Learning Hidden Features for Contextual Bandits

Huazheng Wang, Qingyun Wu, Hongning Wang
Department of Computer Science
University of Virginia, Charlottesville VA, 22904 USA
{hw7ww,qw2ky,hw5x}@virginia.edu

ABSTRACT

Contextual bandit algorithms provide principled online learning solutions to find optimal trade-offs between exploration and exploitation with companion side-information. Most contextual bandit algorithms simply assume the learner would have access to the entire set of features, which govern the generation of payoffs from a user to an item. However, in practice it is challenging to exhaust all relevant features ahead of time, and oftentimes due to privacy or sampling constraints many factors are unobservable to the algorithm. Failing to model such hidden factors leads a system to make constantly suboptimal predictions.

In this paper, we propose to learn the hidden features for contextual bandit algorithms. Hidden features are explicitly introduced in our reward generation assumption, in addition to the observable contextual features. A scalable bandit algorithm is achieved via coordinate descent, in which closed form solutions exist at each iteration for both hidden features and bandit parameters. Most importantly, we rigorously prove that the developed contextual bandit algorithm achieves a sublinear upper regret bound with high probability, and a linear regret is inevitable if one fails to model such hidden features. Extensive experimentation on both simulations and large-scale real-world datasets verified the advantages of the proposed algorithm compared with several state-of-the-art contextual bandit algorithms and existing ad-hoc combinations between bandit algorithms and matrix factorization methods.

Keywords

Contextual bandits, latent feature learning, online recommendations, regret analysis

1. INTRODUCTION

Contextual bandit algorithms [3, 7, 15, 14] have become a reference solution for modern information service systems to adaptively find good mappings between available content and users. It models the interaction between a system and its users as a stochastic game, and addresses the notorious explore-exploit dilemma [3, 4, 15] at a per-user basis. In particular, a service system equipped with a contextual bandit algorithm sequentially selects items to serve users using side information about the user and item, while adapting its selection strategy based on the immediate user feedback to maximize users' long-term satisfaction. Contextual bandits are especially advantageous when the space of recommendation is large but the payoffs are interrelated, such as content recommendation [15, 6, 26] and display advertising [8, 17].

A common practice in contextual bandits assumes the expected payoff is determined by a conjecture of unknown bandit parameters and given context, which is represented as a set of manually crafted features extracted from both users and recommendation candidates [1, 10, 15]. In other words, it assumes the stochastic game is *transparent*: the features that the environment (i.e., system users) uses to generate the payoff of each action are entirely accessible to the learner (i.e., the bandit algorithm). This assumption is unfortunately oversimplified and will introduce systematic bias during online learning, if the observed features are insufficient to predict the expected payoffs. Take news recommendation as an example. Typical features for news recommendation include a news article's recency, topical categories, popularity, and a user's location [15, 25]. However, some users might care more about the source of the news: they seldom read news from unconfirmed sources, but the trustworthiness of a news article's source is clearly orthogonal to the aforementioned features. If this dimension is not disclosed to the learner, it will inevitably lead to constantly suboptimal recommendations for such group of users. Arguably, it is impossible to exhaust all relevant attributes before developing a practical recommender system, and even more challenging to figure out the missing features after the system is deployed. Furthermore, in practice there are many factors unobservable to the learner, such as gender, age, location and income due to privacy constraints, but they are crucial for accurate recommendations. As a result, enhance the reward generation assumptions in contextual bandits and give algorithm the freedom to estimate those *hidden features* in addition to the observed ones become necessary and vital.

To the best of our knowledge, few work has been done in learning hidden features for contextual bandits; but the idea of hidden feature learning has been proved in the domain of collaborative filtering [22]. Most state-of-the-art collaborative filtering solutions are based on latent factor models, which outperform traditional content-based methods in many application scenarios [13, 5]. The latent features are learned via a low-rank approximation of the observed user-item interaction matrix. Recent developments also include observable attributes into factorization to improve the recommendation accuracy. Typical solutions include regression-based factor models [2] and factorization machines [20]. Nevertheless, those factorization-based models are incompetent to handle the interaction between a system and its users on the fly. For example, there is no mechanism for them to explore currently less promising items nor to capture the dynamics of users' interests, given

CIKM'16 , October 24-28, 2016, Indianapolis, IN, USA

DOI: http://dx.doi.org/10.1145/2983323.2983847

the latent factors are learned ahead of time on an isolated training set. Besides, there is no theoretical justification of how such latent factor models would perform in an online setting, e.g., how fast the discrepancy between the algorithm's choice and optimal choice will converge or diverge over time.

Some recent empirical studies combined bandit algorithms with factorization-based methods for online collaborative filtering [27, 12]. Basically, bandit algorithms are used to control the exploration of less promising recommendations for user feedback, and matrix factorization is applied over the incrementally constructed user-item matrix on the fly. However, these two components are integrated in an ad-hoc manner: both contextual and context-free bandits have been explored on top of matrix factorization, given they only provide an index of candidate items for feedback acquisition. There is no theoretical analysis to justify whether the combination would ensure desirable performance in a long run and how good/bad such algorithm is against the oracle recommendations over time, i.e., regret bound analysis.

In this paper, we propose to perform hidden feature learning for contextual bandits, with thorough regret analysis. Specifically, hidden features are explicitly introduced in our reward generation assumption, in addition to the observable contextual features. To simplify the discussion, linear dependency structure is postulated in our reward generation assumption; but it can be readily extended to more complicated dependency structures, e.g., generalized linear models [10]. Coordinate descent with provable exploration bound is used to iteratively estimate the hidden features and unknown model parameters on the fly. At each iteration, closed form solutions exist and can be efficiently computed. Most importantly, we rigorously prove that with proper initialization the developed contextual bandit algorithm with hidden features can still obtain a sublinear upper regret bound with high probability. Our analysis also demonstrates that if one fails to model the hidden features that play a role in reward generation, a linear regret is inevitable at the worst case, and the regret growth rate depends on the variance of those latent dimensions. In addition, we also prove that to scale up our algorithm in practice, periodic model update can be performed and it does not change the order of resulting regret bound. Extensive experimentations on both simulations and large-scale real-world datasets verified the advantages of the proposed algorithm compared with several state-of-the-art contextual bandit algorithms and existing ad-hoc combinations of bandit algorithms and matrix factorization methods.

