MultiScale Policy Learning for Alignment with Long Term Objectives

Richa Rastogi [*1] **Yuta Saito** [*1] **Thorsten Joachims** [1]

Abstract

AI systems deployed in practical settings (e.g., conversation systems, recommender systems) naturally collect user feedback. Alignment is an important goal of these systems, but it is not clear what objective should be optimized in the first place so that they are aligned with diverse human preferences. Being a higher level objective, alignment is naturally associated with long term outcomes. Importantly, there is a disconnect in the timescale of observed feedback (e.g., collecting click data from rankings in a recommender system) and the downstream effect they strive to achieve (e.g., long-term satisfaction of users on the platform). To achieve alignment with desired long-term objectives, this disconnect at different levels, namely, the lower micro level at which fast-acting feedback is collected, and the upper macro level, concerned with higher-level objectives, needs to be reconciled. We introduce *MultiScale Policy Learning (MSPL)* with nested contextual bandits for policy learning at multiple levels to bridge this disconnect. MSPL uses bi-level optimization to select the shorter-term objective at the next lower scale to optimize the longer-term objective at the next higher scale. The policy for both upper and lower level are learned to optimize for long term goals. As part of ongoing project, we present preliminary results on a recommendation system simulator that shows promising results.

1. Introduction

Designing AI systems that align with human preferences has been a goal of growing importance. However, a fundamental limitation towards this goal lies in the feedback that is typically used for optimizing these systems. In particular, training data is primarily derived from short-term signals like clicks and views in recommender systems, or the performance of the next-sentence prediction in conversational systems. Though this naturally occurring feedback is abundant, it is noisy and biased, and experience with optimizing engagement in social media platforms (Mansoury et al., 2020) has shown that its unmitigated optimization can adversely affect the desired long-term behavior of these systems. On the other hand, optimizing long-term user retention is difficult due to sparse and low frequency of feedback.

While conventional approaches have predominantly ignored the long-term feedback in the optimization objective, some recent work aims to include long-term feedback. Most prominently, reinforcement learning based approaches to this problem propose a decomposition of long-term goals into short-term goals (Maystre et al., 2023; McDonald et al., 2023). However, these approaches consider the effect of short term interventions (e.g., rankings) on long term goals (e.g, user retention) and thus only optimize for short term interventions. In particular, how to take interventions at different timescales to achieve the optimal long term objective is still an open research question.

Our approach is to contextually reconcile the disconnect between different timescales of feedback and interventions in a tractable manner, allowing the steering and control of these systems. This involves learning the policy (and interventions) at multiple scales to steer for desirable outcomes at different levels. At each scale, we optimize the most appropriate metric – from short-term signals (e.g., click) at the lowest scale, to long-term signals (e.g., user retention) at the highest scale. Our approach, which we call MultiScale Policy Learning (MSPL), uses bi-level optimization to select the shorter-term objective at the next lower scale so that we maximize the longer-term objective at the next higher scale. In this way, both long-term and short-term interventions are optimized for achieving long term outcomes.

In the following, we discuss three practical examples that illustrate the impact and need for this approach.

Example 1. Consider a recommender system for media streaming for children, where the platform is interested in

[*]Equal contribution [1]Department of Computer Science, Cornell University, Ithaca, NY, USA. Correspondence to: Richa Rastogi <rr568@cornell.edu>, Yuta Saito <ys552@cornell.edu>, Thorsten Joachims <tj@cs.cornell.edu>.

Proceedings of the 41^{st} International Conference on Machine Learning (ICML) 2024 Workshop on Models of Human Feedback for AI Alignment, Vienna, Austria. PMLR 235, 2024.

the long-term outcomes such as user retention or subscription renewal. For simplicity, let's consider two levels – the upper level concerned with user retention and the lower level concerned with short-term engagement. Depending on the context (user profiles), the platform can take upper-level interventions such as boosting specific item groups (e.g., documentaries) for ranking. The feedback from this intervention – whether a user renews the subscription or not – is observed after a month. The lower-level interventions are rankings, and they are affected by the higher-level intervention throughout the month. Clicks are clearly an important signal about preferences readily available at the lower level for learning, but unmitigated maximization of clicks is not necessarily aligned with customer retention. Arguably, an upper-level intervention (e.g. boosting documentaries) can optimize customer retention even if it leads to fewer clicks. The goal of our multi-scale learning approach is to optimize interventions at both levels to improve long-term outcomes.

