Learning Multi-Objective Rewards and User Utility Function in Contextual Bandits for Personalized Ranking

Nirandika Wanigasekara[1] , **Yuxuan Liang**[1] , **Siong Thye Goh**[2] , **Ye Liu**[1] , **Joseph Jay Williams**[3] and **David S. Rosenblum**[1]

[1]National University of Singapore
[2]Singapore Management University
[3]University of Toronto
{nirandiw, liangyx, liuye, david}@comp.nus.edu.sg, stgoh@smu.edu.sg , williams@cs.toronto.edu

Abstract

This paper tackles the problem of providing users with ranked lists of relevant search results, by incorporating contextual features of the users and search results, and learning how a user values multiple objectives. For example, to recommend a ranked list of hotels, an algorithm must learn which hotels are the right price for users, as well as how users vary in their weighting of price against the location. In our paper, we formulate the context-aware, multi-objective, ranking problem as a Multi-Objective Contextual Ranked Bandit (MOCR-B). To solve the MOCR-B problem, we present a novel algorithm, named Multi-Objective Utility-Upper Confidence Bound (MOU-UCB). The goal of MOU-UCB is to learn how to generate a ranked list of resources that maximizes the rewards in multiple objectives to give relevant search results. Our algorithm learns to predict rewards in multiple objectives based on contextual information (combining the Upper Confidence Bound algorithm for multi-armed contextual bandits with neural network embeddings), as well as learns how a user weights the multiple objectives. Our empirical results reveal that the ranked lists generated by MOU-UCB lead to better click-through rates, compared to approaches that do not learn the utility function over multiple reward objectives.

1 Introduction

There are many learning settings where we want to tease out the human preferences to provide users with ranked lists of resources to choose from, such as suggesting hotels on TripAdvisor, jobs on LinkedIn, or products to buy on Amazon. In this study, we focus on the problem of actively choosing ranked lists of resources to present to users, and dynamically using users' interactions (e.g., user clicks, ratings, feedback) to decide which resources to present in the future. Such problems often rely critically on using contextual information about the users and resources [Xiang *et al.*, 2010; Zamani *et al.*, 2017].

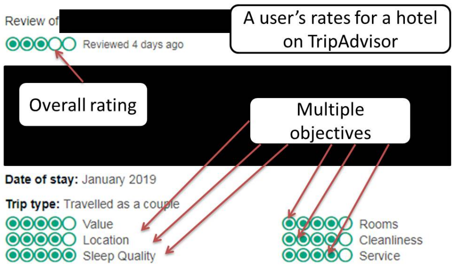

Figure 1: A user's overall rating and ratings in different objectives for a hotel in TripAdvisor

For example, consider an online ranking algorithm that aims to learn which hotels to suggest to users in TripAdvisor. The algorithm can access context information about the user, such as users' previous reviews, ratings, location, or the user's current activity (e.g., relaxing, jogging). The algorithm can also access context information about the hotels such as location, services provided, and stars-level. When suggesting hotels for a user to stay, in addition to the overall rating, the algorithm can obtain multiple reward signals (i.e., location rating, value for money, cleanliness rating, business service rating) about a hotel as shown in Figure 1. Users may value these rewards for the multiple objectives differently when deciding to book a hotel. For instance, whether a user reserves a hotel would reflect the extent to which the user values the hotel's location rating, versus the room services ratings. The algorithm must learn to model how a user's overall rating of a hotel depends on the user's weighting of these multiple objective rewards. However, learning the users' utility function over the multi-objective rewards is a challenging task since both the utility function and the multi-objective rewards are often unknown a priori. Therefore, an algorithm must combine multiple capacities: (1) online learning by trading off exploration and exploitation in suggesting resources, (2) discovery and modeling of how contextual features of users and resources are mapped to rewards, and (3) discovery and modeling of how an individual's preferences map multiple objective or competing rewards to actual behavior. Also, incorporating high-dimensional and unstructured context information effectively to the learning process is a non-trivial task.

The extant literature contains efforts to solve this problem by optimizing a single objective function, such as the click-through-rate (CTR) or query relevancy (QR) [Liu, 2009; Zamani *et al.*, 2017]. We formulate the above context-aware, multi-objective, ranking problem as a novel Multi-Armed Bandit (MAB) [Robbins, 1985] problem that considers multi-objective reward vectors, contextual information and ranking. Our formulated problem is named as the Multi-Objective Contextual Ranked Bandit (MOCR-B) problem. In the MOCR-B problem, a learning algorithm observes a context for each objective, based on the querying user and available resources at the beginning of each trial. Then, it sequentially selects a ranked list of web resources to serve to the user based on the context of the user and web resources. In each trial, a user clicks on a relevant web resource from the ranked list and reveals a reward vector. Based on the user's click behavior and the revealed reward vectors, the algorithm adapts its ranking-strategy to minimize the total number of poor rankings based on the rewards in each objective.