2. RELATED WORK

To the best of our knowledge, no previous work has studied the problem of hidden feature learning for contextual bandits. But there are several lines of related work: 1) contextual bandit algorithms; 2) factorization-based latent feature learning; and 3) online collaborative filtering with bandits.

Contextual bandit algorithms assume the distributions of payoffs pertaining to each arm are connected by a set of common unknown parameters [3, 7, 15, 14, 10]. Two different types of models have been studied in literature. In the first type of models [14, 24], at each trial, side-information or context is given to the learner first. The payoffs of arms depend on both side-information and index of the arm. Thus the optimal arm changes with the context. In the second type [3, 7, 15, 10], which we are interested in this work, the learner is given a model that predicts the arms' payoffs based on the given context vectors of arms. Both linear models [9, 15] and generalized linear models [10] have been explored to capture such dependency relation. However, all contextual bandit algorithms assume the features that govern the underlying payoff generation are fully observable to the learner. If the payoff is also determined by some unobservable features, as discussed in the introduction, we can rigorously prove such algorithms suffer from linearly accumulated regret at the worst case. Our solution extends classical contextual bandit algorithms by explicitly learning the hidden features during online update; this reduces the linear regret back to sublinear with respect to the total number of iterations.

The idea of learning latent features has been successfully explored in collaborative filtering through matrix factorization [13, 22]. The basic idea roots in low rank approximation of the input user-item affiliation matrix. Traditional latent factor models only take user-item pairs as input and cannot easily incorporate additional attributes of users nor items. They are thus inept to make predictions on new users or new items, known as *cold-start* in recommendation [21]. Some recent developments include side-information for latent factor learning [2, 11, 18] to alleviate the cold-start challenge. Agarwal and Chen replaced traditionally used zero-mean Gaussian distribution with a regression-based mean, such that different types of contextual features can be effectively introduced [2]. The factorization machines proposed in [20] handle arbitrary orders of interactions between variables (i.e., tensor decomposition) and naturally incorporate observable features. However, all the aforementioned factorization-based methods take a static view of the interactions between users and items, and therefore are trained offline. It is computationally prohibited to update the model in a timely fashion, and there has been little theoretical development to justify its effectiveness in an online setting. In our solution, the estimation confidence of both model parameters and hidden features is used to control the explore/exploit trade-off during online learning, and it therefore leads to a provable sub-linear regret bound.

There are some recent developments that focus on online collaborative filtering with bandits. Zhao et al. studied interactive collaborative filtering via probabilistic matrix factorization [27]. Several bandit algorithms are introduced to perform online item selection based on the factorization results. Kawale et al. developed a Thompson sampling scheme for online matrix-factorization [12]. Latent features are extracted via online low-rank matrix completion, where the explore/exploit trade-off is balanced via Thompson sampling. Nakamura developed a UCB-like strategy to perform online collaborative filtering [19]. The algorithm deterministically selects feedback user-item pairs using an index which depends on the covariance matrices of the posterior distributions of both latent user and item vectors. However, due to the ad-hoc combination between factorization method and bandit method, little theoretical analysis is provided in these works. Our work for the first time gives rigorously proof of regret with hidden feature learning in contextual bandits. We provide upper regret bounds under different conditions to guide future research in this direction. In addition, all above methods fall into the traditional matrix factorization paradigm, i.e., no context features are considered. Our proposed solution leverages the observed attributes for hidden feature learning, which further improves the estimation quality of expected payoffs.

3. METHODOLOGY

We develop a contextual bandit algorithm with hidden feature learning. An enhanced reward generation assumption is given in the proposed model, where the hidden features are explicitly introduced in addition to the observed contextual features. Coordinate decent is used to estimate the unknown bandit parameters and hidden features, and to derive the exploration strategy for online learning. We rigorously prove that under proper initialization the result algorithm's upper regret bound stays in sublinear with high probability. Our theoretical analysis also demonstrates that if a contex-

tual bandit algorithm fails to model the hidden features that affect reward generation, a linear regret is inevitable at the worst case.

In this section, we will first describe the notations and our model assumptions about the hidden features in a contextual bandit problem, then carefully illustrate our developed bandit algorithm and corresponding regret analysis.

3.1 Contextual Bandit with Hidden Features

In a contextual bandit problem, at each of T rounds, a learner needs to make a choice among a finite, but possibly large, number of arms, which correspond to the candidate item set to be presented (such as articles in a content recommendation system). In particular, each arm is associated with certain observable side-information that is related to the expected payoff of this arm. We denote the arm set as $\mathcal{A}$ and the cardinality of $\mathcal{A}$ as K. Formally, a contextual bandit algorithm proceeds at discrete trials $t = 1, 2, 3, ..., T$ as follows. At each trial t, the learner first observes a given user u and a subset of arms from $\mathcal{A}$, where each arm a is associated with a feature vector $\mathbf{x}_a \in \mathbb{R}^d$ summarizing the side-information of arm a at trial t. Then based on the observed payoffs in previous trials, the learner chooses an arm a_t, displays it to user u, and receives the corresponding payoff $r_{a_t,u}$ from u. The goal of the learner is to update its arm-selection strategy with respect to the historic observations $\{(\mathbf{x}_{a_t}, r_{a_t,u})\}_{t=1}^{T}$, such that after T trials its *regret* with respect to the oracle arm selection strategy is minimized. In particular, the accumulated T-trail regret is defined formally as,

$$\mathbf{R}(T) = \sum_{t=1}^{T} R_t = \sum_{t=1}^{T} (r_{a_t^*,u} - r_{a_t,u}) \tag{1}$$

where a_t^* is the best arm to be presented to the user according to the oracle strategy, $r_{a_t^*,u}$ is the corresponding payoff, and R_t is one-step regret at trial t.

