# Fairness-aware Bandit-based Recommendation

Wen Huang, Kevin Labille, Xintao Wu
*University of Arkansas*
Fayetteville, AR, USA
{wenhuang, kclabill, xintaowu}@uark.edu

Dongwon Lee
*Penn State University*
University Park, Pennsylvania, USA
dongwon@psu.edu

Neil Heffernan
*Worcester Polytechnic Institute*
Worcester, Massachusetts, USA
nth@wpi.edu

***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 bandit based 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 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 during personalized online recommendation. 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.**

***Index Terms*—Contextual Bandit, Fairness, Online Recommendation**

## I. Introduction

Personalized recommendation based on multi-arm bandit (MAB) algorithms has become a popular topic of research. However, it is also known that such personalization could incur biases or even discrimination. Recently researchers have started taking fairness and discrimination into consideration in the design of MAB based personalized recommendation algorithms [1]–[3]. 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. In this paper, we study how to achieve the user-side fairness in the classic contextual bandit algorithm. The contextual bandit framework [4], 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. Our work is different from existing work of fair bandit-based recommendation, e.g., [1]–[3], [5], [6], which requires some fairness constraint on arms at every round of the learning process.

978-1-6654-3902-2/21/$31.00 

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 three-fold. 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. Second, we provide a theoretical regret analysis to show that our algorithm has a regret bound higher than LinUCB up to only an additive constant. Third, we conduct comprehensive experiment evaluations and report comparisons against baselines in terms of fairness-utility trade-off and effects of various factors and hyper parameters on the performance of our algorithm.

## II. Related Work

Recently researchers have started taking fairness and discrimination into consideration in the design of MAB based personalized recommendation algorithms [1]–[3], [5]–[10]. Among them, [5] 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 [6] for infinite and contextual bandits, [7] for reinforcement learning, [2] for the simple stochastic bandit setting with calibration based fairness. In [11], the authors studied the problem of learning fair stochastic multi-armed bandit where each arm is required to be pulled for

at least a given fraction of the total available rounds. In [12], the authors studied fairness in the setting that multiple arms can be simultaneously played and an arm could sometimes be sleeping. [13] 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. In [14], the authors mainly focused on fairness from the arm perspective. 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, [15] also focused on fairness on the recommended items, i.e., arms. Specifically, they aimed to ensure that each arm is pulled at least a pre-specified fraction of times throughout all times. In [16], the authors used causal inference techniques with counterfactual estimation to propose recommendation policies that jointly optimize the relevance of recommendation and the supplier fairness. 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.

## III. 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 $||\mathbf{x}||_A = \sqrt{\mathbf{x}^{\mathrm{T}} A\mathbf{x}}$.

We use the linear contextual bandit [17] 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.

We adopt the idea of upper confidence bound (UCB) for our personalized recommendation and use the classic LinUCB algorithm as introduced by [18]. LinUCB 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 $\epsilon_t$ 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 $m_a$ 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 $\boldsymbol{\theta}_a$ is expressed as:

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

where $\lambda$ is the penalty factor of the ridge regression. The solution to Equation 1 is:

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

[18] 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

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

The parameter $\lambda$ could be tuned to a suitable value in order to improve the algorithm's performance. The arm-related matrices $A_a = D_a^{\mathrm{T}} D_a + \lambda I_d$ and $\mathbf{b}_a = D_a^{\mathrm{T}} \mathbf{r}_a$ are then iteratively updated. 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.

## IV. Fair Contextual Bandits

### A. 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 (cmr) 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.

### *B. Fair-LinUCB algorithm*

We describe our fair LinUCB algorithm and show its pseudo code in Algorithm 1. The key difference from the traditional LinUCB is the strategy of choosing an arm during recommendation (shown in Line 12 of Algorithm 1). In the remaining of this section, we explain how this new strategy achieves user-side group-level fairness.

