is receiving increased attention. Discrimination is unfair treatment towards individuals based on the group to which they are perceived to belong. In machine learning, training data may have historically biased decisions against the protected group; models trained on such data may make discriminatory predictions against the protected group. The fair learning research community has developed extensive fair machine learning algorithms based on a variety of fairness metrics, e.g., equality of opportunity and equalized odds [17, 41], direct and indirect discrimination [6, 42, 43], counterfactual fairness [25, 33, 37], and path-specific counterfactual fairness [38].

Recently researchers have started taking fairness and discrimination into consideration in the design of MAB based

Table 1 Illustrative example

Student	Gender	Grade	GPA	..
Alice	female	9th	2.6	..
Bob	male	9th	2.6	..
..	..	..	..	..

(a) Students

Video	Gender of speaker	Rating	Length	..
2501	Female	4.3	4 min	..
0964	Male	4.3	6 min	..
..	..	..	..	..

(b) Videos

Student	Video	Reward
Alice	2501	0.60
Bob	0964	0.80
..	..	...

(c) Recommendations

personalized recommendation algorithms [3, 4, 11, 21–23, 30, 44]. Among them, [23] was the first paper of studying fairness in classic and contextual bandits. It defined fairness with respect to one-step rewards introduced a notion of meritocratic fairness, i.e., the algorithm should never place higher selection probability on a less qualified arm (e.g., job applicant) than on a more qualified arm. This was inspired by equal treatment, i.e., similar people should be treated similarly. The following works along this direction include [22] for infinite and contextual bandits, [21] for reinforcement learning, [30] for the simple stochastic bandit setting with calibration based fairness. In [28], the authors studied fairness in the setting that multiple arms can be simultaneously played and an arm could sometimes be sleeping. [15] used an unknown Mahalanobis similarity metric from some weak feedback that identifies fairness violations through an oracle rather than adopting a quantitative fairness metric over individuals. The fairness constraint requires that the difference between the probabilities that any two actions are taken is bounded by the distance between their contexts. All the above papers require some fairness constraint on arms at every round of the learning process, which is different from our user-side fairness setting. How to achieve fairness in other related contexts have also been studied, e.g., sequential decision making [18], online stochastic classification [34], offline contextual bandits [31], and collaborative filtering based recommendation systems [10, 39].

Our work is mainly motivated by the recent research works that focused on fairness from the arm perspective. Specifically, in [5], fairness is defined as a minimum rate that a task or a resource is assigned to a user in their context, which means the probability of each arm being pulled should be larger than a threshold for each time. Similarly, [32] also aimed to ensure that each arm is pulled at least a pre-specified fraction of times throughout all times. Since most of the existing fair bandit algorithms require some fairness constraint on arms at every round of the learning process, it is imperative to develop fairness-aware bandit algorithms such that the decisions made by those algorithms could achieve user-side fairness.

3 Preliminary

Throughout this paper, we use bold letters to denote a vector. We use $||\mathbf{x}||_2$ to define the L-2 norm of a vector $\mathbf{x} \in \mathbb{R}^d$. For a positive definite matrix $A \in \mathbb{R}^{d\times d}$, we define the weighted 2-norm of $\mathbf{x} \in \mathbb{R}^d$ to be $||\mathbf{x}||_A = \sqrt{\mathbf{x}^{\mathrm{T}} A\mathbf{x}}$.

3.1 LinUCB Algorithm

We use the linear contextual bandit [7] as one baseline model for our personalized recommendation. In the linear contextual bandit, the reward for each action is an unknown linear function of the contexts. Formally, we model the personalized recommendation as a contextual multi-armed bandit problem, where each user u is a "bandit player", each potential item $a \in \mathcal{A}$ is an arm and k is the number of item candidates. At time t, there is a coming user u. For each item $a \in \mathcal{A}$, its contextual feature vector $\mathbf{x}_{t,a} \in \mathbb{R}^d$ represents the concatenation of the user and the item feature vectors. The algorithm takes all contextual feature vectors as input, recommends an item $a_t \in \mathcal{A}$ and observes the reward r_{t,a_t}, and then updates its item recommendation strategy with the new observation $(\mathbf{x}_{t,a_t}, a_t, r_{t,a_t})$. During the learning process, the algorithm does not observe the reward information for unchosen items.

The total reward by round t is defined as $\sum_t r_{t,a_t}$ and the optimal expected reward as $\mathbb{E}[\sum_t r_{t,a^*}]$, where a^* indicates the best item that can achieve the maximum reward at time t. We aim to train an algorithm so that the maximum total reward can be achieved. Equivalently, the algorithm aims to minimize the regret $R(T) = \mathbb{E}[\sum_t r_{t,a^*}] - \mathbb{E}[\sum_t r_{t,a_t}]$. The contextual bandit algorithm balances exploration and exploitation to minimize regret since there is always uncertainty about the user's reward given the specific item.

Algorithm 1 LinUCB

1: Input: $\alpha \in \mathbb{R}^+$
2: **for** t = 1,2,3,...,T **do**
3: Observe contextual features of all arms $a \in \mathcal{A}_t : \mathbf{x}_{t,a} \in \mathbb{R}^d$
4: **for** $a \in \mathcal{A}_t$ **do**
5: **if** a is new **then**
6: $A_a \leftarrow \mathbf{I_d}$ (d-Dimension identity matrix)
7: $\mathbf{b}_a \leftarrow \mathbf{0}_{d\times 1}$ (d-Dimension zero vector)
8: **end if**
9: $\hat{\boldsymbol{\theta}}_a \leftarrow A_a^{-1}\mathbf{b}_a$
10: $p_{t,a} \leftarrow \hat{\boldsymbol{\theta}}_a^{\mathrm{T}}\mathbf{x}_{t,a} + \alpha\sqrt{\mathbf{x}_{t,a}^{\mathrm{T}}A_a^{-1}\mathbf{x}_{t,a}}$
11: **end for**
12: Choose arm $a_t = argmax_{a\in\mathcal{A}_t}p_{t,a}$ with ties broken arbitrarily, and observe a real-valued payoff r_{t,a_t}
13: $A_{a_t} \leftarrow A_{a_t} + \mathbf{x}_{t,a_t}\mathbf{x}_{t,a_t}^{\mathrm{T}}$
14: $\mathbf{b}_{a_t} \leftarrow \mathbf{b}_{a_t} + r_{t,a_t}\mathbf{x}_{t,a_t}$
15: **end for**

We adopt the idea of upper confidence bound (UCB) for our personalized recommendation. Algorithm 1 shows the LinUCB algorithm as introduced by [29]. It assumes the expected reward is linear in its d-dimensional features $\mathbf{x}_{t,a}$ with some unknown coefficient vector $\boldsymbol{\theta}_a^*$. Formally, for all t, we have the expected reward at time t with arm a as $\mathbb{E}[r_{t,a}|\mathbf{x}_{t,a}] = \boldsymbol{\theta}_a^{*\mathrm{T}}\mathbf{x}_{t,a}$. Here the dot product of $\boldsymbol{\theta}_a^*$ and $\mathbf{x}_{t,a}$ could also be succinctly expressed as $\langle\boldsymbol{\theta}_a^*, \mathbf{x}_{t,a}\rangle$. At each round t, we observe the realized reward $r_{t,a} = \langle\boldsymbol{\theta}_a^*, \mathbf{x}_{t,a}\rangle + \epsilon_t$ where ϵ_t is the noise term.