In a standard contextual bandit problem, the payoffs of each arm with respect to different users are assumed to be governed by a conjecture of unknown bandit parameters and given context vector of the arm. To simplify our discussions, linear dependency is postulated, but it can be readily extended to more complicated dependency structures, such as generalized linear models [10]. Specifically, each user u is assumed to be associated with an unknown preference parameter $\boldsymbol{\theta}_u \in \mathbb{R}^d$. This preference parameter, together with the given arm's context vector $\mathbf{x}_a \in \mathbb{R}^d$, determine the payoff of a_t by $r_{a_t,u} = \mathbf{x}_{a_t,u}^{\mathsf{T}} \boldsymbol{\theta}_u + \eta_t$, where the random noise η_t is drawn from a zero-mean Gaussian distribution $N(0, \sigma^2)$.

As we discussed in the introduction, one important assumption made in the above reward generation process is that the context features $\mathbf{x}_a$ revealed to the learner are sufficiently informative to capture the entire reward generation process. In other words, the stochastic game between the environment and the learner is assumed to be transparent. However, this assumption is oversimplified. In many real-world applications, it is challenging to exhaust all relevant features ahead of time, and oftentimes because of privacy or sampling constraints many important factors are unobservable to the algorithm. Due to the existence of those hidden factors, the stochastic game is no longer transparent to the learner: the optimal choice is made by the environment according to the whole feature set (both observable and hidden features), while the learner can only learn from the observed features. When these two types of features are independent, the learner's inconsistent knowledge about reward generation will cause systematic bias at every step during online learning, and eventually lead to a linearly increasing divergency from optimality over time. This limitation motivates us to introduce hidden feature learning into contextual bandits.

We generalize the linear contextual bandits by introducing the concept of hidden features. We assume that in addition to the observed contextual features, there is also a set of hidden features that affect the expected payoffs. This can be formalized as,

$$r_{a_t,u} = (\mathbf{x}_{a_t}, \mathbf{v}_{a_t})^{\mathsf{T}} (\boldsymbol{\theta}_u^{\mathbf{x}}, \boldsymbol{\theta}_u^{\mathbf{v}}) + \eta_t \tag{2}$$

where $\mathbf{x}_{a_t} \in \mathbb{R}^d$ and $\mathbf{v}_{a_t} \in \mathbb{R}^l$ (with $\|(\mathbf{x}_a, \mathbf{v}_a)\|_2 \leq L$) are the observed and hidden features of item a_t, and $\boldsymbol{\theta}_u^{\mathbf{x}}$ and $\boldsymbol{\theta}_u^{\mathbf{v}}$ are the corresponding bandit parameters. We denote $\boldsymbol{\theta}_u = (\boldsymbol{\theta}_u^{\mathbf{x}}, \boldsymbol{\theta}_u^{\mathbf{v}}) \in \mathbb{R}^{d+l}$ as the unknown preference parameters of user u (with $\|\boldsymbol{\theta}_u\|_2 \leq S$). We assume the dimension l of hidden features is known to the learner ahead of time, and we will discuss its impact to the algorithm in our regret analysis.

The reward generation assumption specified in Eq (2) differentiates our bandit problem from existing ones. Because only $\mathbf{x}_{a_t}$ will be disclosed to the learner for arm selection, the residual between the true reward and the learner's estimate no longer has a zero mean (as assumed in most linear contextual bandit algorithms [15, 1, 10]). Instead, the residual of reward estimation is constantly shifted by $\mathbf{v}_{a_t}^{\mathsf{T}} \boldsymbol{\theta}_u^{\mathbf{v}}$, which is unlikely to be recovered from $\mathbf{x}_{a_t}$ when $\mathbf{v}_{a_t}$ and $\mathbf{x}_{a_t}$ are independent. This becomes the source of linearly increased regret in conventional contextual bandits, if $\mathbf{v}_{a_t}$ is not properly modeled.

Due to the coupling between $\boldsymbol{\theta}_u$ and $\mathbf{v}_a$ in the reward generation postulated in Eq (2), we appeal to a coordinate decent algorithm built on ridge regression to estimate the unknown bandit parameter $\boldsymbol{\theta}_u$ for each user and the unknown hidden feature $\mathbf{v}_a$ for each item. Specifically, the objective function of ridge regression can be written as follows,

$$\min_{\boldsymbol{\theta}_u, \mathbf{v}_a} \frac{1}{2} \sum_{t=1}^{T} \Big((\mathbf{x}_{a_t}, \mathbf{v}_{a_t})^{\mathsf{T}} \boldsymbol{\theta}_u - r_{a_t,u} \Big)^2 + \frac{\lambda_1}{2} \|\boldsymbol{\theta}_u\|_2 + \frac{\lambda_2}{2} \|\mathbf{v}_a\|_2 \tag{3}$$

where λ_1 and λ_2 are the trade-off parameters for L2 regularization. We should note the regularization is critical in our solution in two folds. First, it makes the subproblems in coordinate decent well-posed, so that we have closed form solutions for $\boldsymbol{\theta}_u$ and $\mathbf{v}_a$ at each iteration. Second, it helps remove the scaling indeterminacy between the estimate of $\boldsymbol{\theta}_u$ and $\mathbf{v}_a$, and makes the q-linear convergence rate of parameter estimation possible [23].