To tackle the MOCR-B problem, in this paper, we present a novel algorithm, Multi-Objective Utility-UCB (MOU-UCB), based on generalized linear contextual bandits [Li *et al.*, 2017] and neural network embeddings [Mikolov *et al.*, 2013]. In a generalized linear contextual bandit, the stochastic reward is generated from an unknown distribution, where the expected reward can be a linear, logistic or probit model conditioned on the context and the chosen resource. To maximize the long-term reward in each objective, MOU-UCB first models the expected payoff of each objective of a resource as a linear function of a context feature vector, and it learns the goodness of the match between the user's context and a resource's expected multi-objective reward. MOU-UCB then learns an unknown utility function over the multiple objective rewards for each user based on the user's click behavior and ranks the web resources to minimize the total number of poor rankings. There have been past different lines of work on contextual bandits, ranking, and weighting the utility reward. Our contribution is on combining these, as discussed in detail in the Related Work section. To the best of our knowledge, our work is the first to present a bandit solution that combines context-awareness, multi-objective optimization, and ranking. We summarize the main contributions of this paper as follows:

- We present a novel problem formulation to handle context information, user preferences and multi-objective reward functions in online ranking called Multi-Objective Contextual Ranked Bandits (MOCR-B).
- We develop a novel algorithm, named MOU-UCB, to solve the MOCR-B problem. MOU-UCB models context information in each objective and directly learns the utility over the multiple objectives for each user to generate ranked lists that maximizes the number of clicks.
- We demonstrate the effectiveness of our algorithm using real-world data from TripAdvisor.

2 Problem Definition

In this section, we define the Multi-Objective Contextual Ranked Bandit (MOCR-B) problem, where the goal is to rank resources to maximize user satisfaction based on multiple objective rewards. In the MOCR-B problem setting, users interact with a system over several trials. An algorithm solving the MOCR-B problem must learn to provide users with a list of resources (also called arms). The resources are ranked according to their utility for the user, where the utility function is a weighted combination of multiple observed rewards in different objectives. Also, the rewards in each multiple objective are a function of contextual features of the user *and* the resource.

In this problem setting, there is a set of users denoted as $\mathcal{U}$, a set of arms denoted as $\mathcal{A}$, I number of objectives, and K number of ranks. An algorithm solving the MOCR-B problem runs over a sequence of trials indexed by $t \in [T]$.

On trial t, the algorithm presents a ranked list $\tilde{\mathcal{A}}(t, K)$ to a user $u_t \in \mathcal{U}$. Let $\tilde{\mathcal{A}}(t, K) = \{a_k(t) | k \in [K]\}$ where $a_k(t)$ is the resource displayed in the k^{th} rank. At trial t, the algorithm must choose a set of resources with a high utility for a user. More precisely the arm chosen by the algorithm at rank k must maximize the expected binary utility reward $f_k(t, u_t)$, which is 1 if the user selects the arm and 0 otherwise.

In the MOCR-B problem setting, the utility reward is *unknown* a priori, but it is known that $f_k(t, u_t)$ is associated with multiple objective rewards. Let $\mathbf{r}_a(t) := [r_a^1(t), \dots, r_a^I(t)]$ be the multiple objective rewards where $a \in \tilde{\mathcal{A}}(t, K)$ and $\mathbf{r}_a(t) \in [0, 1]^I$. Then,

$$f_k(t, u_t) = g(\mathbf{r}_a(t)) + \epsilon \tag{1}$$

where $g(.)$ is an unknown function and ϵ is the 1-subgaussian error term. In this problem setting, the multiple objective rewards $\mathbf{r}_a(t)$ are an unknown a priori too, but they are associated with the context of users and arms. Suppose the length of the context vector is l. Formally, let $\mathbf{x}_a^i(t) \in \mathbb{R}^l$ be the context feature vector for an arm $a \in \mathcal{A}$ in objective i where $i \in [I]$. Let $r_a^i(t)$ be the corresponding multi-objective reward in $\mathbf{r}_a(t)$. Then,

$$r_a^i(t) = h(\mathbf{x}_a^i(t)) + \epsilon' \tag{2}$$

where $h(.)$ is an unknown function and ϵ' is the 1-subgaussian error term with mean zero. When the algorithm presents the ranked list $\tilde{\mathcal{A}}(t, K)$ to the user u_t, the user selects arms he found to be useful. Additionally, for useful arms the users provide multiple objective rewards. Thus, in trial t, for each resource $a_k(t) \in \tilde{\mathcal{A}}(t, K)$ the algorithm observes the corresponding utility reward $f_k(t, u_t)$, and the multiple objective rewards $\mathbf{r}_{a_k(t)}(t)$. In the MOCR-B problem setting, the utility reward function $g(.)$ and the multi-objective reward function $h(.)$ are an unknown a priori. The key to solving the MOCR-B problem is to learn $g(.)$ and $h(.)$ while offering a minimal number of sub-optimal arms to users.