Example 2. In the next example, we consider a question generation platform to assist in education (Elkins et al., 2023). Optimizing for the next question at the lowest level as shown in Figure 1 leads to the maximum number of questions answered (feedback at that level) and keeps the students' attention for the short-term. However, this could adversely affect the ultimate goal of achieving long-term learning outcomes, e.g., developing a deep understanding of the *subject*. In that sense, we wish to optimize the lower levels as long as the highest level is optimal. Crucially, the feedback at each level is observed at different resolution and each of the levels form a hierarchy of objectives affecting the level below it. At the level of *topic*, depending on the context, the platform can intervene to control the difficulty or hardness of questions, affecting the lower level next *question* generation. At the highest level of *subject*, the intervention relates to when to move to the next topic based on understanding of the current topic at the lower level.

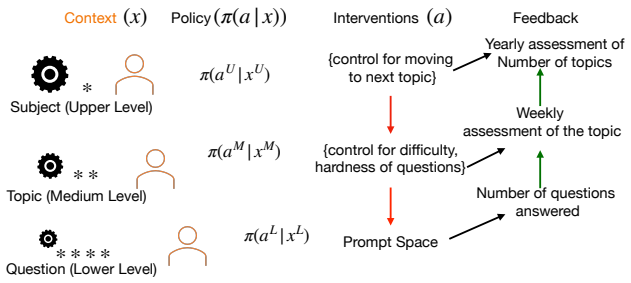

Figure 1. MultiScale resolution of feedback and hierarchy of objectives on an education platform

Example 3. In the last example, we consider a movie streaming app with the long term goal of subscription renewal of the customers. In general, these systems use the same objective function for every user but different users can have widely different preferences in terms of accuracy versus diversity versus novelty of a movie recommendation, etc. As a result, optimizing for the same objective function for every user cannot capture the diverse user preferences. If instead, an upper level policy selected interventions to weight diversity, novelty etc. differently for different users, then this control from the upper level to the lower level can align the lower level policy with the desired user preferences, leading to long term satisfaction. The accuracy and diversity of movie recommendations relate to short term feedback based on the policy of presenting movies while alignment to the user preferences relates to long term feedback.

Our key insight is to elevate the different timescale feedback and the corresponding interventions to multiple levels by assuming the knowledge of this structure as prior information.

2. A BiLevel Optimization Framework

Our approach to reconciling this disconnect between interventions and feedback from multiple scales is to view the long-term interventions as meta-parameters to the short-term interventions, similar to the design of controllers for physical processes. From this perspective, we want to design systems that can learn these meta-parameters contextually at multiple hierarchical levels. With this viewpoint, we propose a novel formulation of the problem as a bi-level optimization (Colson et al., 2007) of the policies at each level. The lower level (LL) subproblem in Figure 2 represents the higher frequency task, e.g. retrieval augmented generation parameterized by meta parameters from the upper level (UL) subproblem. The upper level involves the optimization of these meta parameters via long term objective that encodes the user intent with feedback observed at a lower frequency.

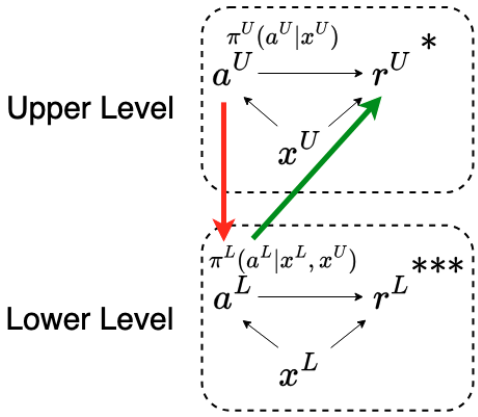

Figure 2. Overview of BiLevel Framework

In a single-level setting, let $x \in \mathcal{X}$ denote the context vector, drawn i.i.d. from an unknown distribution $p(x)$. A possibly stochastic policy $\pi(a|x)$ chooses action $a \in \mathcal{A}$. The reward $r \in [0, r_{max}]$ is observed from an unknown distribution

$p(r|x,a)$. The value of policy is denoted as

$$V(\pi) = \mathbb{E}_{p(x)\pi(a|x)p(r|x,a)}[r] = \mathbb{E}_{p(x)\pi(a|x)}[q(x,a)]$$

where $q(x,a) = \mathbb{E}_{p(r|x,a)}[r]$ is the expected reward function.