**Algorithm 1** Fair-LinUCB

1: Input: $\alpha$ , $\gamma \in \mathbb{R}^+$
2: $\bar{r}^{s^+}, \bar{r}^{s^-} \leftarrow 0$
3: **for** t = 1,2,3,..., T **do**
4: Observe features of all arms $a \in \mathcal{A}_t : \mathbf{x}_{t,a} \in \mathbb{R}^d$
5: **for** $a \in \mathcal{A}_t$ **do**
6: **if** $a$ is new **then**
7: $A_a \leftarrow \lambda \mathbf{I}_d$ (d-Dimension identity matrix)
8: $\mathbf{b}_a \leftarrow \mathbf{0}_{d\times 1}$ (d-Dimension zero vector)
9: $\bar{r}_a^{s^+}, \bar{r}_a^{s^-} \leftarrow 0$
10: **end if**
11: $\hat{\boldsymbol{\theta}}_a \leftarrow A_a^{-1}\mathbf{b}_a$
12: $p_{t,a} \leftarrow \hat{\boldsymbol{\theta}}_a^{\mathrm{T}}\mathbf{x}_{t,a} + \alpha||\mathbf{x}_{t,a}||_{A_a^{-1}} + \mathcal{L}(\gamma, F_a)$
13: **end for**
14: Choose arm $a_t = argmax_{a\in\mathcal{A}_t} p_{t,a}$ with ties broken arbitrarily, and observe a real-valued payoff $r_{t,a_t}$
15: $A_a \leftarrow A_a + \mathbf{x}_{t,a_t}\mathbf{x}_{t,a_t}^{\mathrm{T}}$
16: $\mathbf{b}_a \leftarrow \mathbf{b}_a + r_{t,a_t}\mathbf{x}_{t,a_t}$
17: **if** $S_t = s^+$ **then**
18: update $\bar{r}^{s^+}, \bar{r}_a^{s^+}$ with $r_{t,a_t}$
19: **else**
20: update $\bar{r}^{s^-}, \bar{r}_a^{s^-}$ with $r_{t,a_t}$
21: **end if**
22: **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 lesser 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 Equation 4.

$$p_{t,a} \leftarrow \hat{\boldsymbol{\theta}}_a^{\mathrm{T}}\mathbf{x}_{t,a} + \alpha||\mathbf{x}_{t,a}||_{A_a^{-1}} + \mathcal{L}(\gamma, F_a) \tag{4}$$

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

$$\mathcal{L}(\gamma, F_a) = \frac{\alpha_t||\mathbf{x}_{t,a_m}||_{A_t^{-1}}}{2}(F_a + 1)\gamma \tag{5}$$

$$a_m = argmin_{a\in\mathcal{A}_t}||\mathbf{x}_{t,a}||_{A_a^{-1}} \tag{6}$$

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 $\gamma$ controls fairness-accuracy trade-off on the practical performance of Fair-LinUCB.

### *C. Regret analysis*

In this section, we prove that our Fair-LinUCB algorithm has a high-probability regret bound $R_T \leq C'd\sqrt{T}log(TL)$ ($C'$ is a suitably large constant) 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. Our detailed theoretical results and proofs can be found in [19].

Comparing the regret bound of 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.

## V. Experimental Evaluation

We conduct our empirical evaluation. In Section V-A, we present datasets, reward function, evaluation metrics, and baselines. In Section V-B, we compare our Fair-LinUCB with LinUCB and a naive method that tries to achieve fairness by simply removing from the context sensitive attribute and its correlated attributes. We then conduct comprehensive evaluations on how various factors and hyper parameters would affect the fairness-utility trade-off. Due to space limits, we only report results about noise level in the reward function in Section V-C and include in [19] those results about $\gamma$ that controls the weight of the fairness penalty, arm and user order distribution, and $\alpha$ that controls the balance between exploration and exploitation in our Fair-LinUCB algorithm.

### A. Experiment setup

*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 [20] 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. It is composed of 1,580 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$^{\text{th}}$ 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 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 findings 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 5,000 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 1,500 male individuals and 1,500 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.

*2) Reward function:* We compare Fair-LinUCB against the original LinUCB using a simple reward function wherein we manually set the $\boldsymbol{\theta}^*$ coefficients. The reward $r$ is defined as

$$r = \theta_1^* \cdot x_1 + \theta_2^* \cdot x_2 + \theta_3^* \cdot x_3 \tag{7}$$

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.

*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 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 Section IV-A) of the male group and female group: 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.

*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

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

TABLE I: 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 |

the sensitive attribute and all its correlated attributes from the context. We name this method as Naive in our evaluation. In our Fair-LinUCB and baseline algorithms, we set $\alpha$ (balancing between exploration and exploitation) with the default value of 0.5 in all our experiments.

### *B. Comparison with baselines*

*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 1 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 Figure 1a 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 $\gamma$ coefficient set to 3 (Figure 1b). 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 I.

*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 50k. We report in the last row of Table I 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 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.