3 MOU-UCB Algorithm

To solve the MOCR-B problem, we have chosen generalized linear contextual bandits. As $f_k(t, u_t) \in \{0, 1\}$, we modeled the utility reward function $g(.)$ as a logistic function with an unknown user-specific coefficient vector $\boldsymbol{\beta}_{u_t}^{k*}$ as given in Equation 3. Since $r_a^i(t) \in [0, 1]$ and continuous, we modeled the multi-objective reward function $h(.)$ as a linear function with an unknown arm-specific coefficient vector $\boldsymbol{\theta}^*_{i,a}$ as given in Equation 4.

$$f_k(t, u_t) = 1/1 + \exp\{-\mathbf{r}_a(t)^\intercal \boldsymbol{\beta}^{k*}_{u_t}\} + \epsilon \tag{3}$$

$$r^i_a(t) = \mathbf{x}^i_a(t)^\intercal \boldsymbol{\theta}^*_{i,a} + \epsilon' \tag{4}$$

As a result, the MOCR-B problem introduced in the previous section can be reduced to a parametric bandit problem where we need to learn the optimal $\beta^{k*}_{u_t}$ for all $u \in \mathcal{U}, k \in [K]$ and the optimal $\boldsymbol{\theta}^*_{i,a}$ for $a \in \mathcal{A}, i \in [I]$ with minimum regret. The rest of this section describes the MOU-UCB algorithm we designed to solve this reduced MOCR-B problem.

3.1 Ranking Model

MOU-UCB runs a separate multi-armed bandit instance for each rank and selects the arm for the respective rank. Let $\Pi = \{M_k | k \in [K]\}$ be the set of these multi-armed bandit instances. When ranking resources for a user, each M_k selects the resource to be shown at rank k. The goal of each M_k is to choose an arm a that has the maximum expected utility reward $g(\mathbf{r}_a(t)^\intercal \boldsymbol{\beta}^{k*}_{u_t})$ at trial t. Note that both $\boldsymbol{\beta}^{k*}_{u_t}$ and $\mathbf{r}_a(t)$ are unknown at the beginning of a trial. Next, we explain how each of these model parameters is learned.

Modeling the Multi-Objective Reward

First, we explain how $\mathbf{r}_a(t)$ is estimated. At the beginning of the trial t, the algorithm observes the context of the user and arm. Let $\{\mathbf{x}^i_a(t) | a \in \mathcal{A}, i \in [I]\}$ be this context set. We assumed in expectation the multi-objective reward to be linear with the context feature vector and unknown coefficient vector $\boldsymbol{\theta}^*_{i,a}$ as described in Equation 4. Suppose $\hat{\boldsymbol{\theta}}_{i,a}$ is the estimate of $\boldsymbol{\theta}^*_{i,a}$ after t trials. MOU-UCB uses ridge regression [Li *et al.*, 2010] and minimizes the following equation to calculate $\hat{\boldsymbol{\theta}}_{i,a}$.

$$\sum_{s=1}^{t-1} \left(r^i_a(s) - (\mathbf{x}^i_a(s))^\intercal \hat{\boldsymbol{\theta}}_{i,a} \right)^2 + \lambda ||\hat{\boldsymbol{\theta}}_{i,a}||^2 \tag{5}$$

where λ is the regularization parameter. In our problem setting at trial t an algorithm has access only to the sequential data that arrived in the previous trials. Unlike in a supervised learning setting, this dataset is not a fair sample of the population. Thus, the expected mean reward $h(\mathbf{x}^i_a(t)^\intercal \hat{\boldsymbol{\theta}}_{i,a})$ calculated based on the $\hat{\boldsymbol{\theta}}_{i,a}$ has a margin of error. We can show that this margin of error can be upper bounded as $\alpha ||\mathbf{x}^i_a(t)||_{\mathbf{A}_a[i]^{-1}}$ where α is an exploration parameter that needs to be tuned and $||\mathbf{x}^i_a(t)||_{\mathbf{A}_a[i]^{-1}}$ is the weighted l_2-norm of the context vector associated with a positive-definite matrix $\mathbf{A}_a[i]$. The matrix $\mathbf{A}_a[i] = \sum_{s=0}^{t} \mathbf{x}^i_a(s)(\mathbf{x}^i_a(s))^\intercal$. In order to account for this margin of error, we decided to model the multi-objective reward in trial t, denoted as $\hat{r}^i_a(t)$, as the sum of the empirical mean reward and the one sided confidence interval of the empirical mean as shown below.

$$\hat{r}^i_a(t) = (\mathbf{x}^i_a(t))^\intercal \hat{\boldsymbol{\theta}}_{i,a} + \alpha ||\mathbf{x}^i_a(t)||_{\mathbf{A}_a[i]^{-1}} \tag{6}$$

This model allows us to efficiently estimate multi-objective rewards when users and arms do not have a lot of historical data. As the number of trials increases, our algorithm can discover how contextual features of users and resources are mapped to the rewards in each objective.