Basically, LinUCB applies ridge regression technique to estimate the true coefficients. Let $D_a \in \mathbb{R}^{m_a\times d}$ denote the context of the historical observations when arm a is selected and $\mathbf{r}_a \in \mathbb{R}^{m_a}$ denote the relative rewards. The regularised least-square estimator for $\boldsymbol{\theta}_a$ could be expressed as:

$$\hat{\boldsymbol{\theta}}_a = \underset{\boldsymbol{\theta}\in\mathbb{R}^d}{\arg\min}\left(\sum_{i=1}^{m_a}(r_{i,a} - \langle\boldsymbol{\theta}, D_a(i,:)\rangle)^2 + \lambda||\boldsymbol{\theta}||_2^2\right) \tag{1}$$

where λ is the penalty factor of the ridge regression. The solution to Eq. 1 is:

$$\hat{\boldsymbol{\theta}}_a = (D_a^{\mathrm{T}}D_a + \lambda I_d)^{-1}D_a^{\mathrm{T}}\mathbf{r}_a \tag{2}$$

[29] derived a confidence interval that contains the true expected reward with probability at least $1-\delta$:

$$\left|\hat{\boldsymbol{\theta}}_a^{\mathrm{T}}\mathbf{x}_{t,a} - \mathbb{E}[r_{t,a}|\mathbf{x}_{t,a}]\right| \le \alpha\sqrt{\mathbf{x}_{t,a}^{\mathrm{T}}(D_a^{\mathrm{T}}D_a + \lambda I_d)\mathbf{x}_{t,a}}$$

for any $\delta > 0$, where $\alpha = 1 + \sqrt{ln(2/\delta)/2}$. Following the rule of optimism in the face of uncertainty for linear bandits (OFUL), this confidence bound leads to a reasonable arm-selection strategy: at each round t, pick an arm by

$$a_t = argmax_{a\in\mathcal{A}_t}\left(\hat{\boldsymbol{\theta}}_a^{\mathrm{T}}\mathbf{x}_{t,a} + \alpha\sqrt{\mathbf{x}_{t,a}^{\mathrm{T}}A_a^{-1}\mathbf{x}_{t,a}}\right) \tag{3}$$

where $A_a = D_a^{\mathrm{T}}D_a + \lambda I_d$. The parameter λ could be tuned to a suitable value in order to improve the algorithm's performance. Line 13 and 14 in Algorithm 1 provide an iterative way to update the arm-related matrices A_a and b_a. In the remaining content we will denote the weighted 2-norm $\sqrt{\mathbf{x}_{t,a}^{\mathrm{T}}A_a^{-1}\mathbf{x}_{t,a}}$ as $||\mathbf{x}_{t,a}||_{A_a^{-1}}$ for the sake of simplicity.

3.2 Regret Bound of LinUCB

Existing research works (e.g., [1, 36]) on deriving the regret bound of LinUCB are based on the following four assumptions:

1. The true coefficient $\boldsymbol{\theta}^*$ is shared by all arms.
2. The error term ϵ_t follows 1-sub-Gaussian distribution for each time point.
3. T $\{\alpha_t\}_{i=1}^n$ is a non-decreasing sequence with $\alpha_1 \ge 1$.
4. T $||\mathbf{x}_{t,a}||_2 < L, ||\boldsymbol{\theta}^*||_2 < M$ for all time points and arms.

For assumption 1, since there is only one unified $\boldsymbol{\theta}$, we change the notation of D_a, $\mathbf{r}_a$ to D_t and $\mathbf{r}_t$ to denote the historical observations up to time t for all arms. The matrix A_a will be denoted as A_t accordingly. For assumption 3, following [1] and [36], we modify α in Algorithm 1 to be a time dependent sequence to get a suitable confidence set for $\boldsymbol{\theta}^*$ at each round, but use a fixed and tuned α in the experiment part to make the online computation more efficient.

To derive the regret bound, the first step is to construct a confidence set $\mathcal{C}_t \in \mathbb{R}^d$ for the true coefficient. At each round t, a natural choice is to make $\mathcal{C}_t$ centered at $\hat{\boldsymbol{\theta}}_{t-1}$. [1] shows that the confidence ellipsoid could be a suitable choice for constructing the confidence region, which is defined as follows:

$$\mathcal{C}_t = \{\boldsymbol{\theta} \in \mathbb{R}^d : ||\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_{t-1}||_{A_{t-1}} < \alpha_t\}$$

The key point is how to obtain an appropriate α_t at each round to make $\mathcal{C}_t$ contain the true parameter $\boldsymbol{\theta}^*$ with high probability and be as small as possible simultaneously. [1] takes the advantages of the martingale techniques and derives a confidence bound in terms of the weighted 2-norm shown in Lemma 1.

Lemma 1 (Theorem 2 in [1]) *Suppose the noise term is* 1*-sub-Gaussian distributed, let* $\delta \in (0, 1)$*, with probability at least* $1 - \delta$*, it holds that for all* $t \in \mathbb{N}^+$*,*

$$||\boldsymbol{\theta}^* - \hat{\boldsymbol{\theta}}_t||_{A_t} \leq \sqrt{\lambda}||\boldsymbol{\theta}^*||_2 + \sqrt{2log(|A_t|^{1/2}|\lambda I_d|^{-1/2}\delta^{-1})} \tag{4}$$

The RHS of Eq. 4 gives an appropriate selection of α_t for the confidence ellipsoid. Under the fact that $\theta^* \in \mathcal{C}_t$ and the optimistic arm selection rule of LinUCB we could further bound the regret at each round with high probability by $r_t = \langle \boldsymbol{\theta}^*, \mathbf{x}_{t,a} \rangle - \langle \hat{\boldsymbol{\theta}}, \mathbf{x}_{t,a} \rangle \leq 2\alpha_t ||\mathbf{x}_{t,a}||_{A_t^{-1}}$. Summing up the regret at each round, the following corollary gives a $\tilde{\mathcal{O}}(dlog(T))$ cumulative regret bound up to time T.

Corollary 1 (Corollary 19.3 in [27]) *Under the assumptions above, the expected regret of LinUCB with* $\delta = 1/T$ *is bounded by*

$$R_T \leq Cd\sqrt{T}log(TL) \tag{5}$$

where C is a suitably large constant.

4 Methods

We focus on how to achieve user-side fairness in contextual bandit based recommendation and present our fair contextual bandit algorithm, called Fair-LinUCB and derive its regret bound.

4.1 Problem Formulation

We define a sensitive attribute $S \in \mathbf{x}_{t,a}$ with domain values $\{s^+, s^-\}$ where s^+ (s^-) is the value of the privileged (protected) group. Let T_s denote a time index subset such that the users being treated at time points in T_s all hold the same sensitive attribute value s. We introduce the group-level cumulative mean reward as $\bar{r}^s = \frac{1}{|T_s|}\sum_{t \in T_s} r_{t,a}$. Specifically, $\bar{r}^{s^+}$ denotes the cumulative mean reward of the individuals with sensitive attribute $S = s^+$, and $\bar{r}^{s^-}$ denotes the cumulative mean reward of all individuals having the sensitive attribute $S = s^-$.

We define the group fairness in contextual bandits as $\mathbb{E}[\bar{r}^{s^+}] = \mathbb{E}[\bar{r}^{s^-}]$, more specifically, the expected mean reward of the protected group and that of the unprotected group should be equal. A recommendation algorithm incurs group-level unfairness in regards to a sensitive attribute S if $|\mathbb{E}[\bar{r}^{s^+}] - \mathbb{E}[\bar{r}^{s^-}]| > \tau$ where $\tau \in \mathbb{R}^+$ reflects the tolerance degree of unfairness.

4.2 Fair-LinUCB algorithm

We describe our fair LinUCB algorithm and show its pseudo code in Algorithm 2. The key difference from the traditional LinUCB is the strategy of choosing an arm during recommendation (shown in Line 12 of Algorithm 2). In the remainder of this section, we explain how this new strategy achieves user-side group-level fairness.

