Transferable Contextual Bandits with Prior Observations

Kevin Labille[1 [0000−0001−8115−2353]], Wen Huang[1 [0000−0002−8210−4676]], and Xintao Wu[1 [0000−0002−2823−3063]]

University of Arkansas, Fayetteville, AR 72701, USA
{kclabill, wenhuang, xintaowu}@uark.edu

Abstract. Cross-domain recommendations have long been studied in traditional recommender systems, especially to solve the cold-start problem. Although recent approaches to dynamic personalized recommendation have leveraged the power of contextual bandits to benefit from the exploitation-exploration paradigm, very few works have been conducted on cross-domain recommendation in this setting. We propose a novel approach to solve the cold-start problem under the contextual bandit setting through the cross-domain approach. Our developed algorithm, T-LinUCB, takes advantage of prior recommendation observations from multiple domains to initialize the new arms' parameters so as to circumvent the lack of data arising from the cold-start problem. Our bandits therefore possess knowledge upon starting which yields better recommendation and faster convergence. We provide both a regret analysis and an experimental evaluation. Our approach outperforms the baseline, LinUCB, and experiment results demonstrate the benefits of our model.

Keywords: Contextual bandits · Cross-domain recommendation · Personalized recommendation.

1 Introduction

Personalized recommendation has long been studied through traditional approaches such as content-based techniques and collaborative filtering techniques. Yet, in recent years, it has been tackled through a new approach known as the exploration-exploitation dilemma. Indeed, an efficient recommender system should be able to recommend items that are both diverse and accurate. Naturally, diversity can be achieved through the exploration of new horizon and unknown interests while accurate predictions can be achieved through the exploitation of historical and known user interests. The key factor of such an approach thus becomes to properly balance exploration and exploitation in order to optimize the recommendation. Early works to tackle this problem were formulated as the multi-armed bandit (MAB) problem [2].

Although multi-armed bandits directly tackle the exploration-exploitation dilemma, they would be ineffective to use for personalized recommendation purposes, since they do not incorporate user-side information. To circumvent such limitations, contextual multi-armed bandits (or CMAB) [12] were introduced. Contextual bandits have the capability to observe, at each iteration, some features related to both the arm and the user.

As opposed to regular multi-armed bandits which only use the rewards to update their model, contextual bandits use the rewards along with the contextual feature vector to update the arm-picking strategy. By exploring the relationship between the context and the observed reward, contextual bandits are able to improve upon multi-armed bandits by making personalized decisions.

Both the MAB and CMAB have been applied to recommendation systems [4]. However, these approaches still suffer from the cold-start problem. A common method is to leverage observations from another domain and transfer them to the new domain. We study the problem of cross-domain recommendations under the linear contextual bandit setting. Specifically, we focus on the task of using a bandit capable of recommending educational videos (i.e. the arms) across various topics. We make the following assumptions: (1) the set of users remain unchanged across topics, (2) the topic and the set of arms change over time, and (3) the topics or domains are homogeneous, that is, they have the same feature space. In such a setting, the challenge for the bandit is to maintain accurate recommendations across topics (or domains) without restarting its learning strategy from scratch. To address this problem, we develop a new algorithm, T-LinUCB, which leverages recommendation observations of similar arms from prior topics. Consequently, the learning process is sped up and the estimation of the true reward parameters is improved, which results in better recommendations.

2 Related Work

There exist many approaches to solve the contextual bandits problem. Langford and Zhang [10] introduced an epoch-greedy approach, Li et al. [12] used a UCB-based approach that assumes a linear payoff model, and Agrawal et al. [1] tackled the problem using a Thompson sampling approach. Bandits have been widely applied to recommendation systems. Zhou et al. [19] and Nguyen & Kofod-Petersen [14] leveraged the context-free bandit to solve the widely-known cold-start problem present in recommender systems. Li et al. [12] used a contextual bandit based on the UCB algorithm while Chapeller & Li [5] investigated a Thompson-sampling approach for news item recommendation purposes. Bouneffouf et al. [3] used contextual bandits for recommendations in mobile environments. Nguyen & Lauw [15] proposed to tackle recommendation with a large population by dynamically clustering users into several clusters that are each served by a contextual bandit. Huang et al. [9] studied how to achieve user-side group fairness in contextual bandits. Tang et al. [17] explored ensemble strategies of different contextual bandits to make a recommendation decision.