Modeling the Utility Reward

Now we explain how $\boldsymbol{\beta}^{k*}_{u_t}$ is learned in each $M_k \in \Pi$ instance. Each M_k instance selects a resource $a_k(t)$ that maximizes the expected utility reward $g(\mathbf{r}_a(t)^\intercal \boldsymbol{\beta}^{k*}_{u_t})$ for rank k. Suppose $\hat{\boldsymbol{\beta}}^k_{u_t}$ is our current maximum likelihood estimator of $\boldsymbol{\beta}^{k*}_{u_t}$ after t trials. We use logistic regression [Li *et al.*, 2012] and solve the following equation to calculate $\hat{\boldsymbol{\beta}}^k_{u_t}$,

$$\sum_{s=1}^{t-1} (f_k(s, u_t) - g(\langle \mathbf{r}_{a_k(s)}(s)^\intercal \hat{\boldsymbol{\beta}}^k_{u_t} \rangle) \mathbf{r}_{a_k(s)}(s) = 0 \tag{7}$$

Since at the beginning of trial t we do not have access to the multi-objective rewards to calculate the expected utility reward, we rely on our estimated multi-objective rewards $\{\hat{\mathbf{r}}_a(t) | \hat{\mathbf{r}}_a(t) = [\hat{r}^1_a(t), \ldots, \hat{r}^I_a(t)], a \in \mathcal{A}\}$ obtained from the previous step. Suppose in trial t we provide each multi-armed bandit instance the set of estimated multiple objective rewards. Then we calculate the expected utility reward for each arm $a \in \mathcal{A}$ as $g(\hat{\mathbf{r}}_a(t)^\intercal \hat{\boldsymbol{\beta}}^k_{u_t})$.

As explained in the problem setting, we can observe the utility rewards only for a subset of arms in a given trial. To account for the uncertainty, we relied on a upper confidence bound based exploration-exploitation technique to select the best arm for each rank. Each M_k instance chooses the action that maximizes the upper confidence bound of the empirical mean reward $g(\hat{\mathbf{r}}_a(t)^\intercal \hat{\boldsymbol{\beta}}^k_{u_t})$. More formally, let $\mathbf{V}_u(t) = \sum_{s=0}^{t} \mathbf{r}_a(s)(\mathbf{r}_a(s))^\intercal$ and γ be the exploration parameter. Then we modeled the ranking strategy for each M_k instance as,

$$a_k(t) \leftarrow \underset{a \in \mathcal{A} \cap \tilde{\mathcal{A}}(t, k-1)^c}{\text{argmax}} \left(g(\hat{\mathbf{r}}_a(t)^\intercal \hat{\boldsymbol{\beta}}^k_{u_t}) + \gamma ||\hat{\mathbf{r}}_a(t)||_{\mathbf{V}^{-1}_{u_t}(t)} \right) \tag{8}$$

where $||\hat{\mathbf{r}}_a(t)||_{\mathbf{V}^{-1}_{u_t}(t)}$ is the weighted l_2 norm of the multi-objective reward vector associated with the matrix $\mathbf{V}_u(t)$. The significance of this model is that each multi-armed bandit instance running in rank k can learn the users preferred trade-off over the multiple objectives for rank k. Furthermore, it can learn the users' preferences over the multiple objectives in different contexts as we have modeled the contexts with each objective reward.

3.2 Online Parameter Updates

As explained above MOU-UCB generates the ranked list $\tilde{\mathcal{A}}(t, K)$ based on arms selected by each M_k instance and displays it to the user. On trial t, a user u_t considers the ranked list $\tilde{\mathcal{A}}(t, K)$ in order and selects relevant arms and reveals multiple objective rewards for some arms. Each M_k uses the observed utility reward (selected or not) in rank k and the rewards in multiple objectives to update its estimated parameters $\hat{\boldsymbol{\beta}}^k_{u_t}$, $\hat{\boldsymbol{\theta}}_{i,a}$ and $\hat{\mathbf{r}}_a(t)$. To elaborate, let a^*_k be a selected resource in trial t at rank k and $\mathbf{r}_{a^*_k}(t)$ be the multiple objective rewards for a^*_k. Based on $\mathbf{r}_{a^*_k}(t)$ we update $\hat{\boldsymbol{\theta}}_{i,a^*_k}$ to minimize the objective reward estimation error given in Equation 5. Based on the observed utility reward, we update the maximum likelihood estimator $\hat{\boldsymbol{\beta}}_{u_t}^k$ in each M_k as given in Equation 7. As t increases we can learn the users utility function over the multiple objective rewards for each rank simultaneously.

4 Experiments

We evaluate the effectiveness of our model in generating relevant ranked lists using data from TripAdvisor [Liu *et al.*, 2011; Wang *et al.*, 2010; Wang *et al.*, 2011].

TripAdvisor dataset. The dataset contains a list of users who have rated hotels, giving an overall evaluation, as well as ratings along dimensions like value and location. Their overall rating for the hotel is a function of these multi-objective ratings as can be seen in Figure 2. The dataset also contains user profiles, hotels profiles, hotel reviews, date, location, number of readers for reviews, number of helpful votes for reviews and overall rating for a hotel and explicit ratings for 5 different objectives. The objectives are value aspect rating, rooms aspect rating, location aspect rating, cleanliness aspect rating, and service aspect rating. Aspect ratings range from 0 to 5 stars.

Data analysis. First, we analyzed the above dataset to justify our claim that rewards on different objectives determine the overall rating users give for a hotel. We clustered [Liu *et al.*, 2011] users with similar preferences over the multiple objective rewards and analyzed the correlation between their rating for the preferred objective and the overall rating. We present the correlation between the multi-objective rewards and the overall rating for two user clusters in Figure 2. The preferred objective for one cluster was ‘hotel services’ and for the other cluster, it was ‘value for money’. As can be seen, in each cluster we can see the overall rating has a high correlation with the respective preferred objective. In our experiments, we demonstrate how this user behaviour is effectively captured by MOU-UCB to generate ranked lists that maximize users clicks considering multiple objective rewards.