Algorithm 2 Fair-LinUCB

Input: α , $\gamma \in \mathbb{R}^+$
$\bar{r}^{s^+}, \bar{r}^{s^-} \leftarrow 0$
for t = 1,2,3,..., T **do**
  Observe features of all arms $a \in \mathcal{A}_t : \mathbf{x}_{t,a} \in \mathbb{R}^d$
  for $a \in \mathcal{A}_t$ **do**
    if a is new **then**
      $A_a \leftarrow \lambda \mathbf{I}_d$ (d-Dimension identity matrix)
      $\mathbf{b}_a \leftarrow \mathbf{0}_{d\times 1}$ (d-Dimension zero vector)
      $\bar{r}_a^{s^+}, \bar{r}_a^{s^-} \leftarrow 0$
    end if
    $\hat{\boldsymbol{\theta}}_a \leftarrow A_a^{-1}\mathbf{b}_a$
    $p_{t,a} \leftarrow \hat{\boldsymbol{\theta}}_a^{\mathrm{T}}\mathbf{x}_{t,a} + \alpha||\mathbf{x}_{t,a}||_{A_a^{-1}} + \mathcal{L}(\gamma, F_a)$
  end for
  Choose arm $a_t = argmax_{a\in\mathcal{A}_t} p_{t,a}$ with ties broken arbitrarily, and observe a real-valued payoff r_{t,a_t}
  $A_a \leftarrow A_a + \mathbf{x}_{t,a_t}\mathbf{x}_{t,a_t}^{\mathrm{T}}$
  $\mathbf{b}_a \leftarrow \mathbf{b}_a + r_{t,a_t}\mathbf{x}_{t,a_t}$
  if $S_t = s^+$ **then**
    update $\bar{r}^{s^+}, \bar{r}_a^{s^+}$ with r_{t,a_t}
  else
    update $\bar{r}^{s^-}, \bar{r}_a^{s^-}$ with r_{t,a_t}
  end if
end for

Given a sensitive attribute S with domain values $\{s^+, s^-\}$, the goal of our fair contextual bandit is to minimize the cumulative mean reward difference between the protected group and the privileged group while preserving its efficiency. Note that Fair-LinUCB can be extended to the general setting of multiple sensitive attributes $S_j \in \boldsymbol{S} = \{S_1, S_2, ..., S_l\}$ where $\boldsymbol{S} \subset \mathbf{x}_{t,a}$ and each S_j can have multiple domain values. In order to measure the unfairness at the group-level, our Fair-LinUCB algorithm will keep track of both cumulative mean rewards along the time, e.g., $\bar{r}^{s^+}$ and $\bar{r}^{s^-}$. We capture the orientation of the bias (i.e., towards which group the bias is leaning) through the sign of the cumulative mean reward difference. By doing so, Fair-LinUCB is able to know which group is being discriminated and which group is being privileged.

When running context bandits for recommendation, each arm may generate a reward discrepancy and therefore contribute to the unfairness to some degree. Fair-LinUCB captures the reward discrepancy at the arm level by keeping track of the cumulative mean reward generated by each arm a for both groups s^+ and s^-. Specifically, let $\bar{r}_a^{s^+}$ denote the average of the rewards generated by arm a for the group s^+, and let $\bar{r}_a^{s^-}$ denote the average of the rewards generated by arm a for the group s^-. The bias of an arm is thus the difference of both averages: $\Delta_a = (\bar{r}_a^{s^+} - \bar{r}_a^{s^-})$. Finally, by combining the direction of the bias and the amount of the bias induced by each arm a, we define the fairness penalty term as $F_a = -sign(\bar{r}^{s^+} - \bar{r}^{s^-}) \cdot \Delta_a$, and exert onto the UCB value in our fair contextual bandit algorithm. Note that the less an arm contributes to the bias, the smaller the penalty.

As a result, if an arm has a high UCB but incurs bias, its adjusted UCB value will decrease and it will be less likely to be picked by the algorithm. In contrast, if an arm has a small UCB but is fair, its adjusted UCB value will increase, and it will be more likely to be picked by the algorithm, thereby reducing the potential unfairness in recommendation. Different from the traditional LinUCB that picks the arm to solely maximize the UCB, our Fair-LinUCB accounts for the fairness of the arm and picks the arm that maximizes the summation of the UCB and the fairness. Formally, we show the modified arm selection criteria in Eq. 6.

$$p_{t,a} \leftarrow \hat{\boldsymbol{\theta}}_a^{\mathrm{T}}\mathbf{x}_{t,a} + \alpha||\mathbf{x}_{t,a}||_{A_a^{-1}} + \mathcal{L}(\gamma, F_a) \tag{6}$$

We adopt a linear mapping function $\mathcal{L}$ with input parameters γ and F_a to transform the fairness penalty term proportionally to the size of its confidence interval. Specifically,

$$\mathcal{L}(\gamma, F_a) = \frac{\alpha_t||\mathbf{x}_{t,a_m}||_{A_t^{-1}}}{2}(F_a + 1)\gamma \tag{7}$$

$$a_m = argmin_{a\in\mathcal{A}_t}||\mathbf{x}_{t,a}||_{A_a^{-1}} \tag{8}$$

Assuming that the reward generated is in the range [0, 1], the fairness penalty F_a lies in $[-1, 1]$. When designing the coefficient of the linear mapping function, we choose a_m to be the arm with the smallest confidence interval to guarantee a unified fairness calibration among all the arms. Under the effect of $\mathcal{L}$, the range of the fairness penalty is mapped from $[-1, 1]$ to $[0,\ \gamma\alpha_t||\mathbf{x}_{t,a_m}||_{A_t^{-1}}]$, which implies a similar scale with the confidence interval. In our empirical evaluations, we show how γ controls fairness-accuracy trade-off on the practical performance of Fair-LinUCB.

Our purposed Fair-LinUCB algorithm studies a contextual linear bandit problem and follows the rule of optimism in the face of uncertainty for linear bandits (OFUL) to conduct arm selections. For an arm set $\mathcal{A}_t$ with k arms at each time step, Fair-LinUCB has a $\Theta(k)$ per-step time complexity. There are some state-of-the-art research works that try to further reduce the computational complexity of linear bandits [40], but it is not the main focus of this paper.

4.3 Handling a Single Sensitive Attribute with Multiple Domain Values

It is possible to extend our algorithm to handle a sensitive attribute with multiple domain values. For example, the sensitive attribute of race has multiple domain values such as black, white, asian. Consider a sensitive attribute S with multiple domain values belonging to either privileged group $S^+ = \{s_i^+\}$ or protected group $S^- = \{s_j^-\}$ with finite cardinalities. Similarly to the binary case, we can keep track of the cumulative mean reward along the time for all domain values, e.g., $\bar{r}_a^{s_i^+}, \bar{r}_a^{s_j^-}$ .... We can then define the bias of an arm by taking the difference of the averaged cumulative mean reward of all domain value for each group as follows:

$$\Delta_a = \left(\frac{\sum_i \sum_{t=1}^{T} \bar{r}_a^{s_i^+} \cdot \mathbb{1}_{s_t = s_i^+}}{\sum_{t=1}^{T} \mathbb{1}_{s_t \in S^+}} - \frac{\sum_j \sum_{t=1}^{T} \bar{r}_a^{s_j^-} \cdot \mathbb{1}_{s_t = s_j^-}}{\sum_{t=1}^{T} \mathbb{1}_{s_t \in S^-}} \right)$$

We can further define F_a accordingly as follows:

$$F_a = -sign(\bar{r}^{S^+} - \bar{r}^{S^-}) \cdot \Delta_a$$

Such changes will handle multiple domain values for the sensitive attribute, including the usual case where the protected group has a single value and the privileged group has multiple domain values, as well as the case where the protected group also has multiple domain values. The remaining of the algorithm needs no change.