Cross-domain recommendation has long been studied and is still an active research topic [8]. However, very few works take advantage of the powerful contextual bandit framework. Azar et al. [11] introduced the transfer-UCB Bandit algorithm that uses a transfer learning approach wherein they leverage prior knowledge by transferring the estimated bandit parameters from one task to another. Zhang & Bareinboim [18] tackled the offline transfer problem between bandits using a causal inference approach named B-kl-UCB. Although these two works are related, they focus on the context-free multi-armed bandit (MAB) as opposed to the contextual bandit. More recently, Liu et al. [13] introduced TCB where they tackled the cross-domain problem in contextual bandit us-

ing a transfer learning approach. TCB relies on a source and a target domain, as well as a matrix of correspondence data that captures the relatedness of the source and the target observations. It uses a translation matrix to align feature spaces between both domains and to translate the contexts. Their approach successfully outperformed several single-domain bandits. However, their TCB algorithm is set in the uniform contextual bandit model wherein there exists a single unknown reward parameter vector shared between all arms. We consider the more common disjoint contextual bandit model wherein each arm has its own unknown reward parameter. Furthermore, their setting only considers the problem of cross-domain recommendations from a single source domain. We extend upon this limitation and consider the more general case of having multiple source domains. Indeed, in the event that the bandit has access to several past topics or domains, their TCB algorithm would have to choose a single one to be the source to learn from.

3 Background

Throughout this paper, we use bold letters to denote a vector, e.g., $\mathbf{x}$, and capital bold letters to denote a matrix, e.g., $\mathbf{A}$. We use $||\mathbf{x}||_2$ to define the $\boldsymbol{\ell}_2$-norm of a vector $\mathbf{x} \in \mathbb{R}^d$. For a positive definite matrix $\mathbf{A} \in \mathbb{R}^{d \times d}$, we define the weighted $\boldsymbol{\ell}_2$-norm of $\mathbf{x} \in \mathbb{R}^d$ to be $||\mathbf{x}||_{\mathbf{A}} = \sqrt{\mathbf{x}^{\mathrm{T}}\mathbf{A}\mathbf{x}}$. We define the operation $\mathbf{A} \oplus \mathbf{B}$ as the row concatenation of matrices $\mathbf{A}$ and $\mathbf{B}$ and $\mathbf{a} \oplus \mathbf{b}$ as the regular concatenation of vectors $\mathbf{a}$ and $\mathbf{b}$. The notation $|\mathbf{x}|$ represents the magnitude of a vector $\mathbf{x}$. Finally, we denote $diag(\mathbf{v})$ the operation of making a square diagonal matrix with the elements of vector $\mathbf{v}$ on the main diagonal.

We revisit the linear contextual bandit (LinUCB) [6]. Formally, there is a set of users u also known as "bandit players" and a set of arms $a \in \mathcal{A}$ that are the items to be recommended. At time t, a user u comes in with the set of arm $\mathcal{A}$, and the bandit observes the contextual feature vector $\mathbf{x}_{t,a} \in \mathbb{R}^d$ for arm a, that represents the information of both the user and the arm. LinUCB assumes that the expected reward for each action is linear in its d-dimensional features $\mathbf{x}_{t,a}$ with some unknown coefficient vector $\boldsymbol{\theta}^*_a$.

The algorithm chooses an arm $a_t \in \mathcal{A}$ to recommend, observes the reward $r_{t,a} = \langle \boldsymbol{\theta}^*_a, \mathbf{x}_{t,a} \rangle + \epsilon_t$ where ϵ_t is the noise term, and then updates its arm recommendation strategy with the new observation $(\mathbf{x}_{t,a_t}, a_t, r_{t,a_t})$. LinUCB applies ridge regression to estimate the true coefficients. Let $D_a \in \mathbb{R}^{m_a \times d}$ denote the context of the historical observations when arm a is selected and $\mathbf{b}_a \in \mathbb{R}^{m_a}$ denote the relative rewards. The regularised least-square estimator for $\boldsymbol{\theta}_a$ could be expressed as:

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

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

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

Li et al. [12] derived a confidence interval that contains the true expected reward. Following the rule of optimism in the face of uncertainty for linear bandits (OFUL), this

confidence bound leads to a reasonable arm-selection strategy:

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

where $A_a = D_a^{\mathrm{T}} D_a + \lambda I_d$.

Formally, the expected reward at time t with arm a is expressed as $\mathbb{E}[r_{t,a}|\mathbf{x}_{t,a}] = \boldsymbol{\theta}_a^{*\mathrm{T}} \mathbf{x}_{t,a}$. During the learning process, the algorithm only observes the reward of the chosen arm. The total reward by round t is defined as $\sum_t r_{t,a_t}$ while the optimal expected reward is defined as $\mathbb{E}[\sum_t r_{t,a^*}]$, where a^* indicates the arm that can achieve the optimal reward at time t. We call T-trial regret $R(T)$, the difference between the optimal reward and the observed reward over T rounds: $R(T) = \mathbb{E}[\sum_t r_{t,a^*}] - \mathbb{E}[\sum_t r_{t,a_t}]$. The contextual bandit algorithm balances exploration and exploitation to maximize the expected total reward. Equivalently, the algorithm aims to minimize the total regret.