Figure 2 illustrates an overview of our bi-level framework for two levels. Extending the above formulation, for the upper level, we have a stochastic policy $\pi^U(a^U|x^U)$ and the observed reward r^U is drawn from an unknown distribution $p(r^U|x^U, a^U, \pi^L)$. An expected reward function $q^U(x^U, a^U, \pi^L)$ is similarly defined, where π^L is the lower level policy. The reward r^U for action a^U is observed after T timesteps of the lower level, where $t = 1, \dots T$ represents the timescale for lower level feedback. The UL reward r^U represents the long term outcomes of interest such as subscription renewal or return time for a recommender system. The value of the UL policy π^U, is defined as

$$V^U(\pi^U) = \mathbb{E}_{p(x^U)\pi^U(a^U|x^U)}[q^U(x^U, a^U, \pi^L)] \tag{1}$$

In this way, the LL policy affects the long term objective at the UL. This is shown by the connection from LL to UL (green arrow) in Figure 2.

We now define the lower level and associated notation. For LL, the context x^L is drawn from an unknown distribution $p(x^L|x^U)$. A possibly stochastic policy $\pi^L(a^L|x^L, x^U)$ chooses action a^L and reward r^L is observed from an unknown distribution $p(r^L|x^L, x^U, a^L)$. An expected reward function $q^L(x^L, x^U, a^L)$ is similarly defined.

To describe the mechanism by which UL can pass meta parameters to select the right objective function at the LL, consider the red arrow in the bi-level structure of Figure 2. We specify this parameterization with action a^U from UL to LL via a predefined function $f(q^L(.), a^U)$. This function $f(.)$ takes two input parameters, the first being the expected reward function of the LL denoted by $q^L(x^L, x^U, a^L)$ and the second, $a^U \sim \pi^U$. For example, consider a recommendation platform with short term outcomes consisting of accuracy and diversity of movies presented to the users. Each of these short term outcomes depends on lower level intervention a^L (ranking of movies) as well as context from lower and upper level (movie context and user context respectively), so that informally, $q^L(x^L, x^U, a^L) = \text{acc}(x^L, x^U, a^L) + \text{div}(x^L, x^U, a^L)$. This means that the same objective of equally weighted accuracy and diversity of movies is used for all users. In the simplest case, the parameterization $f(.)$ that we propose can be of the following form $f(q^L(.), a^U) = a^U$ acc(.)$+(1 - a^U)$ div(.). The intervention $a^U \sim \pi^U(.|x^U)$ is the weighting for diverse vs accurate movie recommendations based on the user context x^U, and provides a mechanism to select the right objective at the lower level. While this is a simple example, more complex and realistic functional forms of $f(.)$ depending on the application can be used. The value of LL policy π^L is defined as

$$V^L(\pi^L) = \mathbb{E}_{p(x^L,x^U)\pi^L(.)\underbrace{\pi^U(a^U|.)}_{\text{UL}}}[f(q^L(.), a^U)] \tag{2}$$

The overall data generation for the bi-level framework is shown in process 1. The LL operating within the nested loop consists of short term feedback $r_t^L \sim p(r^L|x_t^L, x_{t'}^U, a_t^L)$ observed at the faster timescale t. This feedback not only depends on the LL context x_t^L and action a_t^L at the same timescale but also on the UL context $x_{t'}^U$ observed at the timescale t' of the UL. Finally, the observed UL reward $r_{t'}^U$ depends on the particular π^L operating at LL.

Data Generation Process 1

foreach $t' \in \{T, 2T, \dots mT\}$ **do**
  $x_{t'}^U \sim p(x^U)$
  $a_{t'}^U \sim \pi^U(.|x_{t'}^U)$
  foreach $t = 1, \dots T$ **do**
    $x_t^L \sim p(x^L|x_{t'}^U)$
    $a_t^L \sim \pi^L(.|x_t^L, x_{t'}^U)$
    Observe $r_t^L \sim p(r^L|x_t^L, x_{t'}^U, a_t^L)$
  Observe $r_{t'}^U \sim p(r^U|x_{t'}^U, a_{t'}^U, \pi^L)$