The closed-form estimation of $\boldsymbol{\theta}_u$ and $\mathbf{v}_a$ with respect to Eq (3) at trial t can be achieved by $\hat{\boldsymbol{\theta}}_{u,t} = \mathbf{A}_{u,t}^{-1} \mathbf{b}_{u,t}$ and $\hat{\mathbf{v}}_{a,t} = \mathbf{C}_{a,t}^{-1} \mathbf{d}_{a,t}$, in which,

$$\mathbf{A}_{u,t} = \lambda_1 \mathbf{I}_1 + \sum_{t'=1}^{t} (\mathbf{x}_{a_{t'}}, \hat{\mathbf{v}}_{a_{t'}}) (\mathbf{x}_{a_{t'}}, \hat{\mathbf{v}}_{a_{t'}})^{\mathsf{T}}$$

$$\mathbf{b}_{u,t} = \sum_{t'=1}^{t} (\mathbf{x}_{a_{t'}}, \hat{\mathbf{v}}_{a_{t'}}) r_{a_{t'},u}$$

$$\mathbf{C}_{a,t} = \lambda_2 \mathbf{I}_2 + \sum_{t'=1}^{t} \hat{\boldsymbol{\theta}}_{u,t'}^{\mathbf{v}} \hat{\boldsymbol{\theta}}_{u,t'}^{\mathbf{v}\mathsf{T}}$$

$$\mathbf{d}_{a,t} = \sum_{t'=1}^{t} \hat{\boldsymbol{\theta}}_{u,t'}^{\mathbf{v}} \big(r_{a_{t'},u} - \mathbf{x}_{a_{t'}}^{\mathsf{T}} \hat{\boldsymbol{\theta}}_{u,t'}^{\mathbf{x}} \big)$$

$\mathbf{I}_1$ and $\mathbf{I}_2$ are two identity matrices with dimensions of $(d + l) \times (d + l)$ and $l \times l$ respectively. Projection of the estimated $\hat{\boldsymbol{\theta}}_{u,t}$ and $\hat{\mathbf{v}}_{a,t}$ is necessary to satisfy the constraint on their L2 norms, i.e., $\|\boldsymbol{\theta}_u\|_2 \leq S$ and $\|(\mathbf{x}_a, \mathbf{v}_a)\|_2 \leq L$.

The estimated bandit parameters and hidden features give us a reasonable prediction of the expected payoff from user u to arm a by $\hat{r}_{a_t,u} = (\mathbf{x}_a, \hat{\mathbf{v}}_{a,t})^{\mathsf{T}} \hat{\boldsymbol{\theta}}_{u,t}$. But such predicted payoff might not be accurate when one does not have sufficient observations for parameter estimation at an early stage. Proper exploration of less promising items is thus necessary to balance the explore/exploit

trade-off for long-term optimality. Upper Confidence Bound (UCB) [3, 4] has proved to be an effective strategy, which uses the estimation confidence of predicted payoffs on the selected arms for exploration. In our solution, the uncertainty of reward estimation during online update comes from two aspects: the estimation uncertainty of true bandit parameters, i.e., $\|\hat{\boldsymbol{\theta}}_{u,t} - \boldsymbol{\theta}^*_u\|$, and that of hidden features, i.e., $\|\hat{\mathbf{v}}_{a,t} - \mathbf{v}^*_a\|$. Because of the closed form solution in our coordinate descend estimation, the confidence set of $\hat{\boldsymbol{\theta}}_{u,t}$ and $\hat{\mathbf{v}}_{a,t}$ can be analytically computed by the following lemma,

LEMMA 1. *When the Hessian matrix of the objective function defined in Eq* (3) *is positive definite at the optimizer $\boldsymbol{\theta}^*_u$ and $\mathbf{v}^*_a$, with proper initialization, for any $\epsilon_1 > 0, \epsilon_2 > 0$, and for any $\delta > 0$, with probability at least $1-\delta$, the estimation error of bandit parameters and hidden features from coordinate descent satisfies,*

$$\|\hat{\boldsymbol{\theta}}_{u,t} - \boldsymbol{\theta}^*_u\|_{\mathbf{A}_{u,t}} \leq \sqrt{d \ln\big(\frac{\lambda_1 d + tL^2}{\lambda_1 d \delta}\big)} + \sqrt{\lambda_1} S + \frac{2}{\sqrt{\lambda_1}} \frac{(q_1+\epsilon_1)(1-(q_1+\epsilon_1)^t)}{1-(q_1+\epsilon_1)} \tag{4}$$

$$\|\hat{\mathbf{v}}_{a,t} - \mathbf{v}^*_a\|_{\mathbf{C}_{a,t}} \leq \sqrt{l \ln\big(\frac{\lambda_2 l + tS^2}{\lambda_2 l \delta}\big)} + \sqrt{\lambda_2} L + \frac{2}{\sqrt{\lambda_2}} \frac{(q_2+\epsilon_2)(1-(q_2+\epsilon_2)^t)}{1-(q_2+\epsilon_2)} \tag{5}$$

in which $0 < q_1 < 1$ and $0 < q_2 < 1$.

Lemma 1 gives us a reasonably tight construction of confidence sets for $\hat{\boldsymbol{\theta}}_{u,t}$ and $\hat{\mathbf{v}}_{a,t}$, which can be easily transformed to the estimation uncertainty of payoff $\hat{r}_{a_t,u}$. The detailed proof of this lemma and derivations for the estimation uncertainty of payoff $\hat{r}_{a_t,u}$ can be found in the Appendix.

Based on Lemma 1, we define α^u_t and α^a_t as the upper bound of $\|\hat{\boldsymbol{\theta}}_{u,t} - \boldsymbol{\theta}^*_u\|_{\mathbf{A}_{u,t}}$ and $\|\hat{\mathbf{v}}_{a,t} - \mathbf{v}^*_a\|_{\mathbf{C}_{a,t}}$ respectively, and design the following arm selection strategy for our online learning,

$$a_t = \arg\max_{a \in \mathcal{A}} \Big((\mathbf{x}_a, \hat{\mathbf{v}}_{a,t})^\intercal \hat{\boldsymbol{\theta}}_{u,t} + \alpha^u_t \sqrt{(\mathbf{x}_a, \hat{\mathbf{v}}_{a,t}) \mathbf{A}^{-1}_{u,t} (\mathbf{x}_a, \hat{\mathbf{v}}_{a,t})^\intercal} + \alpha^a_t \sqrt{\hat{\boldsymbol{\theta}}^{\mathbf{v}}_{u,t} \mathbf{C}^{-1}_{a,t} \hat{\boldsymbol{\theta}}^{\mathbf{v}\intercal}_{u,t}} \Big) \tag{6}$$

The first term in Eq (6) is the predicted payoff of arm a_t to user u based on the current estimation of bandit parameters and latent features. This estimate reflects the tendency for exploitation of currently promising arms. The second and third terms are related to the estimation uncertainty of $\boldsymbol{\theta}_u$ and $\mathbf{v}_a$, which reflect the tendency for exploration of currently less promising arms. It is easy to verify that the exploration terms shrink when more observations become available, such that the exploit/explore trade-off is balanced by the payoff prediction confidence.