4 T-LinUCB

4.1 Problem overview

Consider the problem of recommending educational videos to a class of students. Formally, we model the personalized video recommendation as a contextual bandit problem, where each student t is a bandit player and each video $a \in \mathcal{A}$ is an arm. The videos are divided into $\mathcal{L}$ topics where each topic $l \in \mathcal{L}$ has a pool $\mathcal{A}_l$ of videos. Each video belongs to one **single** topic. We assume that the set of students remain unchanged across the topics. Given a topic $l \in \mathcal{L}$ and a student t, the goal of the bandit is to choose an arm $a \in \mathcal{A}_l$ that maximizes the reward. We further assume that each video a has a true unknown coefficient vector $\boldsymbol{\theta}_a^*$ that remains unchanged for the entirety of the topic. Thus, similarly to a typical contextual bandit problem, the goal is to estimate the unknown coefficient vector $\boldsymbol{\theta}_a^*$ for each video of the current topic l. However, unlike a typical contextual bandit problem, the arm pool $\mathcal{A}$ changes from one topic to another, meaning that the unknown coefficient vectors $\boldsymbol{\theta}_a^*$ have to be re-estimated. LinUCB algorithm is not designed to handle changing coefficient vectors $\boldsymbol{\theta}_a^*$. Indeed, for each individual topic $l \in \mathcal{L}$, a new LinUCB algorithm has to be re-started, where it would have to learn the new estimates of $\boldsymbol{\theta}_a^*$ from scratch again. Such an approach would yield lower performances.

We intend to tackle the problem of cross-domain recommendations and to solve these limitations of LinUCB by utilizing observations acquired from past topics to initialize the parameters of the new arms. The bandit therefore possesses knowledge upon the start of a new topic which results in better performances and faster regret convergence when compared to a cold start situation.

4.2 Algorithm design

Henceforth, we assume that we have $\mathcal{L}$ topics denoted by l ($l = 1, 2, ..., \mathcal{L}$) where each has a pool $\mathcal{A}_l$ of videos (or arms). We assume that the contextual feature vector of an arm a is denoted as $\mathbf{x}_a \in \mathbb{R}^n$. The contextual bandit algorithm runs in a sequential

Algorithm 1 T-LinUCB

1: Input: $\alpha \in \mathbb{R}^+, k \in \mathbb{N}^+, l$
2: **for** $a \in \mathcal{A}_l$ **do**
3: Observe contextual features of arm $a \in \mathcal{A}_l : \mathbf{x}_a \in \mathbb{R}^n$
4: $\mathbf{A}_a, \mathbf{b}_a \leftarrow INIT(\mathbf{x}_a, k)$
5: **end for**
6: **for** t = 1,2,...,T **do**
7: **for** $a \in \mathcal{A}_l$ **do**
8: Observe contextual features of arm $a \in \mathcal{A}_l : \mathbf{x}_{t,a} = (\mathbf{x}_t, \mathbf{x}_a) \in \mathbb{R}^d$
9: $\hat{\boldsymbol{\theta}}_a \leftarrow (\mathbf{A}_a)^{-1}\mathbf{b}_a$
10: $p_{t,a} \leftarrow \hat{\boldsymbol{\theta}}_a^{\mathrm{T}}\mathbf{x}_{t,a} + \alpha\sqrt{\mathbf{x}_{t,a}^{\mathrm{T}}(\mathbf{A}_a)^{-1}\mathbf{x}_{t,a}}$
11: **end for**
12: Choose arm $a_t = argmax_{a \in \mathcal{A}_l} p_{t,a}$ with ties broken arbitrarily, and observe a real-valued payoff r_{t,a_t}
13: $\mathbf{A}_{a_t} \leftarrow \mathbf{A}_{a_t} + \mathbf{x}_{t,a_t}\mathbf{x}_{t,a_t}^{\mathrm{T}}$
14: $\mathbf{b}_{a_t} \leftarrow \mathbf{b}_{a_t} + r_{t,a_t}\mathbf{x}_{t,a_t}$
15: **end for**

fashion for $t = 1, 2, \ldots, T$ given a particular topic $l \in \mathcal{L}$. At each time t, a student plays the bandit which reads the student's information and must choose an arm $a \in \mathcal{A}_l$ that maximizes the reward. We thus have the contextual feature vector $\mathbf{x}_{t,a} \in \mathbb{R}^d$ which encompasses both information from the user and the arm. We assume that the dimension of the vectors remains the same throughout and across all topics.