4.1 Experimental Setup

In our setup, the goal is to generate a ranked list of hotels for different users in different contexts considering their preferences for different aspects in hotels. We used our algorithm to present a ranked list of potential hotels to the users. User feedback for this list was captured over a number of trials. If the ranked list contained a hotel rated higher than 2.5 by the user as per the ground truth, it was considered as a user click.

To create an experimental dataset with adequate user ratings for hotels we filtered out users with less than 10 hotel ratings and hotels with less than 2 ratings. The resultant dataset comprised of 777 users and 1875 hotels resulting in a total of 11552 ratings. To evaluate our algorithm, we compared it to several other baseline algorithms for ranking. These are contextual based (Ranked-LinUCB), multi-objective based (Ranked-PUCB1) and non-contextual single objective based (Ranked-epsilon-greedy) algorithms:

- **Ranked-LinUCB**: LinUCB [Li *et al.*, 2010] selects a single item that maximizes the linear contextual payoff without considering multiple objective rewards. We used LinUCB instances in each rank to generate the ranked list.

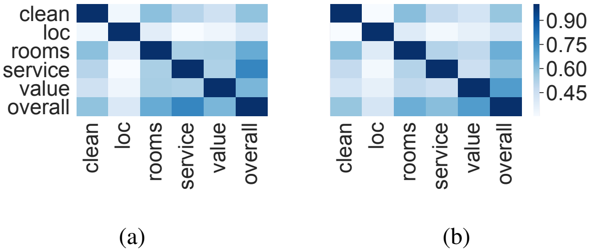

Figure 2: Evidence for the correlation between the overall rating and a specific objective rating for a hotel for different users. Users in (a) provide the overall rating based on the services of the hotel where as users in (b) provide the overall rating based on value for money.

- **Ranked-PUCB1**: PUCB1 [Drugan and Nowé, 2013] selects a single item uniformly from the Pareto-front, which incorporates multiple rewards, but does not try to learn user preferences over these. We used PUCB1 instances in each rank to generate the ranked list.
- **Ranked-ϵ-greedy**: Epsilon-greedy [Watkins, 1989] selects an item randomly with ϵ probability or the highest payoff item with $1 - \epsilon$ probability. We used Epsilon-greedy instances in each rank to generate the ranked list.

We implemented the above baselines by using the existing LinUCB, PUCB1 and ϵ-greedy algorithms as the base algorithm in a Ranked Bandits [Radlinski *et al.*, 2008] algorithm.

We can treat the MOCR-B problem as a sequential decision making problem. Therefore, to evaluate our algorithm we measured the click-through-rate (CTR) over the trials as follows: CTR $= \frac{1}{T}\sum_{t=1}^{T}\frac{1}{K}\sum_{k=1}^{K}\mathbb{I}_{\{f_k(t)=1\}}$. The CTR is interpreted as the average ratio of number of clicks received over the T trials for a K sized ranked list. We also calculated three other measures, precision@k (P@k), recall@k(R@k) and mean reciprocal rank (MRR) [Liu, 2009] to evaluate the distribution of the clicks across the ranked list. When T is the total number of trials and k_t is the rank of the first clicked resource in trial t, then MRR was measured as follows: MRR $= \frac{1}{T}\sum_{t=1}^{T}\frac{1}{k_t}$. MRR helps us to measure whether our solution can rank relevant items higher compared to the baselines. P@k represents the accuracy of the model in retrieving the correct document at the k^{th} rank and it was measured as follows: P@$k = \frac{\text{\# of relevant items @}k}{\text{\# of recommended items @}k}$. R@$k$ is the proportion of relevant items found in the k^{th} rank and it was measured as follows: R@$k = \frac{\text{\# of relevant items @}k}{\text{total \# of relevant items}}$

Model parameters. We chose the exploration parameters that led to the highest click through rates in each baseline algorithm: (1) For MOU-UCB α and γ were set to 0.2 and 0.3 respectively, (2) For ranked-ϵ-greedy ϵ was set to 0.5. The size of the ranked list K was set to 10, although the final results were not highly sensitive to changes in K. We set the number of trials T to $10,000$.

4.2 Feature Construction

As we mentioned before, we incorporate the context features, the user features and the hotel features to make an accurate

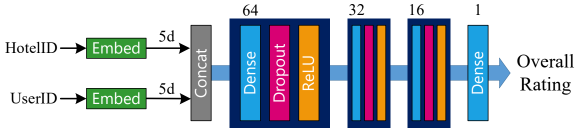

Figure 3: The neural network embedding structure with hidden unit size indicated for each dense layer.

prediction. To specify which hotel is chosen by a certain user, we also need to consider the hotelID and userID as features. However, the vocabulary size of the categorical values can be very large (e.g., there are 1875 hotels and 777 users in our dataset), and thus it is computationally expensive.