Our overall goal is to optimize for the long term objective of UL, given by Eq (1), subject to the nested subproblem of optimizing the short term objective of LL, given by Eq (2). We now formally describe the bi-level optimization problem. In general, to learn policy parameters θ (e.g. neural network), we have $\pi^* \leftarrow \arg\max_\theta V(\pi_\theta)$. To learn policy parameters ψ, ϕ for policy π^U, π^L respectively, we have,

$$\begin{aligned} & & \text{(UL Problem)} & \\ \pi^{U*} & \leftarrow & \arg\max_{\psi} V^U(\pi_\psi^U; \pi_\phi^{L*}) & \quad \text{subject to} \\ & & \text{(LL Problem)} & \\ \pi^{L*} & \leftarrow & \arg\max_{\phi} V^L(\pi_\phi^L; \pi_\psi^U) & \end{aligned}$$

In this way, the shorter term policy π^{L*} is selected such that it optimizes the long term objective $V^U(\pi^U)$. By formulating the bi-level optimization as above, we frame the overall problem as a nested contextual bandit problem. We now discuss how to solve this nested problem based on historical logged datasets in a tractable manner.

3. MultiScale Policy Learning

To learn the UL and LL policies, we turn to offline policy learning from logged bandit data collected from a suboptimal logging policy at each level. Consider a naive approach where logged data is collected for each level and an off-policy estimator is used to learn LL and UL policies.

This approach fails for the UL policy learning, since the logged data contains the bias from both the logged UL and the logged LL policy. Our first approach describes a solution to this issue by sequentially learning the LL policy, collecting logged bandit data for UL, followed by learning the UL policy. We then propose a solution with a second approach, where logged bandit data for UL and LL is pre-collected and we use predicted rewards for UL policy learning. Both methods can be extended to learning policies at multiple scales and are encompassed by our unifying framework of MultiScale Policy Learning.

3.1. Nested Contextual Bandits with Logging Policies

Consider logging policies $\pi^U_{\psi_0}, \pi^L_{\phi_0}$ for upper and lower level respectively. Algorithm 1 describes offline nested policy learning, one each for UL and LL. It consists of a procedure PolicyLearning that requires input as the logging policies $\pi^U_{\psi_0}, \pi^L_{\phi_0}$. The logged bandit data D^L consists of context, action, and reward at LL $\sim \pi^L_{\phi_0}$ and the additional context x^U from the UL. The UL action $a^U \sim \pi^U_{\psi_0}(.|x^U)$ is passed down to the nested optimization problem of learning the LL policy in line 4 of the algorithm. In particular, we use a model-based estimator (Beygelzimer & Langford, 2009) with a regression model $\hat{q}^L$ estimated using logged data D^L as follows, $\hat{q}^L(.) = \arg\min_{q^{L'}} \sum_{(x^L,x^U,a^L,r^L)\in D^L}(r^L - q^{L'}(x^L, x^U, a^L))^2$. The learned policy π^{L*}_ϕ is defined by selecting the action such that $a^{L*} \leftarrow \arg\max_{a^L \in \mathcal{A}^L} \mathbb{E}[f(\hat{q}^L(., a^L), a^U)]$.

Algorithm 1 Offline Nested Bandit Learning

1: **Procedure** PolicyLearning($\pi^U_{\psi_0}, \pi^L_{\phi_0}$)
2: $D^L := \{(x^L, x^U, a^L, r^L)\} \sim \pi^L_{\phi_0}$
3: $a^U \sim \pi^U_{\psi_0}$
4: $\pi^{L*}_\phi \leftarrow \arg\max_\phi V^L(\pi^L_\phi; a^U, D^L)$
5: $D^U := \{(x^U, a^U, r^U)\} \sim \pi^U_{\psi_0}, \pi^{L*}_\phi$
6: $\pi^{U*}_\psi \leftarrow \arg\max_\psi V^U(\pi^U_\psi; D^U)$
7: **return** $\pi^{U*}_\psi, \pi^{L*}_\phi$
8: **end Procedure**