4.4 Handling Multiple Sensitive Attributes

Our algorithm can be further extended to multiple sensitive attributes. For example, one could consider both the gender and the race to be sensitive attributes. Suppose we have k sensitive attributes, consider the set $\mathbf{S}$ which contains all possible cross products of the domain values of all k sensitive attributes. We then have both subsets $\mathbf{S}^+ \subseteq \mathbf{S}$ and $\mathbf{S}^- \subseteq \mathbf{S}$ ($\mathbf{S}^+ \cap \mathbf{S}^- = \emptyset$) representing the privileged group and protected group respectively. Each user therefore belongs to one single group. For example, if we have both the gender with domain values {male, female} and the race with domain values {black, white, asian} as sensitive attributes, our set S will have the following values: {black male, black female, white male, white female, asian male, asian female}. In this case, the calculation method for the cumulative mean reward $\bar{r}_a^{s_i^+}, \bar{r}_a^{s_j^-}$ ... does not change, and both Δ_a and F_a can be computed as in the previous scenario.

4.5 Regret Analysis

In this section, We prove that our Fair-LinUCB algorithm has a $\tilde{\mathcal{O}}(dlog(T))$ regret bound under certain assumptions with carefully chosen parameters. We adopt the regret analysis framework of linear contextual bandit and introduce a mapping function on the fairness penalty term. By applying the mapping function $\mathcal{L}$ we make our fairness penalty term possess the similar scale with the half length of the confidence interval. Thus we can merge the regret generated by UCB term and fairness term together and derive our regret bound.

Theorem 1 *Under the same assumptions shown in Sect. 3.2, further assuming γ is a moderate small constant with $\gamma \leq \Gamma$, there exists $\delta \in (0, 1)$ such that with probability at least $1 - \delta$ Fair-LinUCB achieves the following regret bound*:

$$R_T \leq \sqrt{2Tdlog(1 + TL^2/(d\lambda))} \times (2 + \Gamma)(\sqrt{\lambda}M + \sqrt{2log(1/\delta) + dlog(1 + TL^2/(d\lambda))}) \tag{9}$$

Proof We first introduce three technical lemmas from [1] and [27] to help us complete the proof of Theorem 9.

Lemma 2 (Lemma 11 in appendix of [1]) *If $\lambda \geq max(1, L^2)$, the weighted L2-norm of feature vector could be bounded by*: $\sum_{t=1}^{T} ||\mathbf{x}_{t,a}||^2_{A_t^{-1}} \leq 2log\frac{|A_t|}{\lambda^d}$

Lemma 3 (Lemma 10 in appendix of [1]) *The determinant $|A_t|$ could be bounded by*: $|A_t| \leq (\lambda + tL^2/d)^d$.

Lemma 4 (Theorem 20.5 in [27]) *With probability at least $1 - \delta$, for all the time point $t \in \mathbb{N}^+$ the true coefficient $\boldsymbol{\theta}^*$ lies in the set*:

$$\mathcal{C}_t = \{\theta \in \mathbb{R}^d : ||\hat{\theta}_t - \theta||_{A_t} \leq \sqrt{\lambda}M + \sqrt{2log(1/\delta) + dlog(1 + TL^2/(d\lambda))}\} \quad (10)$$

In Fair-LinUCB, the range of fairness term is $[-1, 1]$, we apply a linear mapping function $\mathcal{L}(\gamma, x) = \frac{\alpha_t ||\mathbf{x}_{t,a_m}||_{A_t^{-1}}}{2}(x+1)\gamma$ to map the range of $\mathcal{L}(\gamma, F_a)$ to $[0,\ \gamma\alpha_t||\mathbf{x}_{t,a_m}||_{A_t^{-1}}]$, where $a_m = argmin_{a\in\mathcal{A}_t}||\mathbf{x}_{t,a}||_{A_a^{-1}}$.

According to the rule, the regret at each time t is bounded by:

$$\begin{aligned}
reg_t &= \mathbf{x}_{t,a}^{\mathrm{T}}\hat{\theta}_t - \mathbf{x}_{t,a}^{\mathrm{T}}\theta^* \\
&\leq \mathbf{x}_{t,a}^{\mathrm{T}}\hat{\theta}_t + \alpha_t||\mathbf{x}_{t,a}||_{A_t^{-1}} + \mathcal{L}(\gamma, F_a) - \mathbf{x}_{t,a}^{\mathrm{T}}\theta^* \\
&\leq \mathbf{x}_{t,a}^{\mathrm{T}}\hat{\theta}_t + \alpha_t||\mathbf{x}_{t,a}||_{A_t^{-1}} + \mathcal{L}(\gamma, F_a) - (\mathbf{x}_{t,a}^{\mathrm{T}}\hat{\theta}_t - \alpha_t||\mathbf{x}_{t,a}||_{A_t^{-1}}) \\
&\leq 2\alpha_t||\mathbf{x}_{t,a}||_{A_t^{-1}} + \mathcal{L}(\gamma, 1) \\
&= 2\alpha_t||\mathbf{x}_{t,a}||_{A_t^{-1}} + \gamma\alpha_t||\mathbf{x}_{t,a_m}||_{A_t^{-1}} \\
&\leq 2\alpha_t||\mathbf{x}_{t,a}||_{A_t^{-1}} + \gamma\alpha_t||\mathbf{x}_{t,a}||_{A_t^{-1}} \\
&\leq (2+\Gamma)\alpha_t||\mathbf{x}_{t,a}||_{A_t^{-1}}
\end{aligned}$$

The second line above is derived based on the theoretic result in Lemma 1 and following the selection rule of the Fair-LinUCB algorithm, specifically, $\mathbf{x}_{t,a^*}^{\mathrm{T}}\theta^* \leq \mathbf{x}_{t,a^*}^{\mathrm{T}}\hat{\theta}_t + \alpha_t||\mathbf{x}_{t,a^*}||_{A_t^{-1}} \leq \mathbf{x}_{t,a^*}^{\mathrm{T}}\hat{\theta}_t + \alpha_t||\mathbf{x}_{t,a^*}||_{A_t^{-1}}$. $+ \mathcal{L}(\gamma, F_{a^*}) \leq \mathbf{x}_{t,a}^{\mathrm{T}}\hat{\theta}_t + \alpha_t||\mathbf{x}_{t,a}||_{A_t^{-1}} + \mathcal{L}(\gamma, F_a)$
Note that Lemma 1 can be equally applied here because the estimator $\hat{\theta}_t$ is still a valid ridge regression estimator at each round.

Summing up the regret at each bound, with probability at least $1-\delta$ the cumulative regret up to time T is bounded by:

$$R_T = \sum_{t=1}^{T} reg_t \leq \sqrt{T\sum_{t=1}^{T} reg_t^2} \leq (2+\Gamma)\alpha_T\sqrt{T\sum_{t=1}^{T}||\mathbf{x}_{t,a}||_{A_t^{-1}}^2} \quad (11)$$

Since $\{\alpha_t\}_{i=1}^n$ is a non-decreasing sequence, we can enlarge each element α_t to α_T to obtain the inequalities in Eq. 11. By applying the inequalities from Lemmas 2 and 3 we could further relax the regret bound up to time T to:

$$\begin{aligned}
R_T &\leq (2+\Gamma)\alpha_T\sqrt{2Tlog\frac{|A_t|}{\lambda^d}} \\
&\leq (2+\Gamma)\alpha_T\sqrt{2Td(log(\lambda + TL^2/d) - log\lambda)} \\
&= (2+\Gamma)\alpha_T\sqrt{2Tdlog(1 + TL^2/(d\lambda))}
\end{aligned} \quad (12)$$

Following the result of Lemma 1, by loosing the determinant of A_t according to Lemmas 3, Lemma 4 provides a suitable choice for α_T up to time T. By plugging in the RHS from Eq. 10 we get the regret bound shown in Theorem 1:

$$R_T \leq \sqrt{2Tdlog(1 + TL^2/(d\lambda))} \times (2+\Gamma)(\sqrt{\lambda}M + \sqrt{2log(1/\delta) + dlog(1 + TL^2/(d\lambda))})$$

□

Corollary 2 *Setting $\delta = 1/T$, the regret bound in Theorem 1 could be simplified as $R_T \leq C'd\sqrt{T}log(TL)$.*

Comparing Corollary 2 with Corollary 1 (for LinUCB), we can see the regret bound of Fair-LinUCB is worse than the original LinUCB only up to an additive constant. This perfectly matches the intuition that Fair-LinUCB is able to keep aware of the fairness and guarantee there is no reward gap between different subgroups or individuals, however, it suffers from a relatively higher regret.