TABLE II: Impact of a noisy reward function

| standard deviation of noise | Utility Loss | Reward Difference | Male cmr | Female cmr |
|---|---|---|---|---|
| LinUCB | | | | |
| 0.0 | **0.048** | 0.029 | 0.807 | 0.838 |
| 0.01 | 0.050 | 0.030 | 0.806 | 0.836 |
| 0.1 | 0.056 | 0.047 | 0.788 | 0.835 |
| 0.2 | 0.059 | 0.045 | 0.790 | 0.835 |
| 0.3 | 0.072 | **0.023** | 0.777 | 0.800 |
| 0.4 | 0.084 | 0.067 | 0.749 | 0.816 |
| 0.5 | 0.099 | 0.113 | 0.715 | 0.828 |
| Fair-LinUCB $\gamma = 3$ | | | | |
| 0.0 | **0.052** | **0.000** | 0.819 | 0.819 |
| 0.01 | 0.053 | **0.000** | 0.818 | 0.818 |
| 0.1 | 0.063 | **0.000** | 0.808 | 0.808 |
| 0.2 | 0.074 | **0.000** | 0.801 | 0.801 |
| 0.3 | 0.061 | **0.000** | 0.810 | 0.810 |
| 0.4 | 0.080 | 0.005 | 0.794 | 0.799 |
| 0.5 | 0.099 | 0.025 | 0.765 | 0.790 |

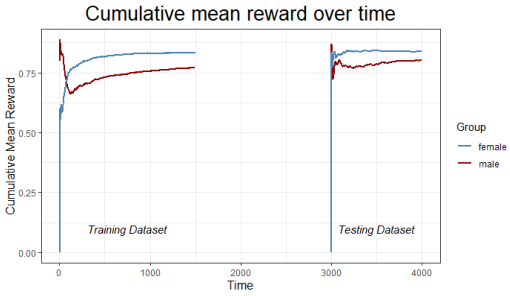
### *C. Impact of a noisy reward function*

We now investigate the impact on the utility loss and fairness for both LinUCB and our Fair-LinUCB with a noisy reward mechanism. We redefine the reward function described in Equation 7 to the following:

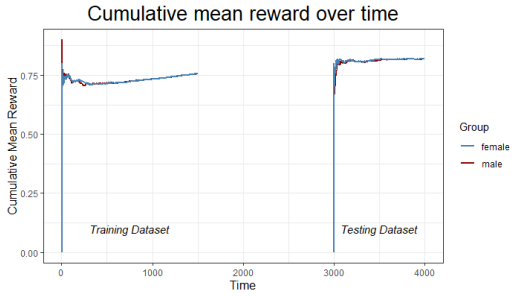
$$r = \theta_1^* \cdot x_1 + \theta_2^* \cdot x_2 + \theta_3^* \cdot x_3 + \eta \tag{8}$$

where $\eta \sim \mathcal{N}(0.0, \sigma)$ is the noise term drawn randomly from a Gaussian distribution with mean 0.0 and standard deviation $\sigma$. While the mean of the noise term remains the same throughout this experiment, we explore various values of standard deviation and report our findings in Table II. The first row of the table, i.e., with a standard deviation of 0.0, reflects the performances without any noise.

Table II shows that, as expected, the performances with a noisy reward function are poorer compared to a reward function with no noise. The best results are achieved with the smallest noise (i.e., $\sigma$=0.01) for both algorithms. We notice that as the noise becomes larger, both LinUCB and Fair-LinUCB suffer from a greater decrease in performances. Indeed, for both algorithms the utility loss almost doubles when the standard deviation of the noise reaches 0.5. However, unlike LinUCB, the reward discrepancy for our Fair-LinUCB appears to be robust against a noisy reward function. Indeed, LinUCB's reward difference worsens as the noise becomes larger, reaching 0.113 when the standard deviation is 0.5. Conversely, the reward discrepancy for our Fair-LinUCB remains zero or almost zero regardless of the magnitude of the noise. In fact, even with a large noise (i.e. $\sigma$ =0.5) the reward difference of our Fair-LinUCB remains smaller than that of LinUCB with a non-noisy reward function.




(a) LinUCB




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

Fig. 1: LinUCB vs Fair-LinUCB with reward function $r$

This experiment shows that as the reward function becomes noisier, not only LinUCB's performance decreases but the gap of the reward discrepancy between the protected group and the privileged group widens. On the other hand, while the performance of our Fair-LinUCB decreases with a noisier reward function, it is robust in terms of fairness, by maintaining a minor or no gap in the reward discrepancy.

## VI. Conclusion and Future Work

In this paper, we have proposed a fair contextual bandit algorithm for personalized recommendation. 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. 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 $\gamma$. 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, and causal bandits [21] that leverage the causal relationship between interventions and outcomes to learn optimal interventions. Finally, we plan to study how to achieve individual fairness and counterfactual fairness in bandits algorithms.

