

Context-aware Multi-stakeholder Recommender Systems

Tahereh Arabghalizi, Alexandros Labrinidis

Department of Computer Science, University of Pittsburgh
tahereh.arabghalizi@pitt.edu, labrinid@cs.pitt.edu

Abstract

Traditional recommender systems help users find the most relevant products or services to match their needs and preferences. However, they overlook the preferences of other sides of the market (aka stakeholders) involved in the system. In this paper, we propose to use contextual bandit algorithms in multi-stakeholder platforms where a multi-sided relevance function with adjusting weights is modeled to consider the preferences of all involved stakeholders. This algorithm sequentially recommends the items based on the contextual features of users along with the priority of the stakeholders and their relevance to the items. Our extensive experimental results on a dataset consisting of MovieLens (1m), IMDB (81k+), and a synthetic dataset show that our proposed approach outperforms the baseline methods and provides a good trade-off between the satisfaction of different stakeholders over time.

Keywords: Recommender Systems; Multi-stakeholder Recommender Systems; Multi-armed Bandits; Contextual Bandits;

Introduction

The pervasiveness of information technology in all aspects of modern life has led to recommender systems being used extensively to personalize the user experience. From e-commerce, social media, advertising, to movie/music services, and everything in between. Traditional recommender systems only consider the needs or preferences of users and ignore the preferences of the other sides of the market which also benefit from the actions of the recommender system, for example, the businesses being recommended to users.

Online recommender systems, which ingest data one observation at a time, have been recently modelled as a very popular problem called “Contextual Multi-armed Bandit”. The multi-armed bandit problem is a form of reinforcement learning where we are given a slot machine with n arms (aka actions) with each arm having a different probability distribution of success. By pulling any of the arms, we receive a random reward or payoff based on that distribution, which is not known in advance. The gambler’s goal is to maximize the rewards earned through a series of lever pulls. At each

trial, the gambler must choose between “exploiting” the machine with the highest expected payoff and “exploring” other machines to get more information about their expected payoffs. Contextual multi-armed bandit is a generalization of the multi-armed bandit problem in which arms are selected based on a given context and the rewards depend on both selected arms and the context (wik 2022).

Motivating Example: In this work we are addressing the problem of how to best recommend coupons (offered by local businesses located nearby) to bus passengers waiting for their bus to arrive. The coupons would be targeted for times when the next bus is expected to be full and encourage the (future) bus passengers to enjoy the recommended offer instead of trying to ride a full bus. Such an online recommender system is required to consider the preferences of all sides of the marketplace (aka stakeholders) and recommend the most relevant coupons to the passengers accordingly. One of the potential stakeholders (other than bus passengers and local businesses) are the minority-owned businesses whose coupons might be ignored by the recommender system due to suffering from the popularity bias problem. Another requirement for such recommender system is to be able to prioritize different stakeholders at different times. Although our motivating application is focusing on people arriving at bus stops, our proposed solution is easily expandable to a much broader application space, which includes people walking around a city and receiving coupons on their mobile phone from nearby businesses.

Background and Related Work

Offline recommendation approaches such as Collaborative Filtering (Ricci, Rokach, and Shapira 2011), which rely on the historical preferences of the users, perform poorly in online recommendation tasks such as ad selection and news recommendations, where the data becomes available over time. Multi-armed bandits (MAB), a classic reinforcement learning problem, have recently drawn attention in online recommendation tasks. They belong to a class of online learning algorithms that provide an approach for the dilemma between exploration and exploitation in order to maximize the expected cumulative payoff/reward up to a finite time horizon T (Auer, Cesa-Bianchi, and Fischer 2002). Bandit algorithms can be categorized into *context-free* and *contextual bandits* depending on whether or not side infor-

mation (aka context) is taken into account. In context-free bandits, a learning algorithm selects an arm from a set of possible arms while the observed reward depends only on the selected arm. In contextual bandits, an arm is chosen in each round based on a given context and the observed reward depends on both the chosen arm and the context (Lu, Pál, and Pál 2010).