By comparing our arm selection strategy to those in other contextual bandit algorithms, e.g., LinUCB [15] and GLM-UCB [10], we can find that our algorithm considers not only the prediction confidence of user preference parameters (the second term), but also the confidence of learnt hidden features (the third term). Therefore ours is a more general solution for contextual bandits: when hidden features do not exist, i.e., $\|\mathbf{v}^*_a\| = 0$, our algorithm degenerates to those conventional contextual bandit algorithms (since $\alpha^a_t = 0$). Most importantly, with proper initialization of coordinate descent, our algorithm guarantees a sublinear regret with high probability. We rigorously prove this conclusion in Section 3.2. Besides, although the UCB-like algorithm developed by Nakamura [19] also used an index that depends on the covariance matrices of the posterior distributions of both latent user and item vectors, no theoretical regret analysis is provided in their solution.

Algorithm 1 Online learning with hLinUCB

1: **Inputs:** $\lambda_1, \lambda_2 \in (0, +\infty)$, $l \in \mathbb{Z}^+$
2: **for** $t = 1$ to T **do**
3: Receive user u
4: **if** user u is new **then**
5: initialize $\mathbf{A}_{u,t} \leftarrow \lambda_1 \mathbf{I}$, $\mathbf{b}_{u,t} \leftarrow \mathbf{0}$, $\hat{\boldsymbol{\theta}}_{u,t} \leftarrow \mathbf{0}$
6: **end if**
7: Observe feature vectors, $\mathbf{x}_a \in \mathbb{R}^d$
8: For $\forall a \in \mathcal{A}$
9: **if** item a is new **then**
10: initialize $\mathbf{C}_{a,t} \leftarrow \lambda_2 \mathbf{I}$, $\mathbf{d}_{a,t} \leftarrow \mathbf{0}$, $\hat{\mathbf{v}}_{a,t} \leftarrow \mathbf{0}$
11: **end if**
12: Select action by $a_t = \arg\max_{a \in \mathcal{A}} \Big((\mathbf{x}_a, \hat{\mathbf{v}}_{a,t})^\intercal \hat{\boldsymbol{\theta}}_{u,t} + \alpha^u \sqrt{(\mathbf{x}_a, \hat{\mathbf{v}}_{a,t}) \mathbf{A}^{-1}_{u,t} (\mathbf{x}_a, \hat{\mathbf{v}}_{a,t})^\intercal} + \alpha^a \sqrt{\hat{\boldsymbol{\theta}}^{\mathbf{v}}_{u,t} \mathbf{C}^{-1}_{a,t} \hat{\boldsymbol{\theta}}^{\mathbf{v}\intercal}_{u,t}} \Big)$
13: Observe payoff $r_{a_t,u}$ from user u
14: $\mathbf{A}_{u,t+1} \leftarrow \mathbf{A}_{u,t} + (\mathbf{x}_{a_t}, \hat{\mathbf{v}}_{a_t,t})(\mathbf{x}_{a_t}, \hat{\mathbf{v}}_{a_t,t})^\intercal$
15: $\mathbf{b}_{u,t+1} \leftarrow \mathbf{b}_{u,t} + (\mathbf{x}_{a_t}, \hat{\mathbf{v}}_{a_t,t}) r_{a_t,u}$
16: $\hat{\boldsymbol{\theta}}_{u,t+1} \leftarrow \mathbf{A}^{-1}_{u,t+1} \mathbf{b}_{u,t+1}$
17: $\mathbf{C}_{a_t,t+1} \leftarrow \mathbf{C}_{a_t,t} + \hat{\boldsymbol{\theta}}^{\mathbf{v}}_{u,t} \hat{\boldsymbol{\theta}}^{\mathbf{v}\intercal}_{u,t}$
18: $\mathbf{d}_{a_t,t+1} \leftarrow \mathbf{d}_{a_t,t} + \hat{\boldsymbol{\theta}}^{\mathbf{v}}_{u,t} (r_{a_t,u} - \mathbf{x}^\intercal_{a_t} \hat{\boldsymbol{\theta}}^{\mathbf{x}}_{u,t})$
19: $\hat{\mathbf{v}}_{a_t,t+1} \leftarrow \mathbf{C}^{-1}_{a_t,t+1} \mathbf{d}_{a_t,t+1}$
20: Project $\hat{\boldsymbol{\theta}}_{u,t+1}$ and $\hat{\mathbf{v}}_{a_t,t+1}$ with respect to the constraint $\|\boldsymbol{\theta}_u\|_2 \leq S$ and $\|(\mathbf{x}_a, \mathbf{v}_a)\|_2 \leq L$.
21: **end for**