With the learned policy π^{L*}_ϕ, Algorithm 1 describes collecting logged bandit data D^U for UL with $x^U, a^U \sim \pi^U_{\psi_0}$, while the logged reward r^U depends on both $\pi^U_{\psi_0}$ and π^{L*}_ϕ. This step of collecting logged data D^U after the LL policy is optimized corrects for the bias due to the logging LL policy. Now, a model free approach, such as an Inverse propensity score (IPS) weighting estimator (Horvitz & Thompson, 1952) can correct for the bias in the logging UL policy. The gradient of policy value $\nabla_\psi V^U(\pi_\psi)$ would be estimated as

$$\frac{1}{n}\sum_{i=1}^{n}\left(\frac{\pi_\psi(a^U_i|x^U_i)}{\pi_{\psi_0}(a^U_i|x^U_i)}\right) r^U_i \nabla_\psi \log \pi_\psi(a^U_i|x^U_i)$$

and gradient steps with learning rate η as $\psi \leftarrow \psi + \eta\nabla_\psi V^U(\pi^U_\psi)$ are taken for off policy learning for the upper level. In this way, Algorithm 1 progressively learns the policy starting from the lowest level, collecting log data for the next higher level, and learning the higher level policy.

More generally, Algorithm 1 can be used to learn multiple nested levels by recursively calling the PolicyLearning procedure at line 4 if a lower level exists. Consider a multiscale learning problem with logging policies $\pi^U_{\psi_0}, \pi^M_{\theta_0}, \pi^L_{\phi_0}$ respectively for each of upper, middle, and lower levels. Starting from level U, PolicyLearning($\pi^U_{\psi_0}, \pi^M_{\theta_0}$) is called. Since a level L lower than level M exists, PolicyLearning($\pi^M_{\theta_0}, \pi^L_{\phi_0}$) is called which returns $\pi^{M*}_\theta, \pi^{L*}_\phi$ and the PolicyLearning($\pi^U_{\psi_0}, \pi^M_{\theta_0}$) procedure resumes. The logged data D^U is now collected for the upper level using logging policy $\pi^U_{\psi_0}$ and π^{M*}_θ. Finally, the policy learning step returns the optimal policy π^{U*}_ψ and the algorithm returns $\pi^{U*}_\psi, \pi^{M*}_\theta$. This simple recursive procedure between two levels, solving first for the lower level nested within the upper level forms our MultiScale Policy Learning framework (MSPL).

Note that this approach requires collecting logged bandit data after the policy learning step for the level below it. As the number of levels increase, it would be desirable to be able to use the pre-collected logged data for all the levels, rather than accessing the system after every policy learning step to collect logged data for the next higher level. We now discuss a second approach for a setting where logged bandit data for UL and LL is collected before policy learning.

3.2. Nested Contextual Bandits with Pre-collected Logged Data

Consider logged bandit data $D^U \sim \pi^U_{\psi_0}, D^L \sim \pi^L_{\phi_0}$ consisting of independent observations generated by a logging policy for UL and LL respectively. The key difficulty lies in using the logged rewards r^U for UL policy learning. As discussed earlier, this is because the logged reward r^U consists of the bias from the logged UL policy and a second bias – that of the logged LL policy due to lower level affecting the rewards at upper level. Our solution to this issue is to use predicted rewards $\hat{r}^U$ as a function of the learned LL policy. To quantify the LL performance, we use surrogates that have been commonly adopted in prior literature (Athey et al., 2019) as a proxy of short term policy. The value of UL policy defined earlier in Eq (1) can be written to utilize surrogates from LL as follows.

$$V^U(\pi^U) = \mathbb{E}_{p(x^U)\pi^U(a^U|x^U)}[q^U(x^U, a^U, s^L)]$$

Figure 3 illustrates the bilevel framework with r^U depending on surrogates s^L accumulated over the lower level. The data generation process 1 at the lower level, consequently involves an additional step of accumulating surrogates $s^L \leftarrow$

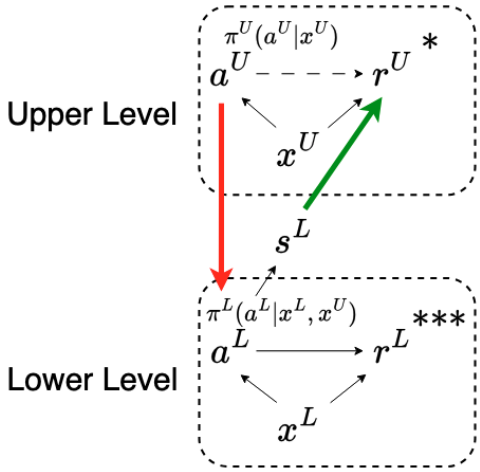

Figure 3. BiLevel Framework with Surrogates

$s^L \cup p(s^L|x_t^L, a_t^L)$ at each time step t of the LL. For the UL, the observed reward $r_{t'}^U \sim p(r^U|x_{t'}^U, a_{t'}^U, s^L)$ for each time step t' of the UL.