To address this issue, inspired by the success of embedding methods in different research areas [Mikolov *et al.*, 2013], we also employ such method to encode the hotelIDs and the userIDs. Compared to one-hot encoding [Gal and Ghahramani, 2016], the embedding method has two main advantages. First, the embedding method effectively reduces the input dimension, which makes our algorithm more computationally efficient. Moreover, it has been shown that the categorical values with similar semantic meaning are usually embedded to close locations [Gal and Ghahramani, 2016]. Hence, the embedding method helps to find and share similar patterns among different hotels as well as users.

In this study, a supervised learning framework is utilized to obtain the embedding vectors. We use userID and hotelID as inputs to predict the *overall rating* of each transaction. More specifically, as depicted in Figure 3, we embed userID and hotelID to $\mathbb{R}^5$ respectively. The embedding vector is randomly initialized but will be refined during the training phase. Once we get their embedding vectors, we simply concatenate them as the input of a multi-layer perceptron to predict the overall rating. During the experiments, we adopt Adam [Kingma and Ba, 2014] optimizer to train the parameters. The learning rate of Adam is $5e^{-4}$ and the batch size during training is 32. Our model is implemented with PyTorch 1.0 on the GPU server with one Titan V.

We concatenated the learned embedding vectors with numerical context features to obtain the final context feature vector for each objective. For example, when modeling the location aspect rating for a user and hotel, we concatenated the users embedding vector, hotels embedding vector, users context vector and the hotels context vector. The dimension of the final context feature vector was 26.

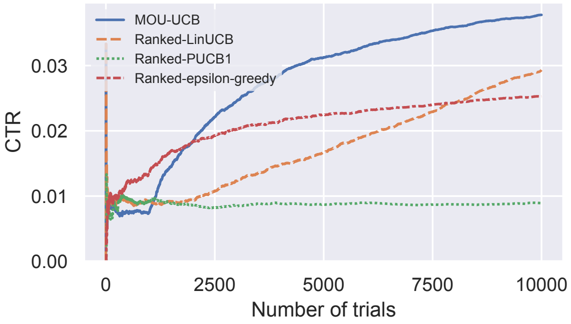

Figure 4: CTR for the ranked lists presented to TripAdvisor users

Method	CTR	MRR	P@1	P@2	R@1	R@2
MOU-UCB	0.038	0.123	0.060	0.060	0.049	0.049
Ranked-PUCB1	0.009	0.025	0.009	0.008	0.008	0.006
Ranked-LinUCB	0.029	0.069	0.030	0.033	0.020	0.021
Ranked-ϵ-greedy	0.025	0.064	0.019	0.028	0.015	0.025

Table 1: CTR, MRR, P@k and R@k values for the TripAdvisor dataset.

4.3 Discussion

The effectiveness of our approach is validated by comparing the CTR, MRR, P@k and R@k measures against other baselines. As seen in Figure 4, the CTR curve for each baseline converges over 10000 trials. As the number of trials increases, MOU-UCB reaches a higher CTR compared to the baselines. This indicates that users find suitable hotels more often in the ranked list generated by MOU-UCB compared to other algorithms. The relatively high CTR for MOU-UCB compared to Ranked-LinUCB suggests that there is value addition in optimizing the utility reward based on multiple objective rewards for online ranking instead of optimizing a single objective reward. The improved results for MOU-UCB can be attributed to its ability to balance the diverse needs of users when choosing hotels to stay in different contexts.

Next, we discuss the quality of the ranked lists. Figure 5 plots the number of clicks in the first 10 ranks after 10000 trials. MOU-UCB has significantly more clicks in rank 1 compared to other ranks. On the contrary, the clicks are distributed more uniformly across the ranks in other baseline methods. These dissimilarities suggest that by modeling context information in multiple objectives and considering user preferences over the multiple objectives MOU-UCB has been able to retrieve relevant items in top ranks whereas baseline methods retrieve items across all the ranking positions. Additionally, when investigating the P@1 and MRR values shown in Table 1 we see a significant difference in the P@1 values for MOU-UCB in comparison to other benchmark algorithms. Likewise, the MRR value for MOU-UCB is significantly higher than the baselines. The results confirm

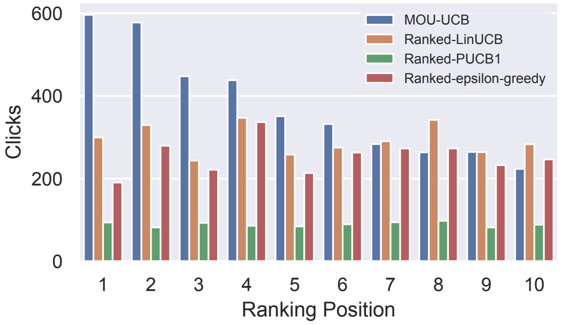

Figure 5: User-clicks for each rank in the TripAdvisor dataset.

that MOU-UCB accurately retrieves the relevant hotels in the top ranks compared to other approaches.