Algorithm 1 shows the main T-LinUCB algorithm. We first initialize all of the arms' parameters $(\mathbf{A}, \mathbf{b})$ of the current topic l using the historical observations from k historical topics (line 4). Once the arms are initialized, the algorithm runs as a traditional LinUCB. We show in Algorithm 2 the procedure that initializes the parameters $(\mathbf{A}, \mathbf{b})$ of an arm using historical observations. $\mathbf{D}_a$ represents a matrix of observations from k previous topic(s), $\mathbf{c}_a$ represents a vector of the corresponding historical responses, and $\mathbf{w}$ represents a vector of weights. We compute the similarity score between the current arm a of the current topic l and every arms a_h in the k previous topics using the Euclidean distance between their respective contextual feature vector, i.e., $\mathbf{x}_a$ and $\mathbf{x}_{a_h}$ (line 6). Because we only make use of historical arms that share some degree of similarity in the feature space, we compare the resulting similarity scores to a threshold τ. The design matrix $\mathbf{D}_{a_h}$ and the corresponding response vector $\mathbf{c}_{a_h}$ of the most similar arms are then concatenated together to form the design matrix $\mathbf{D}_a$ and response vector $\mathbf{c}_a$ of the current arm (line 9-10).

Similarly, a weight vector $\mathbf{w}_{a_h}$ with values ranging from 0 to 1 stores the weight of all historical observations from arm a_h. The weight vector of arm a, $\mathbf{w}_a$, is the aggregation of the weight vector of all similar arms (line 11-12). We consider that the more recent an observation is, the more valuable it is and therefore the larger its weight should be. The impact of the observation in the previous topics will thus decay with the

time interval according to the following formula (line 10) of Algorithm 2:

$$w = exp(-\frac{p||\mathbf{x}_a - \mathbf{x}_{a_h}||_2}{2\eta^2})$$

where η is a parameter that controls the decaying speed. We then create a diagonal matrix $\mathbf{W}_a$ with the elements of the vector $\mathbf{w}_a$ on the main diagonal. The weight matrix $\mathbf{W}_a$ along with the design matrix $\mathbf{D}_a$ and the corresponding response vector $\mathbf{c}_a$ are then used to initialize the arm's parameters $\mathbf{A}_a$ and $\mathbf{b}_a$ (line 17-18). Additionally, τ is computed from the average similarity scores and their standard deviation as follows: $\tau = \bar{s} + \gamma\ \sigma$ where $\bar{s}$ is the average of the similarity scores and σ is their standard deviation. The threshold τ has a parameter γ that controls the weight of the standard deviation. Specifically, the higher it is, the more restrictive the threshold becomes, and the smaller the number of similar arms we will make use of.

Algorithm 2 Initialize - Get the initialized matrix related to each arm

1: $INIT(\mathbf{x}_a, k)$
2: $\mathbf{D}_a \leftarrow \mathbf{0}_{0\times d}, \mathbf{c}_a \leftarrow [\,], \mathbf{w}_a \leftarrow [\,]$
3: **for** $p = 1, ..., k$ **do**
4: Observe contextual features of all arms $a_h \in \mathcal{A}_p : \mathbf{x}_{a_h} \in \mathbb{R}^n$
5: **for** $a_h \in \mathcal{A}_{l-p}$ **do**
6: $SIM(\mathbf{x}_a, \mathbf{x}_{a_h}) = ||\mathbf{x}_a - \mathbf{x}_{a_h}||_2$
7: **if** $SIM(\mathbf{x}_a, \mathbf{x}_{a_h}) \geq \tau$ **then**
8: $\mathbf{D}_a = \mathbf{D}_a \oplus \mathbf{D}_{a_h}$
9: $\mathbf{c}_a = \mathbf{c}_a \oplus \mathbf{c}_{a_h}$
10: $w = exp(-\frac{p||\mathbf{x}_a - \mathbf{x}_{a_h}||_2}{2\eta^2})$
11: $\mathbf{w}_{a_h} \leftarrow w_{|\mathbf{c}_{a_h}|\times 1}$
12: $\mathbf{w}_a = \mathbf{w}_a \oplus \mathbf{w}_{a_h}$
13: **end if**
14: **end for**
15: $\mathbf{W}_a \leftarrow diag(\mathbf{w}_a)$
16: **end for**
17: $\mathbf{A}_a \leftarrow \mathbf{D}_a^{\mathrm{T}}\mathbf{W}_a\mathbf{D}_a + \lambda\mathbf{I}_d$
18: $\mathbf{b}_a \leftarrow \mathbf{D}_a^{\mathrm{T}}\mathbf{W}_a\mathbf{c}_a$
19: **return** $\mathbf{A}_a, \mathbf{b}_a$

4.3 Regret analysis

There are several works that give detailed regret analysis on the non-stationary environments. Among them, [16] has the most similar setting as ours. It assigns time-decaying weight to previous observations and obtains $O(d^{2/3}B_T^{1/3}T^{2/3})$ regret bound, where d represents the feature dimension, T represents time horizon, and $B_T = \sum_{s=1}^{T-1} ||\theta_s^* - \theta_{s+1}^*||_2$ denotes the variation budget of the coefficients.