Further, the reward prediction $\hat{r}^U$ can be simplified if it only depends on the lower level and the effect of a^U on r^U is completely mediated by the lower level. This assumption is formally described as follows.

Assumption 3.2.1. (Mediation by surrogates) The effect of upper level interventions on upper level reward is completely mediated by the surrogates from the lower level.

This implies no direct effect of the actions at the UL on the reward at UL in the presence of surrogates. In other words, $r^U \perp a^U \mid s^L, x^U$. As a result, the observed reward distribution at the upper level can be decomposed as follows

$$p(r^U \mid x^U, a^U, s^L) = p(r^U \mid x^U, s^L)$$

where $s^L \sim p(s^L|\pi^L)$. This assumption implies that learning a predictive model $\hat{q}^U(x^U, a^U, s^L)$ can instead be replaced by learning $\hat{q}^U(x^U, s^L)$ for the mapping of surrogates to reward r^U (green arrow in Figure 2) . We will explore relaxing this assumption in future work.

We now describe how Algorithm 1 can be used for offline nested policy learning in this setting and note that $D^U := \{(x^U, a^U, r^U, (s^L \sim \pi_{\phi_0}^L))\} \sim \pi_{\psi_0}^U$ can be collected at the same time as D^L. This logged data D^U is used to estimate a predictive model, $\hat{q}^U(.) = \arg\min_{q^{U'}} \sum_{(x^U, s^L, r^U) \in D^U} (r^U - q^{U'}(x^U, s^L))^2$. The model $\hat{q}^U(.)$ predicts upper level rewards $\hat{r}^U$ given surrogates accumulated over LL. The surrogates required as input to $\hat{q}^U(.)$ are in turn obtained according to the logged UL action $a^U \sim D^U$ and using π_ϕ^{L*} from line 4 of the algorithm. With the predictive model for UL rewards, an argmax policy provides the optimal action at the upper level. Thus, a purely model-based approach is used to learn UL policy. In this way, while the previous approach required the collection of logged bandit data D^U after LL policy is learned, line 5 for collecting D^U is not required with this approach. Instead, π_ϕ^{L*} from line 4 is passed directly to the UL policy learning step in line 6 of the algorithm. Concretely, $\pi_\psi^{U*} \leftarrow \arg\max_\psi V^U(\pi_\psi^U; D^U, \pi_\phi^{L*})$. We leave the treatment of more advanced estimators that could be easily applied within the MSPL framework for future work. Similar to the previous section, this approach can also be used to learn multiple nested levels by calling the PolicyLearning procedure recursively.

4. Empirical Evaluation

We conduct experiments on a video recommender system simulator to compare the performance of our nested bandit learning against other commonly used baselines. We simulate the environment according to data generation process 1. For these preliminary experiments, we assume access to reward regression model $\hat{q}^L$ and learn a policy π_ψ at upper level which parameterizes the lower level, according to Algorithm 1. Future experiments will focus on learning the reward models $\hat{q}^L$ as part of policy learning and extending to realistic settings of large number of user groups and actions.

4.1. Baselines

Below, we describe three commonly used baselines for optimizing long term outcomes.

Optimizing for short term feedback (Oracle LL) This baseline selects the action a^L at LL that maximizes the expected reward r^L at that level. There is no action a^U at UL and hence no effect of UL to LL. This baseline demonstrates the gap in long term outcomes due to an unmitigated optimization of short term outcomes.

Random intervention at UL (Random UL) This baseline assumes that π^U is a uniform policy, so action a^U at UL is random. At the lower level π^L selects the action that maximizes the expected LL reward parameterized by a^U.

Optimizing for the majority user group (Opt Majority) This baseline maximizes the expected UL reward for the majority users. As in the previous baseline, π^L is an argmax policy that maximizes the expected LL reward parameterized by a^U.

4.2. Recommender System Simulator Data

We leverage a video recommender system simulator derived from a real-world KuaiRand dataset (Zhao et al., 2023) that supports multi-session environment. We modify the simulator environment to create tradeoff between UL and LL as follows. At the upper level, a user i arrives with

context x^U, consisting of features such as how active they are on the platform, whether they are a live streamer etc. A session is started at UL and consists of up to five requests, after which the session is over and the user exits. The expected return day of the user for the next session serves as the upper level feedback, while clicks per session serve as the lower level feedback.