5 Results and Discussion

5.1 Experiment Setup

5.1.1 Simulated Dataset

There are presently no publicly available datasets that fits our environment. We therefore generate one simulated dataset for our experiments by combining the following two publicly available datasets.

- **Adult dataset**: The Adult dataset [9] is used to represent the students (or bandit players). It is composed of 31,561 instances: 21,790 males and 10,771 females, each having 8 categorical variables (work class, education, marital status, occupation, relationship, race, sex, native-country) and 3 continuous variables (age, education number, hours per week), yielding an overall of 107 features after one-hot encoding.
- **YouTube dataset**: The Statistics and Social Network of YouTube Videos[1] dataset is used to represent the items to be recommended (or arms). It is composed of 1580 instances each having 6 categorical features (age of video, length of video, number of views, rate, ratings, number of comments), yielding a total of 25 features after one-hot encoding. We add a 26^{th} feature used to represent the gender of the speaker in the video which is drawn from a Bernoulli distribution with the probability of success as 0.5.

The feature contexts $\mathbf{x}_{t,a}$ used throughout the experiment is the concatenation of both the student feature vector and the video feature vector. In our experiments we choose the

[1] https://netsg.cs.sfu.ca/youtubedata/.

sensitive attribute to be the **gender of adults**, and we therefore focus on the unfairness on the group-level for the male group and female group. Furthermore, based on the findings of [19] and [8] that same-gender teachers positively increase the learning outcome of students, we assume that a male student prefers a video featuring a male speaker and a female student prefers a video featuring a female speaker. Thus, in order to maintain the linear assumption of the reward function, we add an extra binary variable in the feature context vector that represents whether or not the gender of the student matches the gender of the speaker in the video. Overall, $\mathbf{x}_{t,a}$ contains a total of 134 features.

For our experiments, we use a subset of 5000 random instances from the Adult dataset, which is then split into two subsets: one for training and one for testing. The training subset is composed of 1500 male individuals and 1500 female individuals whilst the testing subset is composed of 1000 males and 1000 females. Similarly, a subset of YouTube dataset is used as our pool of videos to recommend (or arms). The subset contains 30 videos featuring a male speaker and 70 videos featuring a female speaker.

5.1.2 Reward Function

We compare our Fair-LinUCB against the original LinUCB using a simple reward function wherein we manually set the θ^* coefficients. The reward r is defined as

$$r = \theta_1^* \cdot x_1 + \theta_2^* \cdot x_2 + \theta_3^* \cdot x_3$$

where $\theta_1^* = 0.3$, $\theta_2^* = 0.4$, $\theta_3^* = 0.3$ and $x_1 =$ video rating, $x_2 =$ education level, $x_3 =$ gender match. The remaining $d - 3$ coefficients are set to 0. Hence, only these three features matter to generate our true reward. The gender match is set to 1 if both the student gender and the gender of the video match, and 0 otherwise. The education level is divided into 5 subgroups each represented by a value ranging from 0.0 to 1.0 with a higher education level yielding a higher value. In our setup, the education level is used to represent the strength of the student. Similarly, the video rating varies from 0 to 1.0, and is used to represent the educational quality of the video. Evidently, a higher reward is generated when the gender of the student matches the gender of the video.

5.1.3 Evaluation Metrics

Throughout our experiments we measure the effectiveness of the algorithms through the average utility loss. Since we know the true reward function, we can derive the optimal reward at each round t. We can thus define

$$\text{utility loss} = \frac{1}{T}\sum_{t=1}^{T}(r_{t,a^*} - r_{t,a})$$

where r_{t,a^*} is the optimal reward at round t by choosing arm a^* and $r_{t,a}$ is the observed reward by the algorithm after picking arm a.

We measure the fairness of the algorithms through the absolute value of the difference between the cumulative mean reward ($\bar{r}_t$, as introduced in Sect. 4.1) of the male group and female group:

$$\text{reward difference} = |\bar{r}_t^{s^+} - \bar{r}_t^{s^-}|$$

Additionally, for all following figures the left hand side plots the cumulative mean reward during the training phase whilst the right hand side reflects the cumulative mean reward over the testing dataset. Due to space limit, all tables report measures on the testing data solely. Note that the contextual bandit continues to learn throughout both phases.

5.1.4 Baselines

As existing fair bandits algorithms focus on item-side fairness, we mainly compare our Fair-LinUCB against LinUCB in terms of utility-fairness trade-off in our evaluations. We also report a comparison with a simple fair LinUCB method that suppresses the unfairness by removing the sensitive attribute and all its correlated attributes from the context. We name this method as Naive in our evaluation.

5.2 Comparison with Baselines

5.2.1 Comparison with LinUCB

Our first experiment compares the performances of the traditional LinUCB against our Fair-LinUCB, using the reward function r described in the previous section. Figure 2 plots the cumulative mean reward of both the male and female groups over time. We can notice that the cumulative mean rewards of both groups suffer a discrepancy with LinUCB, and the outcome can therefore be considered unfair towards the male group. Indeed, as shown on Fig. 2a the cumulative mean reward of the female group (0.839) is greater than the cumulative mean reward of the male group (0.802), yielding a reward difference of 0.037. The utility loss incurred is 0.050. In contrast, Fair-LinUCB is able to seal the reward discrepancy with a γ coefficient set to 3 (Fig. 2b). Our algorithm thereby achieves a cumulative mean reward of 0.819 for both the male group and the female group, which yields a reward difference of 0.0, while incurring a utility loss of 0.052. Our Fair-LinUCB outperforms the traditional LinUCB in terms of reward difference while suffering a slight loss of utility. The comparison results are summarized in the first two rows of Table 2.

To evaluate how the inclusion or exclusion of sensitive attributes affects the fairness-utility tradeoff, we compare

Table 2 Comparison of three algorithms under reward function r

	Utility loss	Reward difference
Fair-LinUCB ($\gamma = 3$)	0.052	**0.000**
LinUCB	0.050	0.037
Naive	**0.046**	0.035

LinUCB against Fair-LinUCB with a modified reward function:

$$r_2 = \theta_1^* \cdot x_1 + \theta_2^* \cdot x_2$$

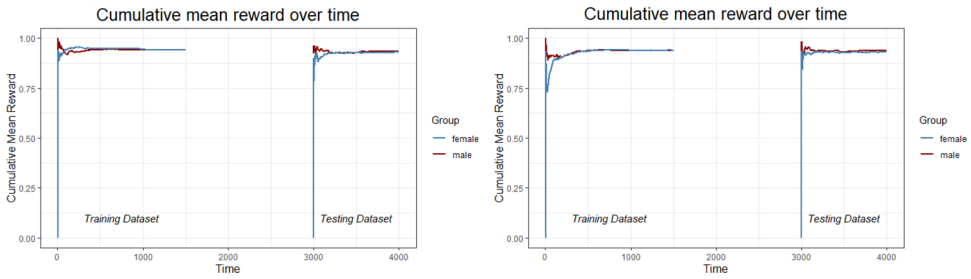
where $\theta_1^* = 0.5$ and $\theta_2^* = 0.5$ and $x_1 =$ video rating, $x_2 =$ education level The remaining $d - 2$ coefficients are set to 0. r_2 is not dependent upon the gender match attribute and expects to incur zero or small discrepancy between both groups. As depicted on Fig. 1, both LinUCB and Fair-LinUCB show a very low cumulative mean reward discrepancy. Specifically, LinUCB incurs a utility loss of 0.037 and a reward difference of 0.006, while Fair-LinUCB incurs 0.034 utility loss and a reward difference of 0.008. Furthermore, in this case, although Fair-LinUCB has additional constraints for the arm picking strategy due to the fairness penalty, it does not induce any loss of utility when compared to LinUCB.