## Acknowledgement

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

## References

[1] L. E. Celis, S. Kapoor, F. Salehi, and N. K. Vishnoi, "An algorithmic framework to control bias in bandit-based personalization," *arXiv preprint arXiv:1802.08674*, 2018.

[2] Y. Liu, G. Radanovic, C. Dimitrakakis, D. Mandal, and D. C. Parkes, "Calibrated fairness in bandits," *arXiv preprint arXiv:1707.01875*, 2017.

[3] Z. Zhu, X. Hu, and J. Caverlee, "Fairness-aware tensor-based recommendation," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. ACM, 2018, pp. 1153–1162.

[4] J. Langford and T. Zhang, "The epoch-greedy algorithm for contextual multi-armed bandits," in *Proceedings of the 20th International Conference on Neural Information Processing Systems*. 2007, pp. 817–824.

[5] M. Joseph, M. J. Kearns, J. H. Morgenstern, and A. Roth, "Fairness in learning: Classic and contextual bandits," in *Advances in Neural Information Processing Systems*, 2016, pp. 325–333.

[6] M. Joseph, M. J. Kearns, J. Morgenstern, S. Neel, and A. Roth, "Meritocratic fairness for infinite and contextual bandits," in *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society, 2018*. ACM, 2018, pp. 158–163.

[7] S. Jabbari, M. Joseph, M. J. Kearns, J. Morgenstern, and A. Roth, "Fairness in reinforcement learning," in *Proceedings of the 34th International Conference on Machine Learning*, 2017, pp. 1617–1626.

[8] R. Burke, "Multisided fairness for recommendation," *arXiv preprint arXiv:1707.00093*, 2017.

[9] R. Burke, N. Sonboli, and A. Ordonez-Gauger, "Balanced neighborhoods for multi-sided fairness in recommendation," in *Conference on Fairness, Accountability and Transparency*, 2018, pp. 202–214.

[10] M. D. Ekstrand, M. Tian, M. R. I. Kazi, H. Mehrpouyan, and D. Kluver, "Exploring author gender in book rating and recommendation," in *Proceedings of the 12th ACM Conference on Recommender Systems*, 2018, pp. 242–250.

[11] V. Patil, G. Ghalme, V. Nair, and Y. Narahari, "Achieving fairness in the stochastic multi-armed bandit problem," *CoRR*, vol. abs/1907.10516, 2019.

[12] F. Li, J. Liu, and B. Ji, "Combinatorial sleeping bandits with fairness constraints," in *2019 IEEE Conference on Computer Communications*. 2019, pp. 1702–1710.

[13] S. Gillen, C. Jung, M. J. Kearns, and A. Roth, "Online learning with an unknown fairness metric," in *Advances in Neural Information Processing Systems*, 2018, pp. 2605–2614.

[14] Y. Chen, A. Cuellar, H. Luo, J. Modi, H. Nemlekar, and S. Nikolaidis, "Fair contextual multi-armed bandits: Theory and experiments," in *Conference on Uncertainty in Artificial Intelligence*. PMLR, 2020, pp. 181–190.

[15] V. Patil, G. Ghalme, V. Nair, and Y. Narahari, "Achieving fairness in the stochastic multi-armed bandit problem," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020, pp. 5379–5386.

[16] R. Mehrotra, J. McInerney, H. Bouchard, M. Lalmas, and F. Diaz, "Towards a fair marketplace: Counterfactual evaluation of the trade-off between relevance, fairness & satisfaction in recommendation systems," in *CIKM*, 2018, pp. 2243–2251.

[17] W. Chu, L. Li, L. Reyzin, and R. Schapire, "Contextual bandits with linear payoff functions," in *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, 2011, pp. 208–214.

[18] L. Li, W. Chu, J. Langford, and R. E. Schapire, "A contextual-bandit approach to personalized news article recommendation," in *Proceedings of the 19th international conference on World Wide Web*. ACM, 2010, pp. 661–670.

[19] W. Huang, K. Labille, X. Wu, D. Lee, and N. Heffernan, "Achieving user-side fairness in contextual bandits," *CoRR*, vol. abs/2010.12102, 2020.

[20] D. Dua and C. Graff, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[21] Y. Lu, A. Meisami, A. Tewari, and W. Yan, "Regret analysis of bandit problems with causal background knowledge," in *Conference on Uncertainty in Artificial Intelligence*. 2020, pp. 141–150.