Furthermore, *user × item* recommender systems recommend items which are tailored to the users’ needs or preferences. However, there are many real-world applications in which users are not the only stakeholders involved and there may be other individuals or organizations who benefit from the delivery of recommendations (Abdollahpour, Burke, and Mobasher 2017). While a lot of research has used contextual bandits for user-centric online recommender systems (Li et al. 2010; Tang et al. 2014), research on multi-stakeholder recommender systems in an online setting has received less attention. Although, multi-objective multi-armed bandit algorithms (MO-MAB) have been introduced to enable bandit algorithms to optimize multiple objectives simultaneously (Drugan and Nowe 2013; Yahyaa, Drugan, and Manderick 2014; Tekin and Turğay 2018), they have not been used much in multi-stakeholder recommender systems. To the best of our knowledge, (Mehrotra, Xue, and Lalmas 2020) is the only work that extended contextual bandits to multi-objective for recommendations in a multi-stakeholder platform. They proposed an online gradient ascent learning algorithm to maximize the long-term payoffs for different objectives (e.g., diversity and fairness) formalized using the the Generalized Gini Index aggregation function.

Contributions: Unlike the previous work, the focus of our paper is to provide a good level of satisfaction for all stakeholders over time. This is obtained based on a given set of weights that can indicate which stakeholder is prioritized by the system or if all stakeholders should be treated the same way. To this end, we propose to use a contextual multi-armed bandit algorithm in which we define a multi-sided relevance function with adjusting weights that takes the preferences of all involved stakeholders (including users) into account. This algorithm selects items based on the given context, the priority of stakeholders and their relevance to the items. Our contributions are as follows:

- we propose a multi-sided relevance function with adjusting weights (to be used with multi-armed bandit algorithms) which considers the priority of each stakeholder and their relevance to the selected items.
- we define and use a metric to evaluate the satisfaction of stakeholders over time.
- we experimentally evaluated the performance of our approach using bandit algorithms as baselines using synthetic data and data from MovieLens and IMDB.

Problem Statement

In this paper, we address *the problem of recommending items to users, considering the relevance and priority of every stakeholder involved, while providing a good level of satisfaction for them over time*. We see this problem as a contex-

Algorithm 1 MAB with multi-sided relevance function

Inputs: MAB: base algorithm, A: arms, T: rounds, $u_1, u_2, \dots, u_t$: users, C_u : context of user u (if any), $w_1, w_2, \dots, w_n$: weights of stakeholders, δ : relevance threshold

```

for  $t = 1, 2, \dots, T$  do
   $a_t \leftarrow \text{MAB.selectArm}(A, u_t, C_{u_t})$ 
   $s_1^{a_t}, s_2^{a_t}, \dots, s_n^{a_t} \leftarrow$  compute relevance scores between  $a_t$  and all stakeholders
   $r^{a_t} = w_1 s_1^{a_t} + \dots + w_n s_n^{a_t} \leftarrow$  compute multi-sided relevance score for  $a_t$ 
  if  $r^{a_t} > \delta$  then
     $\text{reward}^{a_t} = 1$ 
  else
     $\text{reward}^{a_t} = 0$ 
  end if
  MAB.UpdateReward( $u_t, C_{u_t}, a_t, \text{reward}^{a_t}$ )
end for

```

tual bandit problem which is a generalization of the multi-armed bandit problem that extends the model by adding context about the users. Given that, we configure our recommendation problem as follows:

- We need to recommend an item to each user in an online multi-stakeholder platform. (e.g., stakeholders include bus passengers, coupon suppliers and minority-owned businesses)
- We have an already-known finite set of items/arms (e.g., available coupons at each bus stop).
- We are provided with side information about the users (e.g., age range and gender of bus passengers).
- We are provided with the priority (weight) of each stakeholder in the system.
- We aim to maximize the cumulative payoffs and make a reasonable balance between the satisfaction of all stakeholders over time.

Proposed Solution

To address the problem, we propose to use a contextual multi-armed bandit algorithm in which we model a multi-sided relevance function with adjusting weights to consider the priority of every stakeholder and their relevance to each chosen item. In user-centric recommender systems, the item selection strategy is focused on user satisfaction which is usually obtained based on whether the user clicks on the recommended item or not. However, in a multi-stakeholder recommender system, the satisfaction of all involved stakeholders need to be taken into account.