In previous works, the change-points of the reward function are usually unknown in advance. However, in the recommendation process discussed in our paper each transformation of the topic will raise an abrupt change of the reward function, which means that we are able to know each changing time point beforehand. Thus T-LinUCB could be regarded as an oracle linear bandit algorithm that restarts LinUCB algorithm with historical observations as side information at each changing point. It helps us get rid of the variation budget of the coefficients and achieve a long-term regret bound of $\tilde{O}(d^{1/2}T^{1/2})$. From the experiment section we can see that in most cases the regret of T-LinUCB is significantly less than LinUCB algorithm for each topic and enjoys a faster convergence speed. One disadvantage for T-LinUCB is that the computational complexity might be higher since it needs to incorporate historical information when initializing observation matrices and conducting matrix multiplication.

5 Experimental Evaluation

5.1 Experiment Setup

Simulated Dataset We evaluate the performances of our approach on a simulated dataset that fits our scenario and allows us to model a change of topic. Our simulated environment combines both of the following publicly available datasets.

- **Adult dataset**: The Adult dataset [7] is composed of 31,561 instances: 21,790 males and 10,771 females, each having 8 categorical variables (work class, education, marital status, occupation, relationship, race, sex, native-country) and 3 continuous variables (age, education number, hours per week), yielding an overall of 107 features after one-hot encoding.
- **YouTube dataset**: The Statistics and Social Network of YouTube Videos [1] is composed of 4,522 instances separated into four categories: Comedy (1,580), Music (1,819), Sports (932), and Travel & Places (191). Each instance has 6 categorical features (age of video, length of video, number of views, rate, ratings, number of comments), yielding a total of 25 features after one-hot encoding.

Our users (bandit players) are represented using the Adult dataset. For our experiments we use a subset of 10,000 instances drawn randomly and we assume that the set of users remain unchanged across topics. Similarly, our videos (or arms) are represented through the Youtube dataset. For our experiments we will be using several topics which are each represented by a Youtube category. For each topic we select a random subset as our pool of videos to recommend. In particular, topic 1 uses 30 videos from the Comedy category, topic 2 uses 20 videos from the Music category, topic 3 uses 20 videos from the Sports category, and topic 4 uses 30 videos from the Travel & Places category. We reduce the dimensionality of both the user and video feature vectors through Principal Component Analysis (PCA) by choosing a number of components that explains 80% of the variance. Thereafter, the dimensions of the user feature vectors are reduced to 19 while the dimensions of the video feature vectors are reduced to 7. Throughout the experiment, we use the concatenation of both the user feature vector and the video feature vector as our contextual feature vector $\mathbf{x}_{t,a}$, yielding a total of 26 features.

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

Reward Functions The reward mechanism follows that of LinUCB where the reward of an arm a is assumed to be the noisy linear combination of its context vector and and unknown coefficient vector (also called unknown reward parameters vector) $\boldsymbol{\theta}_a^*$. Specifically $r_{t,a} = \langle \mathbf{x}_{t,a}, \boldsymbol{\theta}_a^* \rangle + \epsilon$ where ϵ is a random Gaussian noise, i.e., $\epsilon \sim \mathcal{N}(0, 0.01)$. For each arm within a topic, we generate the unknown coefficient vectors $\boldsymbol{\theta}_a^*$ by randomly drawing each of the 26 dimensions from a Gaussian distribution, i.e., $\mathcal{N}(0.5, \sigma)$ where σ is drawn randomly from a normal distribution, i.e., $\sigma \sim \mathcal{U}(0, 1)$. We then normalize the reward parameters such that the Manhattan norm of the vector is equal to 1. As a consequence, the reward generated in our setting is bounded between 0 and 1.

Evaluation Metric We use the regret to evaluate the performances of the algorithms. Since the true reward function is known in our simulated environment, it is possible to compute the regret over T rounds: $R(T) = \mathbb{E}[\sum_t r_{t,a^*}] - \mathbb{E}[\sum_t r_{t,a_t}]$ where the first term is the optimal reward, and the second term is the observed reward at time t.

5.2 Experimental Results

Our intuition is that using prior knowledge from multiple topics can help initializing the parameters of the bandit for a new topic thereby circumventing the cold-start problem. To confirm our intuition, we compare the performances of our T-LinUCB algorithm to the classic LinUCB algorithm.