5.2.2 Comparison with Naive

Naive method tries to achieve fairness by removing from the context the sensitive attribute and the features that are highly correlated with the sensitive attribute. In our experiment, we first compute the correlation matrix of all the user's features and then remove the gender feature as well as all features that are highly correlated with it. Specifically, features that have a correlation coefficient greater than 0.3 were removed, which include the following: is male, is female, is divorced, is married, is widowed, is a husband, has an administrative clerical job, has a salary less than 50 k. We report in the last row of Table 2 the utility loss and reward difference of Naive with reward function r.

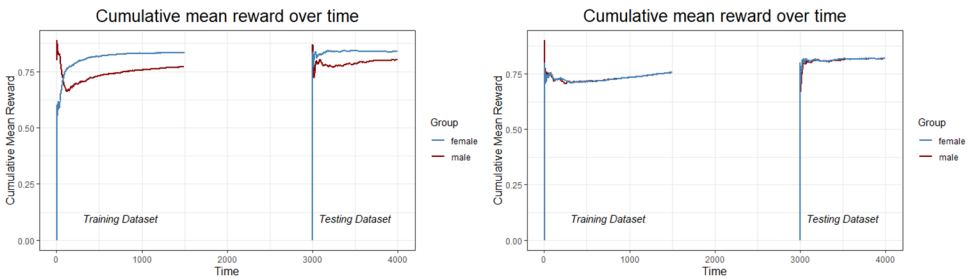
We can see the reward discrepancy between the male and female groups from the Naive method is 0.035, thus showing it cannot completely remove discrimination. The utility loss from the Naive method is 0.046, which is only slightly smaller than LinUCB and Fair-LinUCB. In fact, as shown in Table 3, Fair-LinUCB with $\gamma = 2$ can outperform the Naive method in terms of both fairness and utility. In short, removing the gender information and highly correlated features from the context does not necessarily close the gap of the reward difference.

In summary, although LinUCB learns to pick the arm that maximizes the reward given a particular context, we have seen that it could incur discrimination towards a group of users in some cases. Fair-LinUCB is capable of detecting when unfairness occurs, and will adapt its arm picking strategy accordingly so as to be as fair as possible and reduce any reward discrepancy. When a reward discrepancy is not detected, our algorithm does not need to adjust the arm picking strategy and therefore performs as well as the traditional LinUCB.



(a) LinUCB (b) Fair-LinUCB $\gamma = 3$

Fig. 1 LinUCB vs Fair-LinUCB with reward function r_2



(a) LinUCB (b) Fair-LinUCB $\gamma = 3$

Fig. 2 LinUCB vs Fair-LinUCB with reward function r

Table 3 Impact of γ on the Fairness-Utility Trade-off

	Utility Loss	Reward difference
$\gamma = 0$	0.050	0.037
$\gamma = 1$	0.040	0.016
$\gamma = 2$	**0.035**	0.004
$\gamma = 3$	0.052	**0.000**
$\gamma = 4$	0.081	**0.000**

5.3 Impact of γ on Fairness-Utility Trade-off

The γ coefficient introduced in Sect. 4.2 controls the weight of the fairness penalty that the algorithm will exert onto the UCB value. Indeed, as shown in Equation (7), γ is used to adjust the upper bound of the linear mapping function $\mathcal{L}(\gamma, F_a)$. Thus, when the γ coefficient increases, the range of the fairness penalty increases proportionally which will consequently increase the UCB value in Eq. 6. The γ coefficient therefore reflects the significance of the fairness of Fair-LinUCB. However, as γ becomes larger, the fairness penalty becomes out of proportion to the extent of neglecting the importance of the UCB value, thereby decreasing the utility of the algorithm.

To evaluate the fairness-utility trade-off of Fair-LinUCB, we compare several γ values and report the fairness and utility loss in Table 3. With a γ equal to 0, our algorithm behaves as a traditional LinUCB, therefore it incurs discrimination (reward difference measured at 0.037), and a utility loss of 0.050 is reported. We can observe that when γ increases slightly, the algorithm improves the reward difference and loss of utility. Specifically, a reward difference of 0.016 is achieved for $\gamma = 1$ with a utility loss of 0.040, and a reward difference of 0.004 with a utility loss of 0.035 is achieved with $\gamma = 2$. Although the utility losses are improved, they both remain not fair. In our best case scenario, with $\gamma = 3$, the algorithm is completely fair, i.e., reward difference is 0.000, with a utility loss of 0.052. Finally, when the γ coefficient is too large, the algorithm prioritizes fairness over utility, resulting in a fair algorithm that suffers a greater loss of utility. For example, with a γ set to 4, Fair-LinUCB incurs a utility loss of 0.081.

5.4 Impact of Arm and User Distributions

In certain cases the distribution of the arms (videos) or the users can significantly impact the cumulative mean reward of some groups of users, and therefore incur the large reward difference. In our experiment, given the reward function r, we first explore the impact of the ratio of gender arms, i.e., videos by female or male speakers, and then we investigate the impact of the order of the data in which the algorithm learns. The following results discuss our findings.

Table 4 Impact of different arm ratio on the fairness and utility

Arm ratio m:f	Utility Loss	Reward difference	Male cmr	Female cmr
LinUCB				
7:3	0.061	0.029	0.824	0.795
1:1	0.053	0.012	0.824	0.812
3:7	**0.050**	0.037	0.802	0.839
Fair-LinUCB $\gamma = 3$				
7:3	0.087	0.001	0.784	0.783
1:1	0.162	**0.000**	0.709	0.709
3:7	0.052	**0.000**	0.819	0.819

5.4.1 Gender Arm Ratio

We explore the effect of three different arm ratio values: (1) 70% male and 30% female, (2) 50% male and 50% female, and (3) 30% male and 70% female. Table 4 reports the utility loss, reward difference, as well as both the cumulative mean reward for the male and female groups. As observed with the LinUCB performances, the arm ratio induces unfairness on some user group. Indeed, when there is a majority of male arms, it appears that the male user group will benefit more and will have a higher cumulative mean reward. Likewise, when the arms have more females than males, the female user group will benefit more than the male user group, and will therefore have a higher cumulative mean reward. Although having a balanced ratio of male and female arms minimizes the reward difference, it is not always feasible or convenient to adjust the arms distribution in practice.