In our proposed approach which is described in Algorithm 1, in each round of playing, one appropriate arm is selected based on the policy of a multi-armed bandit algorithm which considers the contextual features (if available) of the current user. To estimate the reward, we propose a linear relevance function which is defined as the weighted sum of the relevance score of each stakeholder to each selected arm a . The relevance scores can be calculated based on the feedback of users and other stakeholders about the quality of the recommended items. The relevance function is defined as below:

$$r^a(w, s) = w_1 s_1^a + \dots + w_n s_n^a \quad (1)$$

where $r^a \in [0, 1]$ is the estimated multi-sided relevance score, n is the number of stakeholders, $s_1^a, s_2^a, \dots, s_n^a$ are the relevance scores of arm a for each stakeholder where each score is either 1 (relevant) or 0 (non-relevant), w_i is the given priority or weight of each stakeholder where $\sum_{i=1}^n w_i = 1$. The relevance threshold, δ is used to let the policy draw a random variable (aka reward) from a Bernoulli distribution ($reward \in \{0, 1\}$). The multi-armed bandit policy updates its reward in each round to improve its arm selection strategy with the new observation and continues exploration and exploitation to maximize the total reward until round T ends.

Evaluation Methodology and Metrics

Online algorithms, such as bandits, do not have access to full sets of data to be trained on. Instead, learning occurs incrementally as data accumulates. However, the performance of these algorithms can be evaluated offline through back-testing using offline evaluation methods. The offline evaluation of multi-armed bandit algorithms can avoid the drawbacks of online evaluation, such as the high evaluation costs, negative effects on the users, etc. Evaluating the performance of these algorithms using historical datasets, however, is challenging. For instance, data can be biased depending on how it was generated. Moreover, the algorithm often generates recommendations that differ from the ones seen by users in the historical datasets. As a result, you cannot provide a reward value for these recommendations because you cannot predict a user’s reaction to a recommendation they never saw.

Evaluation Methodology: *Replay* is a one of the well-known methodologies which is extensively studied in the literature to deal with the above issues. In this evaluation methodology, for each record in the historical data, the bandit algorithm is asked to choose an arm. If this arm is the same as what the user saw in the historical data, its reward is revealed and the replay methodology takes it into account to evaluate the performance of the algorithm. If the arm is different, that record is ignored by the methodology (Li et al. 2011).

Since we address the problem of recommendation in a multi-stakeholder platform, we apply the replay evaluation methodology in a way that it accepts an arm if its multi-sided relevance score is greater than a threshold (δ). In other words, it discards an arm if it is not relevant to the preferences of multiple stakeholders given their past feedback.

Evaluation Metrics: We use two types of metrics to evaluate the performance of the proposed approach and the baseline methods:

- **Mean reward:** after a bandit algorithm selects an arm, the relevance function returns a score in each round of playing and a reward $\in \{0, 1\}$ is revealed accordingly. The final objective of a bandit algorithm is to maximize the total reward when the rounds end. We accumulate the reward values for the accepted arms in each round and return the average of the rewards (aka mean reward) to obtain the performance of different algorithms.

- **Satisfaction percentage:** since we have multiple stakeholders, we need to compare the different algorithms in terms of their satisfaction over time. For this purpose, we compute the percentage of times (out of the total rounds) that an algorithm selects an arm based on the given priority of a stakeholder and define it as the satisfaction percentage of that stakeholder. Given the priority of stakeholders, our goal is to have a reasonable balance between the satisfaction of all stakeholders over time.

Experimental Evaluation

In this section, we present several experiments on a movie recommendation data to validate the performance of our proposed approach. We then compare our method to the baselines using the experimental results.

Dataset

As there are currently no relevant datasets available for a multi-stakeholder platform in the “local business - bus passengers” domain, we created a synthetic dataset, by combining MovieLens (1m) (mov 1998) with IMDB (81k+) (Leone 2019) to include other features such as movie production companies. In our experiments, we assume a scenario where there are three different stakeholders involved in the recommendation platform. Based on this data, movies are considered as coupons, users as bus passengers (first stakeholder), movie production companies as local businesses (second stakeholder) and movies with a specific genre as minority-owned businesses (third stakeholder). A more detailed explanation of data is as follows:

- **User-side data:** the ratings of the users to the movies (integers between 1 and 5) along with the users’ demographic features (e.g., age range and gender as context) are used as the user-side data in our experiments. The missing values in the user rating data are filled by zero assuming that if a user does not rate an item, it is likely that they do not like the item. The total number of users, movies and non-zero user ratings used in our experiments are 6,040, 2,392 and 690,041 respectively.
- **Supplier-side data:** as there aren’t any ratings data for production companies towards the users in the MovieLens or IMDB data, we generate a synthetic dataset. For each production company, we use a Truncated Normal Distribution with mean = 3 and standard deviation = 1, that generates random integer numbers between 1 and 5 as the ratings of that production company towards the users. The total number of ratings generated for 1,023 production companies is 6,178,920.
- **Third-stakeholder-side data:** we consider movies with a particular genre as minority-owned businesses (third stakeholder). In our experiments, we use ‘Sci-Fi’ as the minority genre because firstly, only about 7% of all movies in the MovieLens data are ‘Sci-Fi’ and secondly, about 20% of the top 100, 200 and 300 movies (movies with the highest number of ratings) are Sci-Fi which matches the ratio of the number of regular to minority-owned businesses in real life scenarios.

Multi-armed Bandit Algorithms and Baselines

As mentioned before, we proposed to use a multi-sided relevance function with a contextual multi-armed bandit algorithm to consider the priority and relevance of all stakeholders involved. However, to have a broader comparison, we use both context-free and contextual bandits as the base multi-armed bandit algorithms applied in Algorithm 1. These algorithms are as follows:

- **Random** (context-free): it randomly selects an arm to pull at each round.
- **Epsilon-greedy** (context-free): in each round, this algorithm selects a random arm with probability ϵ , and chooses the arm with the highest empirical mean with probability $1 - \epsilon$ (Cesa-Bianchi and Fischer 1998).
- **UCB1** (context-free): an upper confidence bound has to be calculated for each arm for the algorithm to be able to choose an arm in each round (Auer, Cesa-Bianchi, and Fischer 2002).
- **LinUCB** (contextual): Linear Upper Confidence Bound (LinUCB) is a classical contextual multi-armed bandit algorithm in which there is a linear dependency between the expected reward of an arm and its context (Li et al. 2010). In our experiments, we call the LinUCB algorithm that is applied with our multi-sided relevance function "Multi-stakeholder LinUCB" or **MS-LinUCB**.

We additionally present three different baseline methods, corresponding to the three stakeholders, to compare with our approach. These contextual bandit methods use a single-sided relevance function, $r^a(w, s) = ws^a$, that prioritizes only one stakeholder and ignores the others. These baselines are defined as follows:

- **UC-LinUCB**: user-centered LinUCB only considers the relevance of the users to the chosen arms in the relevance function.
- **SC-LinUCB**: supplier-centered LinUCB only considers the relevance of the suppliers of the chosen arms to the users in the relevance function.
- **MC-LinUCB**: minority-centered LinUCB only considers the relevance of the minority stakeholders to the chosen arms in the relevance function.

Experimental Setup

In this section, we describe different settings that we used in our experiments:

- *number of rounds*: because a bandit is an online learner, we need to construct a simulation environment to train the bandit. At each training iteration (aka round), the bandit observes data from the past, updates its decision-making policy, takes an arm, and observes a reward for this arm. To provide enough exploration opportunities, we set the number of rounds (parameter T in Algorithm 1) to 15,000 and we use a stream of 15,000 users who rated the top movies (arms) in the past to train and evaluate the bandits.

- *number of arms*: we vary the number of arms (parameter A in Algorithm 1) and use 100, 200, and 300 top movies as the number of arms in different experiments. We use top movies with the highest number of ratings as arms because the model can become stuck in offline evaluation if there are too few ratings for a movie.
- *users' context*: the contextual features of users can vary from their profiles, demographic information, their past interaction, etc. In this work, we use the affinity of users to movie genres (computed based on the movies that users rated before) along with their age range and gender as the context of users (parameter C_u in Algorithm 1).
- *relevance scores*: we follow the below details to obtain the relevance scores ($s_1^a, s_2^a, \dots, s_n^a$ in Equation 1) for the three mentioned stakeholders.
 - *user relevance*: in the MovieLens dataset, user ratings are integer values from 1 to 5 where the average ratings of all movies is about 3.5. To compute the relevance scores of users, we assume that if a selected movie has received a rating greater than 3.5 from a user, the movie is likely relevant to the preferences of that user and he/she will click on it. So let $R(u, m)$ be the rating of the user u for the movie m , the user relevance score for this movie can be represented as follows:

$$s_{user} = \begin{cases} 1 & R(u, m) > 3.5 \\ 0 & otherwise \end{cases} \quad (2)$$