We name this resulting bandit algorithm as Hidden LinUCB, or hLinUCB in short, and illustrate the detailed procedure of it in Algorithm 1. This algorithm has several important properties that are worth mentioning. First, its computational complexity is linear with respect to the number of arms and users, and is at most cubic to the number of features (because of matrix inverse in step 16 and 19). This can be further reduced to quadratic by using the Sherman-Morrison formula for matrix inverse, due to the special structure of $\mathbf{A}_{u,t}$ and $\mathbf{C}_{a,t}$ matrices. Second, because the exploration terms in Eq (6) are shrinking when more observations are available, coordinate descent can be performed in a mini-batch mode with adaptive window sizes for different users and items. Specifically, instead of updating all the parameters across users and items in every iteration, we can keep accumulating $(\mathbf{A}_{u,t}, \mathbf{b}_{u,t}, \mathbf{C}_{a_t,t}, \mathbf{d}_{a_t,t})$, but compute $\hat{\boldsymbol{\theta}}_{u,t} = \mathbf{A}^{-1}_{u,t} \mathbf{b}_{u,t}$ and $\hat{\mathbf{v}}_{a_t,t} = \mathbf{C}^{-1}_{a_t,t} \mathbf{d}_{a_t,t}$ with a reduced frequency. The window size can be adaptively decided by the estimation confidence of $\boldsymbol{\theta}_{u,t}$ and $\mathbf{v}_{a_t,t}$ over time. We prove that such postponed model update will not change the order of resulting upper regret bound, but would greatly reduce the computation complexity of our algorithm in large-scale deployment. Last but not least, because of the feature-based reward generation assumption, our bandit algorithm does not need all arms to be played at least once. This can be understood by the regression-based hidden feature learning and parameter estimation in step 14 to 19 in Algorithm 1, which enable information sharing across arms. Therefore, the resulting regret bound will be only determined by the complexity of observed and hidden features, rather than the number of arms. Detailed regret analysis in the next section supports this property.

3.2 Regret Analysis

In this section, we provide detailed regret analysis of our hLinUCB algorithm and compare it with other conventional contextual bandit algorithms.

According to our derivation (details can be found in the proof of Lemma 1 in Appendix), the coordinate descent based parameter estimation in our algorithm satisfy the following two inequalities

and they directly contribute to the final regret of hLinUCB,

$$\|\hat{\boldsymbol{\theta}}_{u,t} - \boldsymbol{\theta}^*_u\|_{\mathbf{A}_{u,t}} \leq \|\sum_{t'=1}^{t}(\mathbf{x}_{a_{t'}}, \hat{\mathbf{v}}_{a_{t'},t'})\eta_{t'}\|_{\mathbf{A}^{-1}_{u,t}} + \sqrt{\lambda_2}L \quad (7)$$
$$+ \frac{LS}{\sqrt{\lambda_1}}\sum_{t'=1}^{t}\|\mathbf{v}^*_{a_{t'}} - \hat{\mathbf{v}}_{a_{t'},t'}\|_2$$

$$\|\hat{\mathbf{v}}_{a,t} - \mathbf{v}^*_a\|_{\mathbf{C}_{a,t}} \leq \|\sum_{t'=1}^{t}\hat{\boldsymbol{\theta}}^{\mathbf{v}}_{u,t'}\eta_{t'}\|_{\mathbf{C}^{-1}_{a,t}} + \sqrt{\lambda_1}S \quad (8)$$
$$+ \frac{LS}{\sqrt{\lambda_2}}\sum_{t'=1}^{t}\|\boldsymbol{\theta}^{*\mathbf{v}}_u - \hat{\boldsymbol{\theta}}^{\mathbf{v}}_{u,t'}\|_2$$

The first terms on the right-hand side of Eq (7) and (8) have a sublinear bound with respect to the time index t due to the property of self-normalized vector-valued martingales [1], since both $(\mathbf{x}_{a_{t'}}, \hat{\mathbf{v}}_{a_{t'},t'})$ and $\hat{\boldsymbol{\theta}}^{\mathbf{v}}_{u,t'}$ have bounded L2 norms and η_t has a finite variance. Hence the estimation quality of $\boldsymbol{\theta}_u$ and $\mathbf{v}_a$ (and therefore the regret of hLinUCB) depends on the two summation terms of $\sum_{t'=1}^{t}\|\mathbf{v}^*_{a_{t'}} - \hat{\mathbf{v}}_{a_{t'},t'}\|_2$ and $\sum_{t'=1}^{t}\|\boldsymbol{\theta}^{*\mathbf{v}}_u - \hat{\boldsymbol{\theta}}^{\mathbf{v}}_{u,t'}\|_2$ in Eq (7) and (8). Due to the nature of coordinate descend, the estimation of $\boldsymbol{\theta}_u$ and $\mathbf{v}_a$ depends on each other and is not necessarily convex. But if the regularization parameters λ_1 and λ_2 are sufficiently large, the Hessian matrix of Eq (3) will be positive definite at the optimizer. As a result, based on the proved convergence property of alternating least square in [23], the estimation of $\boldsymbol{\theta}^{\mathbf{v}}_u$ and $\mathbf{v}_a$ is q-linearly convergent to the optimum $(\boldsymbol{\theta}^{\mathbf{v}*}_u, \mathbf{v}^*_a)$. More specifically, with proper initialization, the last terms of Eq (7) and (8) are bounded by the summation of a geometric sequence with a common ratio smaller than one (details are provided in the Appendix), such that they are also sublinear with respect to the time index t. These properties are essential to prove Lemma 1, and lead to the proof of upper regret bound of hLinUCB.

Formally, based on the Lemma 1 discussed in Section 3.1 and our arm selection strategy defined in Eq (6), Theorem 1 gives a tight upper regret bound of hLinUCB.

THEOREM 1. *Under proper initialization of coordinate descent, with probability at least $1-\delta$, the cumulated regret of Hidden Linear Bandit algorithm satisfies,*

$$\boldsymbol{R}(T) \leq 2\alpha^u_T\sqrt{2dT\ln(1+\frac{TL^2}{\lambda_1 d})} + 2\alpha^a_T\sqrt{2lT\ln(1+\frac{TS^2}{\lambda_2 l})}$$
$$+ 2\alpha^a_T\frac{(q_2+\epsilon_2)(1-(q_2+\epsilon_2)^T)}{1-(q_2+\epsilon_2)}$$

in which q_2 and ϵ_2 are the same as those defined in Lemma 1, α^u_T and α^a_T are the upper bound of $\|\hat{\boldsymbol{\theta}}_{u,t} - \boldsymbol{\theta}^*_u\|_{\mathbf{A}_{u,t}}$ and $\|\hat{\mathbf{v}}_{a,t} - \mathbf{v}^*_a\|_{\mathbf{C}_{a,t}}$, and δ is embedded in α^u_T and α^a_T. Roughly speaking, Theorem 1 indicates that the upper regret bound of hLinUCB is $O\big(\sqrt{T}\ln T + (1-c^T)\sqrt{T\ln T}\big)$, in which c is a constant between 0 and 1. This regret bound is in the same order of a typical contextual bandit algorithm [15, 1, 10], i.e., $O\big(\sqrt{T}\ln T\big)$. The detailed proof of this theorem is provided in the Appendix. We should note that our regret analysis assumes proper initialization of coordinate descent. We empirically evaluated the sensitivity of hLinUCB with respect to the initialization in our experiments and found it was quite robust to different initializations in practice.