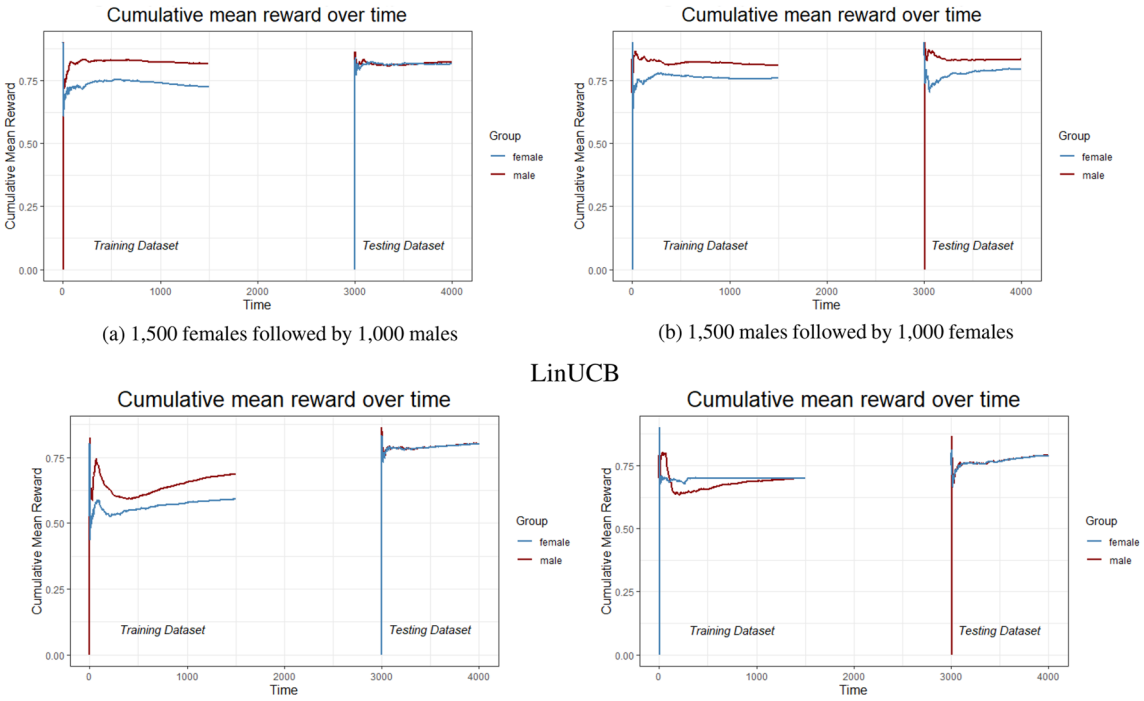
We ran the same experiment with Fair-LinUCB with $\gamma = 3$. As we can see, in all three cases, Fair-LinUCB yields a very low reward difference. Indeed, our Fair-LinUCB learns which group is being discriminated and adjusts its arm picking strategy accordingly so as to remove any discrimination, it however suffers a higher utility loss than LinUCB. Note that a γ different than 3 could yield a better utility loss for the ratios 7:3 and 1:1.

Thus, as opposed to a traditional LinUCB which only learns to maximize the reward given a context, our Fair-LinUCB learns how to achieve fairness at the same time, making it robust against factors that would otherwise induce unfairness.

5.4.2 Order of the Training Data

It is our intuition that the order of the data in which LinUCB learns to recommend an item could affect its recommendation choice or arm pick.

In these experiments, we use the 70% male and 30% female arms setting, and we manually change the order of the training data. In the first setting, we manually set the



(a) 1,500 females followed by 1,000 males (b) 1,500 males followed by 1,000 females

(c) 1,500 females followed by 1,000 males (d) 1,500 males followed by 1,000 females

Fig. 3 Impact of the order of the data on the performances

order of the students in the training data by having all 1500 female students followed by the 1500 males instances. In the second setting we order the data by having all 1500 male instances first, followed by the 1500 female instances. The test data remains shuffled. We then compare LinUCB with Fair-LinUCB in order to see the impact on the learning strategy of both algorithms.

We ran the traditional LinUCB and report the cumulative mean reward of the male user group and female user group over time. As shown in Fig. 3a, b, overall the male group gets a higher cumulative mean reward than the female group. Particularly, the male group achieves 0.822 against 0.816 for the female group in Fig. 3a and 0.834 against 0.795 in Fig. 3b. However, we notice that the reward discrepancy is much higher in the second scenario as compared to the first one. From Fig. 3a, it appears that learning to recommend videos to all females students prior to recommending videos to any male students affects the recommendation process positively (i.e., it yields a higher cumulative mean reward for the female group). Thus, the order of the training data can sometimes affect the recommendation process of LinUCB, which can impact the recommendation outcomes and may also induce discrimination towards one group.

We ran the same experiments with Fair-LinUCB, using a γ coefficient of 3, and we report our results in Fig. 3c, d. We notice that in both situations our Fair-LinUCB remains very fair, that is, we do not observe a cumulative mean reward discrepancy between the male and female user group. In the former setting, both groups achieve a cumulative mean reward of 0.802 against 0.789 in the latter, both yielding a cumulative mean reward difference of 0.00. In addition, we notice that regardless of the order of the training data our Fair-LinUCB performs equivalently in both scenarios. However, the gain in fairness also induces a loss of utility. Indeed, in the first setting LinUCB achieves 0.052 utility loss against 0.070 for Fair-LinUCB. In the second setting, LinUCB achieves 0.057 against 0.082 for Fair-LinUCB. Thus, our results indicate that Fair-LinUCB is able to close the reward discrepancy and is robust against scenarios that might otherwise induce unfairness.

6 Conclusion

Previous research have shown that personalized recommendation can be highly effective at a cost of introducing unfairness. In this paper, we have proposed a fair contextual bandit

algorithm for personalized recommendation. While current research in fair recommendation mainly focus on how to achieve fairness on the items that are being recommended, our work differs by focusing on fairness on the individuals whom are being recommended an item. Specifically, we aim to recommend items to users while ensuring that both the protected group and privileged group improve their learning performance equally. Our developed Fair-LinUCB improves upon the state-of-the-art LinUCB algorithm by automatically detecting unfairness, and adjusting its arm-picking strategy such that it maximizes the fairness outcome. We further provided a regret analysis of our fair contextual bandit algorithm and demonstrate that the regret bound is only worse than LinUCB up to an additive constant. Finally, we evaluate the performances of our Fair-LinUCB against that of LinUCB by comparing both their effectiveness and degree of fairness. Experimental evaluations showed that our Fair-LinUCB achieves competitive effectiveness while outperforming LinUCB in terms of fairness. We further showed that our algorithm is robust against numerous factors that would otherwise induce or increase discrimination in the traditional LinUCB algorithm. In this work we made a linear assumption on the reward function. In the future work, we plan to extend the user-level fairness to more general cases and make it easier to be implemented in multifarious reward functions. We plan to develop heuristics to determine the appropriate value for the fairness-accuracy trade off parameter γ. We also plan to study user-side fairness in the multiple choice linear bandits, e.g., recommending multiple videos to a student at each round. Finally, we plan to study how to achieve individual fairness in bandits algorithms.

Acknowledgements This work was supported in part by NSF 1937010, 1940093, 1940076, and 1940236. This paper is a significant extension of the 6-page conference paper [20] published in IEEE BigData'21 conference paper. This extended version contains complete proofs of all theoretical results and experimental evaluations in addition to expanded related work, preliminaries, introduction, and conclusions.

Author Contributions Wen Huang and Kevin Labille contributed this work in writing, methodology, data preprocessing, and software. Xintao Wu contributed in conceptualization, writing, reviewing, and supervision. Dongwon Lee and Neil Heffernan contributed in editing, reviewing, and validation.

Funding This work was supported in part by NSF 1937010, 1940093, 1940076, and 1940236.

Data Availability The source code and datasets are available at https://www.dropbox.com/s/44bwtnxs0j8wbw4/Achieving_User-Side_Fairness_in_Contextual_Bandits.zip?dl=0. No materials are present.

Declarations

Ethical Approval and Consent to participate Not applicable.

Consent for publication The authors declare consent for publication.

Conflict of interest The authors declare they have no conflicts of interest.