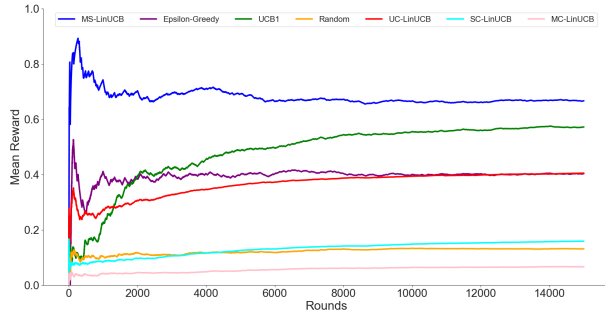
- *supplier relevance*: in our synthetic data, the ratings of suppliers to users are generated as integer values from 1 to 5 where the average ratings is 3. Therefore, to calculate the relevance scores of suppliers, we make this assumption that if a movie production company (supplier) has given a rating greater than 3 to the current user, that user is likely relevant to that company's preferences. So let $R'(p, u)$ be the rating of the production company p for the user u , the supplier relevance score can be obtained as follows:

$$s_{supplier} = \begin{cases} 1 & R'(p, u) > 3.0 \\ 0 & otherwise \end{cases} \quad (3)$$

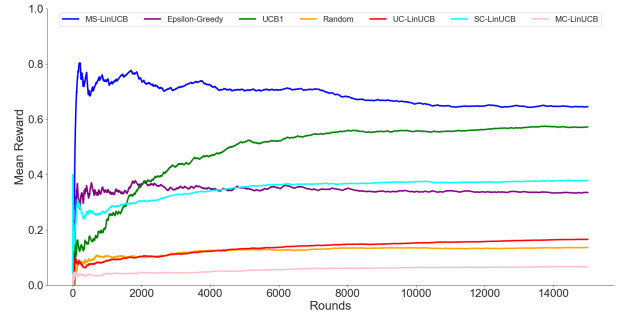
- *minority relevance*: to compute the relevance scores for the third stakeholder (i.e., minority-owned businesses), we simply check the genre of the selected movie as we used 'Sci-Fi' as the minority genre:

$$s_{minority} = \begin{cases} 1 & \text{if movie genre is Sci-Fi} \\ 0 & otherwise \end{cases} \quad (4)$$

- *weights of stakeholders*: we use different sets of weights (parameters $w_1, w_2, \dots, w_n$ in Algorithm 1) for prioritizing different stakeholders in our multi-sided relevance function. For instance, (0.33, 0.33, 0.33) are used as the weights corresponding to the users, suppliers and minority stakeholders respectively when the stakeholders do not have any priority over each other. We use (0.5, 0.25, 0.25) to prioritize the users, (0.25, 0.5, 0.25) to prioritize the

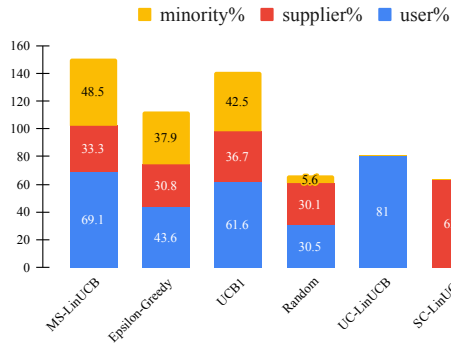


(a) stakeholders' weights: (user: 0.5, supplier: 0.25, minority: 0.25)

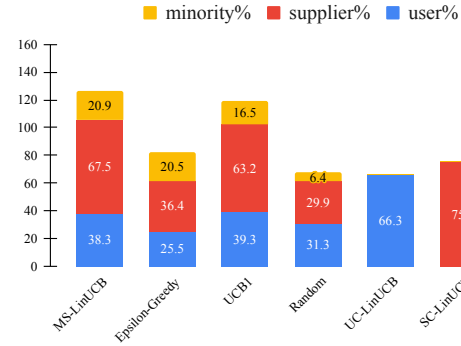


(b) stakeholders' weights: (user: 0.25, supplier: 0.5, minority: 0.25)