Impact of the Decaying Factor η We first investigate the impact of the decaying factor η introduced in Algorithm 2 (line 10) on our first two topics and report the cumulative regret at topic 2 on Figure 1. For this experiment, γ is set to 1 since we have not investigated its effect yet. As introduced in Section 4.2, η is a decaying factor that allows to control the weight of the historical observations. Specifically, the more recent the observations are, the larger the weights are. Our intuition is that larger weights will speed up the learning process thereby decreasing the regret. As Figure 1 shows, the higher η is, the lower the regret is at topic 2. Indeed, with an η close to 0, the weights of the historical observations are almost nil, T-LinUCB will thus behave as a regular LinUCB, achieving a regret of 362.45. The regret stabilizes when η reaches 2, with a regret oscillating between 106.13 and 101.33. These empirical results confirm our intuition that historical observations with larger weights provides the bandit with stronger knowledge and therefore accelerates the learning of the unknown coefficients $\boldsymbol{\theta}_a^*$.

Impact of the Parameter γ As introduced in Section 4.2, γ is used in the computation of the threshold τ as a parameter to control the weight of the standard deviation. A higher γ yields a higher value of τ, which translates into making the algorithm more restrictive as to the inclusion of an arm into the historical data. Therefore, γ has a direct impact on the number of similar arms to consider. We aim at understanding the impact of γ on our algorithm. We run T-LinUCB with the first two topics with various values of γ ranging from 0.0 to 3, and compare their performances. We report the cumulative regret at topic 2 for various γ on Figure 2. For this experiment, η is set to 5 as per the results achieved previously.

As shown on Figure 2, a large regret is achieved when γ is either too small or too large (313.74 for $\gamma = 0$, 319.5 for $\gamma = 0.5$, 252.58 for $\gamma = 1.5$, 361 for $\gamma = 2$ and 3). Indeed, in the former case a large number of arms satisfy the threshold condition (Alg. 2 line 7), yielding too many arms to be deemed similar enough. Considering a large number of arms will introduce noisy observations and negatively impact the performances of T-LinUCB. Conversely, on the latter case, very few arms satisfy the threshold condition, yielding a lower number of arms to be considered. Consequently, T-LinUCB does not have sufficient information to initialize the arms in the current topic, and will behave similarly to a traditional LinUCB. Finally, the performances of T-LinUCB are drastically improved with γ being close to 1. Indeed, when $\gamma = 1.1$ the regret at topic 2 drops to 100.4 In such a case the bandit collects sufficient historical observations that help it initialize the parameters of the arms of the new topic by taking full advantage of the past. This experiment shows that γ, which controls the number of similar arms to use, plays an important role in the initialization process that can severely affect the performances of T-LinUCB. While a low value of γ brings noisy observations, a high value of γ allows not enough historical observations to be used for initialization.

Robustness to the Change of the Unknown Coefficient Vectors We investigate the robustness of our T-LinUCB algorithm to the degree of change of the unknown coefficient vectors (or unknown reward parameters), $\boldsymbol{\theta}_a^*$, from one topic to another. Particularly, in this setting, the unknown coefficient vectors of the first topic remain unchanged whereas the unknown coefficient vectors of the second topics are drawn randomly from a Gaussian distribution $\mathcal{N}(0.5, \sigma)$ with increasing standard deviation σ. Based upon our previous empirical results, we set the parameters η to 1.1 and γ to 5 as they achieved the best performances. We compare and run LinUCB versus T-LinUCB ten times per value of σ and report the averaged regret at topic 2 in Figure 3. As depicted in Figure 3, our T-LinUCB algorithm is much more robust to the degree of change of the reward parameters from one topic to another than LinUCB is. Indeed, our algorithm consistently achieves a lower regret with an average of 126.99 against 307.916 for LinUCB, which has a decrease of 142.45%. Furthermore, T-LinUCB achieves a much steadier regret that has a variance of 87.35 against 342.87 for LinUCB. These results confirm that our T-LinUCB is robust to the change of reward parameters and that it not only achieves a much lower regret than LinUCB but also maintains a consistent regret.

T-LinUCB vs LinUCB with 2 topics We compare our algorithm to LinUCB with two topics with k set to 1, that is, our algorithm only uses observations from 1 prior topic. Based upon our previous empirical results, we set the parameters η to 1.1 and γ to 5 as they achieved the best performances. Figure 4 shows the regret over topic 2 for both LinUCB and our T-LinUCB. Since both algorithms learn without a-priori knowledge during topic 1, they are expected to achieve the same regret. In the second topic, however, the arm pool changes along with new unknown reward parameters, $\boldsymbol{\theta}_a^*$. As Figure 4 shows, our approach outperforms LinUCB greatly and achieves a much lower regret of 359.29 for LinUCB against 100.08 for T-LinUCB.