We analyzed the regret of our algorithm based on the assumptions and conditions presented by Li et al. [Li *et al.*, 2017] for generalized contextual bandits. Let $\rho(t)$ be the maximum instantaneous estimation error of the multi-objective reward in trial t, where $\rho(t) = \max_{i\in[I],a\in\mathcal{A}} |r_a^i(t) - \hat{r}_a^i(t)|$. Based on existing studies [Lattimore and Szepesvári, 2018] we can show that $\rho(t)$ is bounded with high probability. When the instantaneous error in estimating multi-objective rewards is bounded, we can upper bound the regret of a multi-armed bandit instance running in a specific rank as $\mathcal{O}\Big(I\sqrt{T}\log(\frac{TI[1+\rho(t)]^2}{I\xi})\Big)$ with probability at least $1-\xi$ where $\xi \in [0,1]$. Therefore, we can show that the regret is bounded for all K multi-armed bandit instances running in our algorithm.

5 Related Work

We relate our work to similar bandit approaches that have focused on context-awareness, multi-objective optimization, and ranking. Existing contextual bandit approaches such as LinUCB [Li *et al.*, 2010] are robust against cold start problems. Despite successful adaptations of linear contextual bandits [Li *et al.*, 2012; Wanigasekara *et al.*, 2016] to recommend single arms, they have not been used for ranking or multi-objective optimization. PUCB1 [Drugan and Nowé, 2014], and Pareto Thompson sampling [Yahyaa and Manderick, 2015] algorithms focus on multi-objective optimization where similar to our problem setting it is assumed that a reward vector is observed for selected arms instead of a scalar reward as in a classical Multi-Armed Bandit (MAB) setting. PUCB1 adapts an upper-confidence bound approach to explore and exploit. Auer at el. [Auer *et al.*, 2016] have presented two similar algorithms to return all Pareto optimal points in a stochastic bandit feedback setting based on an elimination algorithm and upper confidence bounds. However, these multi-objective multi-armed bandit approaches do not model context information. Additionally, the above methods are used to identify a Pareto-front, but in many practical problems, we need to order the Pareto-front based on a user/domain specific preference. Our algorithm learns the preferences over the multiple objectives over time and ranks the items to match the users utility better.

Ranked Bandits algorithm [Radlinski *et al.*, 2008] is a bandit approach for ranking. In the ranked bandit problem setting, the user sees a list of K items. The user examines the recommended list from the first item to the last and selects the first relevant item. The problem is to generate an optimal list of K items that maximizes the probability that a user finds a relevant item in the recommended list. The key characteristic of ranked bandits algorithm is that each position in the recommended list is an independent bandit problem, which is solved by some base bandit algorithm. We adopted the notion of using a MAB instance in each rank k based on the Ranked Bandits [Radlinski *et al.*, 2008]. The value addition in our setting is that, using our algorithm, we show how the ranked bandits algorithm can be used to determine the user preferences over multiple objectives to populate a ranked list.

A few studies have begun to look into the fusion of contextual bandits, multi-objective multi-arm bandits and ranked bandits. Tekin at el. [Tekin and Turgay, 2017] defined a multi-objective contextual bandit problem for two objectives in which they have explicitly defined one objective as the dominant, and the other as the non-dominant. The MOC-MAB algorithm presented in their work maximizes the long-term reward of the non-dominant objective conditioned on the fact that it maximizes the long-term reward of the dominant objective. They use similarity information in the context space to recommend a single resource whose reward vector is optimized for multiple objectives. This work does not focus on ranking resources. In contrast, our work does not assume that there is a dominant objective. Instead, we learn the user's preferences over the multi-objective space. Lacerda et al. [Lacerda, 2015] combined multi-objective optimization with ranked bandits for recommendation systems. The study uses scalarization functions to find the Pareto-front. Otunba at el. [Otunba *et al.*, 2017] proposed a multi-objective pairwise ranking model that provides item recommendation and audience retrieval simultaneously using a scalarized approach for multi-objective optimization. An existing work [Roijers *et al.*, 2017] that studies a problem similar to the MOCR-B problem uses a utility based view for multi-objective optimization in a bandit setting. However, the algorithms Utility-MAP UCB and Interactive Thompson Sampling [Roijers *et al.*, 2017] presented in their work do not discuss returning a ranked list. Additionally, there is an expensive pairwise reward vector comparison to learn user preferences. In our algorithm, we do not require pairwise comparisons as the users' preferences are learned independently in an online bandit setting. In summary, the value added by our work is the robust fusion of context-awareness, multi-objective optimization and ranking techniques in a bandit setting.

6 Conclusion

There are important real-world online settings that require algorithms to provide users with ranked lists of relevant web resources based on rewards in multiple objective rewards. Our work provides insight into how a wide range of future work can integrate ranking, sequential-decision making, contextual information, and multiple objective rewards for online ranking. As future work we will validate our algorithm using other multi-criteria datasets [Tallapally *et al.*, 2018] and present a more detail regret analysis.

Acknowledgments

This work is supported by the National University of Singapore and A*STAR - SERC PSF Grant 1521200084.

References

[Auer *et al.*, 2016] Peter Auer, Chao-Kai Chiang, Ronald Ortner, and Madalina Drugan. Pareto front identification from stochastic bandit feedback. In *Artificial Intelligence and Statistics*, pages 939–947, 2016.