Figure 1: Mean reward for 100 arms when prioritizing users (a) and suppliers (b)



(a) stakeholders' weights: (user: 0.5, supplier: 0.25, minority: 0.25)



(b) stakeholders' weights: (user: 0.25, supplier: 0.5, minority: 0.25)

Figure 2: Satisfaction percentage for 100 arms when prioritizing users in (a) and suppliers in (b)

suppliers and (0.25, 0.25, **0.5**) to prioritize the minority stakeholders over the other two stakeholders.

- *relevance threshold*: the relevance threshold (parameter δ in Algorithm 1) indicates whether the selected arm is accepted (reward = 1) or not (reward = 0). We choose the value for the parameter δ such that the relevance of at least two stakeholders to the selected arm at each round is taken into account. Given the different sets of weights for stakeholders, we adjust the relevance threshold to an appropriate value so it can meet the aforementioned requirement. In this paper, we only present the results of experiments with $\delta = 0.5$ for stakeholders' weights mentioned in the previous bullet point. It is worth noting that, we tuned the threshold with different values (e.g., 0.8) while using different sets of weights, e.g., (0.33, 0.33, 0.33), (0.8, 0.1, 0.1), (0.1, 0.8, 0.1), and (0.1, 0.1, 0.8), in different experiments and achieved similar results.
- *hyperparameters*: α and ϵ are the hyperparameters in the LinUCB and Epsilon-Greedy algorithms respectively. They determine the emphasis of exploration versus exploitation and need to be tuned properly. We tuned both parameters with a range of appropriate values from 0.1 to 1.0 but only present the experimental results with $\alpha = 0.6$ and $\epsilon = 0.15$ for this paper.

Experimental Results

Results with 100 arms: we carried out several experiments with different sets of weights when the number of top movies (arms) provided for the bandits is 100. As you can see (Figure 1), our proposed approach, MS-LinUCB, has the highest mean reward compared to the context-free bandits and the single-sided baselines. This indicates that exploiting the context is useful in better learning the quality of arms. Figure 2 illustrates the satisfaction percentage for all approaches when prioritizing users (Figure 2a) and suppliers (Figure 2b). When the system prioritizes one stakeholder, our approach selects more items based on the preferences of that stakeholder (about 70% of times) while it selects items based on the preferences of other stakeholders less frequently. MS-LinUCB provides a better level of satisfaction for all stakeholders compared to other methods as it selects items according to the given context, the priority of the stakeholders, and their relevance to the items. Similar insights are obtained when using other weights but we did not include them in this paper due to the lack of space. It is worth mentioning that the sum of the satisfaction percentages is not equal to 100% because the methods could select items that are relevant to multiple stakeholders simultaneously and therefore there are overlaps between the sets of items that are chosen for different stakeholders.