It is necessary to compare the resulting regret bound of hLinUCB to those of conventional contextual bandit algorithms so as to understand the theoretical advantage of the proposed solution. Take LinUCB [15] as an example, since it is a typical and popularly used linear bandit algorithm. The regret bound of LinUCB in this reward generation environment also depends on the right-hand side of Eq (7) and (8). The first term on the right-hand side of Eq (7) in LinUCB can be bounded similarly as that in our algorithm; however, since LinUCB does not model the hidden features (i.e., set $\hat{\mathbf{v}}_{a_t,t}$ to $\mathbf{0}$), the estimation of $\boldsymbol{\theta}_u$ in Eq (7) will always encounter a constantly increasing term $\sum_{t'=1}^{t}\|\mathbf{v}^*_{a_{t'}}\|_2$ over time. When the hidden features are important in determining the payoffs and independent from the observable features, such a constant is not negligible nor recoverable, and therefore a linear upper regret bound is inevitable. The same conclusion applies to other contextual bandit algorithms that cannot explicitly model the hidden features.

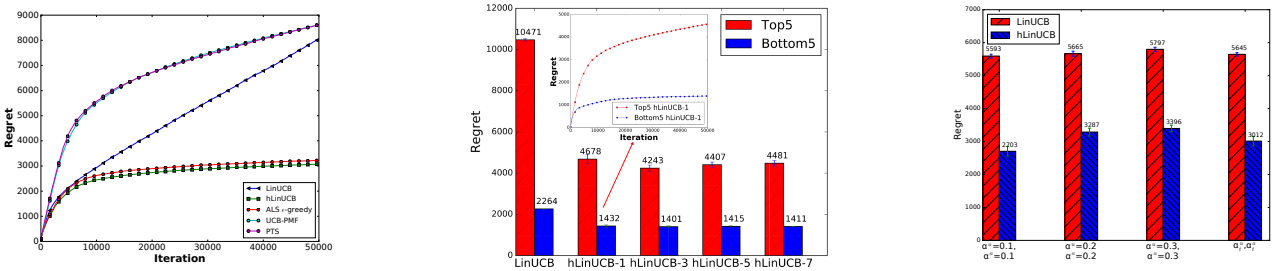
A similar linear upper regret bound conclusion applies to our algorithm when we do not know the dimension of latent features. For example, if we set fewer dimensions in $\hat{\mathbf{v}}_a$ and features in $\mathbf{v}^*_a$ are linearly independent from each other, constant estimation error is unavoidable. One possible solution is to increase the dimension of learnt hidden features, which potentially require more observations during online update. In our empirical evaluations, we investigated the effect of the latent feature dimensions on the algorithm's practical performance. In addition, as we discussed in Section 3.1, to improve computational efficiency in practice, the coordinate descent between $\hat{\boldsymbol{\theta}}_{u,t}$ and $\hat{\mathbf{v}}_{a_t,t}$ can be performed in a mini-batch mode without hurting the algorithm's regret bound. It is because of two facts: 1) the first terms on the right-hand side of Eq (7) and (8) are sublinear to t independently from coordinate descent; 2) the q-linear convergence property still holds whenever an update happens. For example, if the update happens in every M iterations, we will have the q-convergence in every M steps for $\boldsymbol{\theta}_u$ and $\mathbf{v}_a$ estimation. It makes the term $\sum_{t=1}^{T}\|\hat{\boldsymbol{\theta}}_{u,t} - \boldsymbol{\theta}^*_u\|$ be bounded by $M\frac{(q_2+\epsilon_2)\left(1-(q_2+\epsilon_2)^{T/M}\right)}{1-(q_2+\epsilon_2)}$ instead. This can be similarly applied to the summation term in α^u_T and α^a_T for controlling the arm selection. Therefore, as long as M is not too large, we can still make the regret bound stay in the same order as that with real-time update but considerably reduce the computational complexity.

4. EXPERIMENTS

We performed empirical evaluations of our proposed hLinUCB algorithm against several related baseline algorithms, including: 1) two contextual bandit algorithms: LinUCB and hybrid-LinUCB with user features [15]; 2) three bandit-based online collaborative filtering methods: Particle Thompson sampling for Matrix Factorization (PTS) [12], UCB-probabilistic matrix factorization (UCB-PMF) [19], and Alternating Least Square ϵ-greedy, which is a generalized version of linear ϵ-greedy introduced in [27]. Below we provide a brief discussion of all the compared baselines.

- LinUCB: it selects an arm based on an upper confidence bound of the estimated reward with given context vectors. LinUCB only works with the observed features and does not consider the hidden features.
- hybrid-LinUCB: it extends LinUCB via a hybrid feature representation of both items and users.
- PTS: it uses Thompson Sampling for arm selection in online probabilistic matrix factorization.
- UCB-PMF: it is an probabilistic matrix factorization based algorithm using a UCB-like strategy to balance exploration and exploitation during online learning.
- ALS ϵ-greedy: it is a generalized linear ϵ-greedy algorithm considering both observed and hidden features. Alternating Least Square (ALS) is used to estimate hidden features, and ϵ is decaying over time t (set to $\frac{\alpha}{t}$).