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

[Drugan and Nowé, 2014] Madalina M Drugan and Ann Nowé. Scalarization based pareto optimal set of arms identification algorithms. In *Neural Networks (IJCNN), 2014 International Joint Conference on*, pages 2690–2697. IEEE, 2014.

[Gal and Ghahramani, 2016] Yarin Gal and Zoubin Ghahramani. A theoretically grounded application of dropout in recurrent neural networks. In *Advances in neural information processing systems*, pages 1019–1027, 2016.

[Kingma and Ba, 2014] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[Lacerda, 2015] Anisio Lacerda. Contextual bandits for multi-objective recommender systems. In *Intelligent Systems (BRACIS), 2015 Brazilian Conference on*, pages 68–73. IEEE, 2015.

[Lattimore and Szepesvári, 2018] Tor Lattimore and Csaba Szepesvári. *Bandit Algorithms*. Cambridge University Press, 2018. preprint available at https://tor-lattimore.com/downloads/book/book.pdf.

[Li *et al.*, 2010] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 661–670. ACM, 2010.

[Li *et al.*, 2012] Lihong Li, Wei Chu, John Langford, Taesup Moon, and Xuanhui Wang. An unbiased offline evaluation of contextual bandit algorithms with generalized linear models. In *Proceedings of the Workshop on On-line Trading of Exploration and Exploitation 2*, pages 19–36, 2012.

[Li *et al.*, 2017] Lihong Li, Yu Lu, and Dengyong Zhou. Provably optimal algorithms for generalized linear contextual bandits. *arXiv preprint arXiv:1703.00048*, 2017.

[Liu *et al.*, 2011] Liwei Liu, Nikolay Mehandjiev, and Dong-Ling Xu. Multi-criteria service recommendation based on user criteria preferences. In *Proceedings of the fifth ACM conference on Recommender systems*, pages 77–84. ACM, 2011.

[Liu, 2009] Tie-Yan Liu. Learning to rank for information retrieval. *Found. Trends Inf. Retr.*, 3(3):225–331, March 2009.

[Mikolov *et al.*, 2013] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[Otunba *et al.*, 2017] Rasaq Otunba, Raimi A Rufai, and Jessica Lin. Mpr: Multi-objective pairwise ranking. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*, pages 170–178. ACM, 2017.

[Radlinski *et al.*, 2008] Filip Radlinski, Robert Kleinberg, and Thorsten Joachims. Learning diverse rankings with multi-armed bandits. In *Proceedings of the 25th international conference on Machine learning*, pages 784–791. ACM, 2008.

[Robbins, 1985] Herbert Robbins. Some aspects of the sequential design of experiments. In *Herbert Robbins Selected Papers*, pages 169–177. Springer, 1985.

[Roijers *et al.*, 2017] Diederik M Roijers, Luisa M Zintgraf, and Ann Nowé. Interactive thompson sampling for multi-objective multi-armed bandits. In *International Conference on Algorithmic DecisionTheory*, pages 18–34. Springer, 2017.

[Tallapally *et al.*, 2018] Dharahas Tallapally, Rama Syamala Sreepada, Bidyut Kr Patra, and Korra Sathya Babu. User preference learning in multi-criteria recommendations using stacked auto encoders. In *Proceedings of the 12th ACM Conference on Recommender Systems*, pages 475–479. ACM, 2018.

[Tekin and Turgay, 2017] Cem Tekin and Eralp Turgay. Multi-objective contextual multi-armed bandit problem with a dominant objective. *arXiv preprint arXiv:1708.05655*, 2017.

[Wang *et al.*, 2010] Hongning Wang, Yue Lu, and Chengxiang Zhai. Latent aspect rating analysis on review text data: a rating regression approach. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 783–792. ACM, 2010.

[Wang *et al.*, 2011] Hongning Wang, Yue Lu, and ChengXiang Zhai. Latent aspect rating analysis without aspect keyword supervision. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 618–626. ACM, 2011.

[Wanigasekara *et al.*, 2016] Nirandika Wanigasekara, Jenny Schmalfuss, Darren Carlson, and David S Rosenblum. A bandit approach for intelligent iot service composition across heterogeneous smart spaces. In *Proceedings of the 6th International Conference on the Internet of Things*, pages 121–129. ACM, 2016.

[Watkins, 1989] Christopher John Cornish Hellaby Watkins. *Learning from delayed rewards*. PhD thesis, King's College, Cambridge, 1989.

[Xiang *et al.*, 2010] Biao Xiang, Daxin Jiang, Jian Pei, Xiaohui Sun, Enhong Chen, and Hang Li. Context-aware ranking in web search. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 451–458. ACM, 2010.

[Yahyaa and Manderick, 2015] Saba Yahyaa and Bernard Manderick. Thompson sampling for multi-objective multi-armed bandits problem. In *Proceedings*, page 47. Presses universitaires de Louvain, 2015.

[Zamani *et al.*, 2017] Hamed Zamani, Michael Bendersky, Xuanhui Wang, and Mingyang Zhang. Situational context for ranking in personal search. In *Proceedings of the 26th International Conference on World Wide Web*, pages 1531–1540. International World Wide Web Conferences Steering Committee, 2017.