Since different user groups can have preferences for different video types, we form two user groups, based on a user feature that represents their amount of activity. The majority group is 80%. To control the performance of LL from UL, action a^U of the UL policy provides a boost in ranking scores at the LL, depending on the user and item context. This is the pre-defined parameterization $f(.)$ that affects the lower level policy of selecting the highest ranking score video. The surrogates from LL to UL are defined as the proportion of two video types selected over the five requests. The probability of return day at UL is a function of these surrogates based on user preferences for video groups.

We use offline nested bandit learning with logging policies as described in Section 3.1. In particular, we learn policy parameters ψ for UL as a softmax policy and select the argmax action. We use clipped IPS estimator (Swaminathan & Joachims, 2015) using logged bandit data for UL learning. For computing performance at each level, we sample $n =$ 1,174 users and report the expected return day averaged over all the users for upper level. Note that a shorter return day implies better UL performance. For the lower level, we report the expected number of clicks over five requests (horizon T of the LL) per user.

$$\mathbb{E}[r^U] = \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}_{r_i^U\sim P_{r^U}}[r_i^U]$$

$$\mathbb{E}[r^L] = \frac{1}{n}\sum_{i=1}^{n}\sum_{t=1}^{5}\mathbb{E}_{r_{t,i}^L\sim P_{r_i^L}}[r_{t,i}^L]$$

where P_{r^U} is the return day probability. The click probability $P_{r_i^L}$ at the lower level is a bernoulli distribution based on predictive reward model $\hat{q}^L$ of ranking scores. We define Oracle UL policy, which deterministically takes the action a^U in alignment with the unknown user preferences.

Results The results in Figure 4 and 5 demonstrate that our method can successfully learn the UL policy such that the long term objective is maximized. It also shows that the baselines fall short and the gap between our approach and baselines for the UL performance is significant.

For the LL, we use an epsilon-greedy rule to vary the quality of π^L. This is interesting as it simulates the effect of varying quality of the LL policy.

$$a^{L*} \leftarrow (1-\epsilon)\mathbb{I}(\arg\max_{a^L}(f(\hat{q}^L(.,a^L),a^U))) + \frac{\epsilon}{|\mathcal{A}^L|}$$

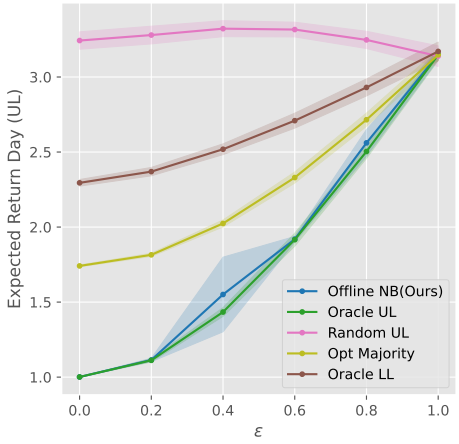

Figure 4. Effect of varying ϵ for the ϵ-greedy LL policy on the expected return day of users. Shorter return day is better.

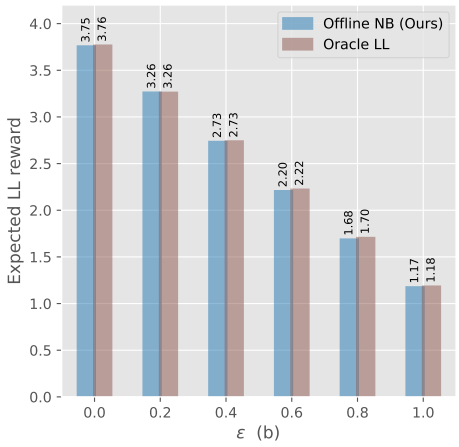

Figure 5. Effect of varying ϵ for the ϵ-greedy LL policy on expected LL reward

where $\epsilon \in [0, 1]$. The results are averaged across 20 random seeds. Figure 4 shows that our nested bandit method matches closely with the Oracle UL and has a much shorter return day as compared to Oracle LL. This shows that our method learns the optimal UL intervention for different user groups. The Opt Majority baseline that selects the optimal intervention deterministically for the majority user group performs better than Oracle LL but worse than our approach. This is because Opt Majority baseline selects the same intervention at UL for all users, resulting in a sub-optimal policy for 20% of the users. Finally, the Random UL baseline has the highest expected return day since it plays a random intervention at UL. Further, across all the varying quality of π^L, from the optimal ($\epsilon = 0.0$) to random ($\epsilon = 1.0$), our nested bandit approach performs similarly to Oracle UL. As the quality of LL policy degrades towards uniform random policy, the return day at UL also increases which shows that the quality of LL policy contributes to the quality of UL