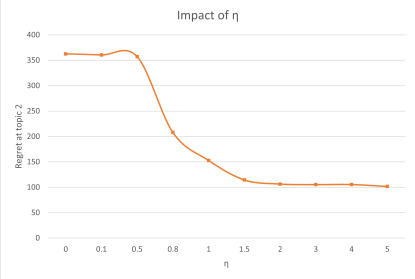

Fig. 1: Regret for various values of η

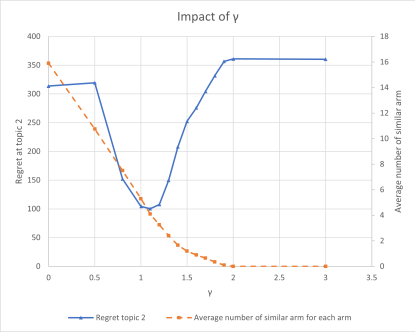

Fig. 2: Regret for various values of γ

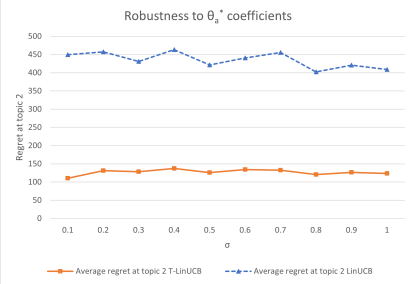

Fig. 3: Regret for various value of σ

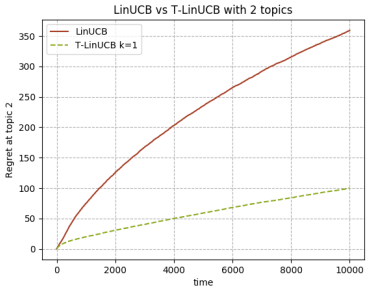

Fig. 4: Regret with two topics

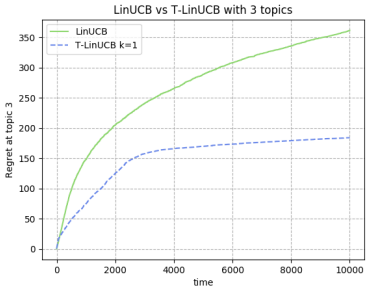

Fig. 5: Regret with three topics

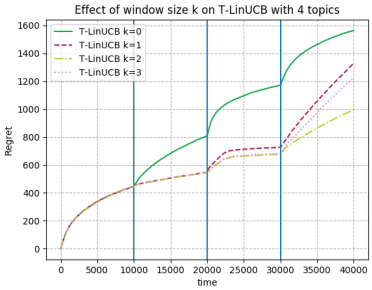

Fig. 6: Regret with various k size

T-LinUCB vs LinUCB with 3 topics We check the long-term benefit of our approach by running the same experiment using three topics instead of two. We report the regret at topic 3 for both LinUCB and our T-LinUCB on Figure 5. Similarly to the previous scenario, LinUCB will start learning from scratch for all three topics while T-LinUCB will make use of historical observations from topic 1 when switching to topic 2, and from topic 2 when switching to topic 3. We notice on Figure 5 that, as expected, T-LinUCB outperforms LinUCB. Indeed, the former achieves a regret of 361.87 at topic 3 against 184.02 for the latter. Moreover, we can see that LinUCB has not converged after

$T = 10,000$ rounds, which indicates that it is still learning, as opposed to T-LinUCB which converges much faster, emphasizing yet another benefit of our approach. Our experiments demonstrate the benefits of using prior knowledge to avoid the cold-start problem. By initializing the parameters of the bandit in the new topic, the bandit already possesses knowledge that speeds up the estimation of the unknown reward parameters $\boldsymbol{\theta}_a^*$, yielding a much lower regret.

Impact of the Number of Historical Topic k We investigate how k affects our T-LinUCB. k controls the number of prior topics to learn from (Alg. 2, line 4). We run T-LinUCB with 4 topics with $k = 0, 1, 2, 3$. We report the cumulative regret over all four topics on Figure 6 wherein a vertical blue line indicates the start of a new topic. Based upon our previous empirical results, we set the parameters η to 1.1 and γ to 5 as they achieved the best performances. Figure 6 shows that all T-LinUCB instances that learn from prior knowledge outperform the baseline LinUCB (i.e., T-LinUCB with k=0). A regular LinUCB learns from scratch at each new topic, yielding a very high regret of 1567.98 at topic 4. When k is set to 1, T-LinUCB achieves a regret of 1328.96 at topic 4. With a k set to 2, T-LinUCB greatly outperforms both LinUCB and T-LinUCB k=1, with a regret of 996.05 at topic 4. Surprisingly enough, with k set to 3, T-LinUCB achieves a regret of 1229.96 at topic 4, which outperforms both LinUCB and T-LinUCB $k = 1$, but performs slightly under T-LinUCB $k = 2$. This could be due to the fact that historical observations that are too obsolete can introduce noisy information. Figure 6 noticeably shows the advantage of using historical knowledge from multiple topics to circumvent the cold-start problem and speed up the learning of the bandit. Indeed, the regret difference between LinUCB and T-LinUCB $k = 2$ is substantial at topic 4. The knowledge acquired from topics 1 and 2 by T-LinUCB allows it to estimate the unknown reward parameters more rapidly, thereby decreasing the regret drastically. The experimental results confirm our intuition that learning from multiple topics not only overcome the cold-start problem, but also allows it to converge faster.