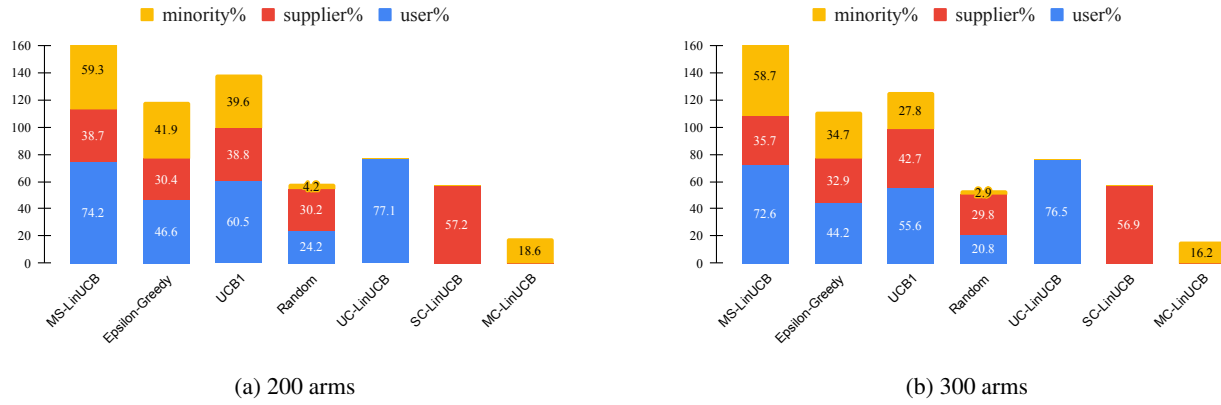


Figure 3: Satisfaction percentage for 200 (a) and 300 (b) arms when prioritizing users with stakeholders’ weights (user: 0.5, supplier: 0.25, minority: 0.25)

Sensitivity Analysis: we also performed multiple sensitivity analyses to see how the satisfaction percentage changes with different number of arms and different set of weights. Figures 3a and 3b, where we prioritize users over other stakeholders for 200 and 300 top movies (arms), show that MS-LinUCB outperforms the other methods in terms of satisfaction percentage and since the user is prioritized, MS-LinUCB selects items based on the users’ preferences more than 70% of times. Given different number of arms and different weights for stakeholders, our results indicate that our proposed approach can make a reasonable balance between the satisfaction of stakeholders over time.

Conclusions

In this paper, we addressed the problem of recommending items in a multi-stakeholder platform where stakeholders can be prioritized. To consider the relevance and priority of all involved stakeholders, we proposed a linear multi-sided relevance function with adjusting weights to be used along with a contextual multi-armed bandit algorithm. Our experimental results showed that our proposed approach outperforms other base bandit algorithms and single-sided baselines in terms of mean reward and satisfaction percentage.

Acknowledgment

This work is part of the PittSmartLiving project which is supported by NSF award CNS-1739413.

References

Abdollahpour, H.; Burke, R.; and Mobasher, B. 2017. Recommender systems as multistakeholder environments. In *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization*, 347–348.

Auer, P.; Cesa-Bianchi, N.; and Fischer, P. 2002. Finite-time analysis of the multiarmed bandit problem. *Machine learning* 47(2):235–256.

Cesa-Bianchi, N., and Fischer, P. 1998. Finite-time regret bounds for the multiarmed bandit problem. In *ICML*, volume 98, 100–108.

Drugan, M. M., and Nowe, A. 2013. Designing multi-objective multi-armed bandits algorithms: A study. In *The 2013 International Joint Conference on Neural Networks (IJCNN)*, 1–8. IEEE.

Leone, S. 2019. Imdb movies extensive dataset.

Li, L.; Chu, W.; Langford, J.; and Schapire, R. E. 2010. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, 661–670.

Li, L.; Chu, W.; Langford, J.; and Wang, X. 2011. Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. In *Proc of the 4th ACM intl conference on Web search and data mining*, 297–306.

Lu, T.; Pál, D.; and Pál, M. 2010. Contextual multi-armed bandits. In *Proceedings of the Thirteenth international conference on Artificial Intelligence and Statistics*, 485–492. JMLR Workshop and Conference Proceedings.

Mehrotra, R.; Xue, N.; and Lalmas, M. 2020. Bandit based optimization of multiple objectives on a music streaming platform. In *Proc of the 26th ACM SIGKDD Intl Conference on Knowledge Discovery & Data Mining*, 3224–3233.

1998. Movielens dataset.

Ricci, F.; Rokach, L.; and Shapira, B. 2011. Introduction to recommender systems handbook. In *Recommender systems handbook*. Springer. 1–35.

Tang, L.; Jiang, Y.; Li, L.; and Li, T. 2014. Ensemble contextual bandits for personalized recommendation. In *Proc. of the 8th ACM Conf. on Recommender Systems*, 73–80.

Tekin, C., and Turgay, E. 2018. Multi-objective contextual multi-armed bandit with a dominant objective. *IEEE Transactions on Signal Processing* 66(14):3799–3813.

2022. wikipedia: Multi-armed bandit.

Yahyaa, S. Q.; Drugan, M. M.; and Manderick, B. 2014. The scalarized multi-objective multi-armed bandit problem: An empirical study of its exploration vs. exploitation tradeoff. In *2014 International Joint Conference on Neural Networks (IJCNN)*, 2290–2297. IEEE.