policy. Figure 5 compares the expected short term rewards between Oracle LL and our ϵ-greedy LL policy. While the short term rewards degrade with ϵ as expected, our method is competitive with the short term Oracle (Oracle LL) baseline.

In summary, these results show that our nested bandits approach learns the optimal long term outcome while hardly compromising the short term reward. Overall, they indicate that our proposed bilevel formulation for steering toward long term outcomes is a promising approach, and has the potential to open new research avenues in this area.

5. Conclusion

In this work, we consider a more realistic model of human feedback at multiple scales of resolution. Our key insight is that, to achieve alignment with long term outcomes, we need to reconcile the disconnect between the long term and short term feedback. Our solution to this issue is to propose a hierarchical structure of interventions and feedback for modeling nuanced contexts with diverse user preferences. By leveraging this structure, we propose a novel bilevel formulation for policy learning that optimizes both upper and lower level policies for achieving long term outcomes. This is an ongoing project and we will explore realistic settings of policy learning at multiple levels for large number of user groups with diverse preferences and for large number of interventions to steer towards desired long term outcomes.

References

Athey, S., Chetty, R., Imbens, G. W., and Kang, H. The surrogate index: Combining short-term proxies to estimate long-term treatment effects more rapidly and precisely. Working Paper 26463, National Bureau of Economic Research, November 2019. URL http://www.nber.org/papers/w26463.

Beygelzimer, A. and Langford, J. The offset tree for learning with partial labels. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 129–138, 2009.

Colson, B., Marcotte, P., and Savard, G. An Overview of Bilevel Optimization. *Annals of Operations Research*, 153(1):235–256, June 2007. ISSN 0254-5330, 1572-9338. doi: 10.1007/s10479-007-0176-2. URL http://link.springer.com/10.1007/s10479-007-0176-2.

Elkins, S., Kochmar, E., Serban, I., and Cheung, J. C. K. How useful are educational questions generated by large language models? In Wang, N., Rebolledo-Mendez, G., Dimitrova, V., Matsuda, N., and Santos, O. C. (eds.), *Artificial Intelligence in Education. Posters and Late Breaking Results, Workshops and Tutorials, Industry and Innovation Tracks, Practitioners, Doctoral Consortium and Blue Sky*, pp. 536–542, Cham, 2023. Springer Nature Switzerland. ISBN 978-3-031-36336-8.

Horvitz, D. G. and Thompson, D. J. A generalization of sampling without replacement from a finite universe. *Journal of the American statistical Association*, 47(260):663–685, 1952.

Mansoury, M., Abdollahpouri, H., Pechenizkiy, M., Mobasher, B., and Burke, R. Feedback loop and bias amplification in recommender systems. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, CIKM '20, pp. 2145–2148, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450368599. doi: 10.1145/3340531.3412152. URL https://doi.org/10.1145/3340531.3412152.

Maystre, L., Russo, D., and Zhao, Y. Optimizing audio recommendations for the long-term: A reinforcement learning perspective, 2023.

McDonald, T. M., Maystre, L., Lalmas, M., Russo, D., and Ciosek, K. Impatient bandits: Optimizing recommendations for the long-term without delay. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '23, pp. 1687–1697, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400701030.

doi: 10.1145/3580305.3599386. URL https://doi.org/10.1145/3580305.3599386.

Swaminathan, A. and Joachims, T. Counterfactual risk minimization: Learning from logged bandit feedback. In *International Conference on Machine Learning*, pp. 814–823. PMLR, 2015.

Zhao, K., Liu, S., Cai, Q., Zhao, X., Liu, Z., Zheng, D., Jiang, P., and Gai, K. Kuaisim: A comprehensive simulator for recommender systems. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. URL https://openreview.net/forum?id=dJEjgQcbOt.