References

1. Abbasi-Yadkori Y, Pál D, Szepesvári C. Improved algorithms for linear stochastic bandits. In Advances in Neural Information Processing Systems, 2011;2312–2320.
2. Bouneffouf D, Rish I, Aggarwal CC. Survey on applications of multi-armed and contextual bandits. In IEEE Congress on Evolutionary Computation, CEC, Glasgow, United Kingdom, July 19–24, 2020. IEEE. 2020;2020:1–8.
3. Burke R, Sonboli N, Ordonez-Gauger A. Balanced neighborhoods for multi-sided fairness in recommendation. In Conference on Fairness, Accountability and Transparency, 2018;202–214.
4. Celis LE, Kapoor S, Salehi F, Vishnoi NK. An algorithmic framework to control bias in bandit-based personalization. arXiv:1802.08674, 2018.
5. Chen Y, Cuellar A, Luo H, Modi J, Nemlekar H, Nikolaidis S, Fair contextual multi-armed bandits: Theory and experiments. In Proceedings of the Thirty-Sixth Conference on Uncertainty in Artificial Intelligence, PMLR, 2020;181–190.
6. Chiappa S, Gillam TPS, Path-Specific Counterfactual Fairness, arXiv preprint arXiv:1802.08139, 2018.
7. Chu W, Li L, Reyzin L, Schapire R, Contextual bandits with linear payoff functions. In Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, 2011;208–214.
8. Dee S T, Teachers and the gender gaps in student achievement, Journal of Human Resources, 2007;528–554.
9. Dua D, Graff C, UCI machine learning repository, 2017, http://archive.ics.uci.edu/ml.
10. Ekstrand MD, Tian M, Azpiazu IM, Ekstrand JD, Anuyah O, McNeill D, Pera MS, All the cool kids, how do they fit in?: Popularity and demographic biases in recommender evaluation and effectiveness. In Conference on Fairness, Accountability and Transparency, FAT 2018, 23-24 February 2018, New York, NY, USA, vol. 81 of Proceedings of Machine Learning Research, PMLR, 2018, pp. 172–186.
11. Ekstrand MD, Tian M, Kazi MRI, Mehrpouyan H, Kluver D, Exploring author gender in book rating and recommendation. In Proceedings of the 12th ACM Conference on Recommender Systems, 2018; 242–250.
12. Epstein R, Robertson RE. The search engine manipulation effect (seme) and its possible impact on the outcomes of elections. Proc Natl Acad Sci. 2015;112:E4512–21.

13. Farahat A, Bailey MC, How effective is targeted advertising?. In Proceedings of the 21st international conference on World Wide Web, ACM, 2012;111–120.
14. Ghalme G, Jain S, Gujar S, Narahari Y, Thompson sampling based mechanisms for stochastic multi-armed bandit problems. In Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2017, São Paulo, Brazil, May 8-12, 2017, ACM, 2017;87–95.
15. Gillen S, Jung C, Kearns MJ, Roth A, Online learning with an unknown fairness metric. In Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada, 2018;2605–2614.
16. Gur Y, Zeevi AJ, Besbes O, Stochastic multi-armed-bandit problem with non-stationary rewards. In Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada, 2014;199–207.
17. Hardt M, Price E, Srebro N. et al. *Equality of opportunity in supervised learning*. In Advances in neural information processing systems, 2016;3315–3323.
18. Heidari H, Krause A, Preventing disparate treatment in sequential decision making. In Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden, ijcai.org, 2018;2248–2254.
19. Hoffmann F, Oreopoulos P, A professor like me the influence of instructor gender on college achievement, Journal of Human Resources, 2009;479–494.
20. Huang W, Labille K, Wu X, Lee D, Heffernan N, Fairness-aware Bandit-based Recommendation. In Proceedings of the 2021 IEEE International Conference on Big Data (Big Data), Orlando, FL, USA, December 15-18, 2021;1273–1278.
21. Jabbari S, Joseph M, Kearns MJ, Morgenstern J, Roth A, Fairness in reinforcement learning. In Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017, vol. 70 of Proceedings of Machine Learning Research, PMLR, 2017;1617–1626.
22. Joseph M, Kearns MJ, Morgenstern J, Neel S, Roth A, Meritocratic fairness for infinite and contextual bandits. In Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society, AIES 2018, New Orleans, LA, USA, February 02-03, 2018, ACM, 2018;158–163.
23. Joseph M, Kearns MJ, Morgenstern JH, Roth A, Fairness in learning: Classic and contextual bandits. In Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain, 2016;325–333.
24. Katehakis MN, Veinott AF Jr. The multi-armed bandit problem: decomposition and computation. Math Oper Res. 1987;12:262–8.
25. Kusner MJ, Loftus J, Russell C, Silva R, Counterfactual fairness. In Advances in Neural Information Processing Systems, 2017;4066–4076.
26. Langford J, Zhang T, The epoch-greedy algorithm for contextual multi-armed bandits. In Proceedings of the 20th International Conference on Neural Information Processing Systems, Citeseer, 2007;817–824.
27. Lattimore T, Szepesvári C. Bandit algorithms. Cambridge University Press; 2020.
28. Li F, Liu J, Ji B, Combinatorial sleeping bandits with fairness constraints. In 2019 IEEE Conference on Computer Communications, INFOCOM 2019, Paris, France, April 29 - May 2, 2019, IEEE, 2019;1702–1710.
29. Li L, Chu W, Langford J, Schapire RE, A contextual-bandit approach to personalized news article recommendation. In Proceedings of the 19th international conference on World wide web, ACM, 2010;661–670.
30. Liu Y, Radanovic G, Dimitrakakis C, Mandal D, Parkes DC, Calibrated fairness in bandits, arXiv preprint arXiv:1707.01875, 2017.
31. Metevier B, Giguere S, Brockman S, Kobren A, Brun Y, Brunskill E, Thomas PS, Offline contextual bandits with high probability fairness guarantees. In Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada, 2019;14893–14904.
32. Patil V, Ghalme G, Nair V, Narahari Y, Achieving fairness in the stochastic multi-armed bandit problem. In Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence, (AAAI-20), New York, New York, USA, February 7-12, 2020;5379–5386.
33. Russell C, Kusner MJ, Loftus J, Silva R, When worlds collide: integrating different counterfactual assumptions in fairness. In Advances in Neural Information Processing Systems, 2017;6414–6423.
34. Sun Y, Ramírez I, Cuesta-Infante A, Veeramachaneni K, Learning fair classifiers in online stochastic settings, CoRR, abs/1908.07009 2019.
35. Syrgkanis V, Krishnamurthy A, Schapire RE, Efficient algorithms for adversarial contextual learning. In Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016, vol. 48 of JMLR Workshop and Conference Proceedings, JMLR.org, 2016;2159–2168.
36. Wu Q, Wang H, Gu Q, Wang H, *Contextual bandits in a collaborative environment*. In Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, 2016;529–538.
37. Wu Y, Zhang L, Wu, Counterfactual fairness: Unidentification, bound and algorithm. In Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019, International Joint Conferences on Artificial Intelligence Organization, 2019;1438–1444.
38. Wu Y, Zhang L, Wu X, Tong H, PC-Fairness: A Unified Framework for Measuring Causality-based Fairness. In Annual Conference on Neural Information Processing Systems 2019, December 8-14, 2019, Vancouver, Canada, 2019, Curran Associates, Inc., Dec. 2019;3399–3409.
39. Yao S, Huang B, Beyond parity: Fairness objectives for collaborative filtering. In Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA, 2017;2921–2930.
40. Yang S, Ren T, Shakkottai S, Price E, Dhillon IS, Sanghavi S, Linear bandit algorithms with sublinear time complexity, arXiv preprint arXiv:2103.02729, 2021.
41. Zafar MB, Valera I, Rodriguez MG, Gummadi KP, *Fairness constraints: Mechanisms for fair classification*. In AISTATS, 2017.
42. Zhang J, Bareinboim E, *Fairness in decision-making - the causal explanation formula*. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018, Feb. 2018;2037–2045.
43. Zhang L, Wu Y, Wu X, A causal framework for discovering and removing direct and indirect discrimination. In Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, 2017;3929–3935.
44. Zhu Z, Hu X, Caverlee J, Fairness-aware tensor-based recommendation. In Proceedings of the 27th ACM International Conference on Information and Knowledge Management, ACM, 2018;1153–1162.