6 Conclusions

We have developed a new contextual bandit algorithm that leverages historical observations from prior domain(s) to overcome the cold-start problem of personalized recommendation. Through the use of prior observations from multiple source domain(s) for initalization of the new arm's parameters, our T-LinUCB algorithm speeds up the learning of the unknown reward parameters and greatly improves the regret of the algorithm. Furthermore, our regret analysis showed that our approach achieves the same regret bound as the oracle linear bandit algorithm under the changing environment. Finally, our experimental results showed that T-LinUCB achieves a much lower regret and benefit from a faster convergence speed than the traditional LinUCB algorithm.

ACKNOWLEDGMENTS

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

References

1. Agrawal, S., Goyal, N.: Thompson sampling for contextual bandits with linear payoffs. In: International Conference on Machine Learning. pp. 127–135 (2013)
2. Berry, D.A., Fristedt, B.: Bandit problems: sequential allocation of experiments (monographs on statistics and applied probability). London: Chapman and Hall **5**(71-87), 7–7 (1985)
3. Bouneffouf, D., Bouzeghoub, A., Gançarski, A.L.: A contextual-bandit algorithm for mobile context-aware recommender system. In: International Conference on Neural Information Processing. pp. 324–331. Springer (2012)
4. Bouneffouf, D., Rish, I., Aggarwal, C.: Survey on applications of multi-armed and contextual bandits. In: 2020 IEEE Congress on Evolutionary Computation (CEC). pp. 1–8. IEEE (2020)
5. Chapelle, O., Li, L.: An empirical evaluation of thompson sampling. In: Advances in Neural Information Processing Systems. pp. 2249–2257 (2011)
6. Chu, W., Li, L., Reyzin, L., Schapire, R.: Contextual bandits with linear payoff functions. In: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics. pp. 208–214 (2011)
7. Dua, D., Graff, C.: UCI machine learning repository (2017), `http://archive.ics.uci.edu/ml`
8. Fernández-Tobías, I., Cantador, I., Kaminskas, M., Ricci, F.: Cross-domain recommender systems: A survey of the state of the art. In: Spanish Conference on Information Retrieval. pp. 1–12. sn (2012)
9. Huang, W., Labille, K., Wu, X., Lee, D., Heffernan, N.: Achieving user-side fairness in contextual bandits. CoRR **abs/2010.12102** (2020), `https://arxiv.org/abs/2010.12102`
10. Langford, J., Zhang, T.: The epoch-greedy algorithm for multi-armed bandits with side information. In: Advances in Neural Information Processing Systems. pp. 817–824 (2008)
11. Lazaric, A., Brunskill, E., et al.: Sequential transfer in multi-armed bandit with finite set of models. In: Advances in Neural Information Processing Systems. pp. 2220–2228 (2013)
12. Li, L., Chu, W., Langford, J., Schapire, R.E.: A contextual-bandit approach to personalized news article recommendation. In: Proceedings of the 19th International Conference on World Wide Web. pp. 661–670 (2010)
13. Liu, B., Wei, Y., Zhang, Y., Yan, Z., Yang, Q.: Transferable contextual bandit for cross-domain recommendation. In: AAAI (2018)
14. Nguyen, H.T., Kofod-Petersen, A.: Using multi-armed bandit to solve cold-start problems in recommender systems at telco. In: Mining Intelligence and Knowledge Exploration, pp. 21–30. Springer (2014)
15. Nguyen, T.T., Lauw, H.W.: Dynamic clustering of contextual multi-armed bandits. In: Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management. pp. 1959–1962 (2014)
16. Russac, Y., Vernade, C., Cappé, O.: Weighted linear bandits for non-stationary environments. In: Advances in Neural Information Processing Systems. pp. 12040–12049 (2019)
17. Tang, L., Jiang, Y., Li, L., Li, T.: Ensemble contextual bandits for personalized recommendation. In: Proceedings of the 8th ACM Conference on Recommender Systems. pp. 73–80 (2014)
18. Zhang, J., Bareinboim, E.: Transfer learning in multi-armed bandit: a causal approach. In: Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems. pp. 1778–1780 (2017)
19. Zhou, Q., Zhang, X., Xu, J., Liang, B.: Large-scale bandit approaches for recommender systems. In: International Conference on Neural Information Processing. pp. 811–821. Springer (2017)