We tested all the algorithms on a synthetic data set via simulation, a large collection of click stream from Yahoo Today Module



(a) Convergence of cumulated regret (b) The effect of unmatched hidden feature dimension (c) The effect of α

Figure 1: Analysis of regret, hidden feature dimension and parameter tuning.

dataset [15], and two real-world datasets extracted from the social bookmarking web service Delicious and music streaming service LastFM [7]. Extensive experiment comparisons confirmed the necessity of learning hidden features in contextual bandit algorithms. And the experiment results also validated the theoretical analysis of hLinUCB algorithm and other contextual bandit algorithms in an environment with hidden features in reward generation.

4.1 Experiments on synthetic dataset

In this experiment, we compared all the online learning algorithms based on simulations, and used the accumulated regret as our performance metric in comparison.

4.1.1 Simulation Settings

In simulation, we first generate a size-K arm pool $\mathcal{A}$, in which each arm a is associated with a $(d+l)$-dimensional feature vector $(\mathbf{x}_a, \mathbf{v}_a)$. Each dimension is drawn from a set of zero-mean Gaussian distributions with variances sampled from a uniform distribution $U(0,1)$. Principle Component Analysis (PCA) is performed to make all the dimensions orthogonal to each other. To simulate the reward generation with hidden features specified in Eq (2), we use all the $(d+l)$-dimensional features to compute the true reward for each arm, while only revealing d dimension of the features (i.e., $\mathbf{x}_a$) to a learning algorithm. We then simulate N users, each of who is associated with a $(d+l)$-dimensional parameter vector $\boldsymbol{\theta}^*_u$. Each dimension of $\boldsymbol{\theta}^*_u$ is drawn from a uniform distribution $U(0,1)$. $\boldsymbol{\theta}^*_u$ is treated as the ground-truth preference parameter for reward generation, and is unknown to the algorithms. To increase the learning complexity, at each trial t, our simulator only discloses a subset of arms in $\mathcal{A}$ to the learners for selection, e.g., randomly select 10 arms from $\mathcal{A}$ without replacement. The ground-truth payoff $r_{a,u}$ is corrupted by a Gaussian noise $\eta = N(0, \sigma^2)$ before feeding back to the learners. In particular, at each trial t, the same set of arms are presented to all the algorithms; and the Gaussian noise η_t is sampled once for all those arms at trial t.

Under this simulation setting, we compared LinUCB, PTS, ALS ϵ-greedy, UCB-PMF and our proposed hLinUCB algorithm. Because our simulator does not generate user features, hybrid-LinUCB is not applicable in this experiment. In simulation, features are randomly split into observable part and hidden part aprior. We fixed the dimension d of observable features to 20, the dimension l of hidden feature to 5, user size N to 100, the standard derivation σ of Gaussian noise to 0.1, and the arm pool size K to 1000 in simulation. We set the latent feature dimension in PTS and UCB-PMF to 10, and that in ALS ϵ-greedy to 5.

4.1.2 Results & Analysis

All algorithms were executed up to 50,000 iterations in simulation. Cumulated regret defined in Eq (1) is used to evaluate the performance of different algorithms as shown in Figure 1 (a). We first fixed α^u_t and α^a_t to 0.1, set hidden feature dimension l to 5, and fix the two trade-off parameters λ_1 and λ_2 of $L2$ regularization to 0.1 in hLinUCB.

From the cumulated regret shown in Figure 1 (a), we can clearly notice that when the payoff is governed by the hidden features in addition to the observed ones, LinUCB suffers from a linearly increasing cumulated regret, while our hLinUCB and ALS ϵ-greedy both converged quickly. This verifies our motivation of learning hidden features in contextual bandits, and validates the conclusion from our regret analysis that in an environment which has hidden features in reward generation, failing to model such features will lead to a linearly increased regret. We also observed that PTS and UCB-PMF needed much more iterations to converge comparing to hLinUCB. Because these two baselines cannot utilize the observed contextual features in reward estimation, they require much more observations to reduce reward prediction uncertainty (i.e., explore more). This further validates the necessity of combining both observed and hidden features in bandit learning. We also varied initialization of $\boldsymbol{\theta}^u$ and $\mathbf{v}_a$ under different zero-mean Gaussian distributions with the standard deviation ranging from 0.1 to 2.0. In all cases, hLinUCB converged sublinearly and the standard deviation of the resulting cumulated regret stayed within 10% of average cumulated regret. This indicates hLinUCB is robust to initialization.

Because hLinUCB requires the dimension of hidden features as input, we test its sensitivity to the setting of hidden dimension l with simulation. In this experiment, the dimension of ground-truth hidden features in the simulator is fixed to 5 and the dimension of hidden features used in hLinUCB varies from 0 to 7. In such a setting, LinUCB becomes a special case of hLinUCB when the dimension of hidden features is 0. To investigate the importance of hidden features, we tested two different ways of hidden feature construction in our simulator: 1) we chose the top 5 features with largest eigenvalue from PCA's result as hidden features, i.e., we hid the top 5 most informative features in reward generation from the learners; 2) we hid the bottom 5 most informative features. From the results shown in Figure 1 (b), we can reach three conclusions. First, when the hidden features are the most informative ones, we obtain much worse regret than that in the case of the least informative features are hidden. This explains the importance of modeling hidden features in a bandit algorithm, especially when they are crucial in reward generation. This result is also expected based on our regret analysis. Second, the large difference between the regret of an algorithm that does not model the hidden features (such as LinUCB) and the one that models hidden features (even with wrong dimensions) emphasizes the necessity of hidden feature learning. Third, although our theoretical analysis predicts a linear regret in hLinUCB if the hidden feature dimension is not accurately set, the actual performance is much more promising. Detailed convergence trace can be found in the embedded subplot in Figure 1 (b). The major reason is that our analysis estimates the upper regret bound;

when the hidden feature dimension is set close to the ground-truth, the coefficient in front of the linear term is small and satisfactory online learning performance is still achievable.

In addition, we also investigated the effect of exploration parameter α_t^u and α_t^a in hLinUCB, compared with LinUCB. In Figure 1 (c), each column of the bar plot illustrates a combination of α_t^u and α_t^a used in hLinUCB (LinUCB uses the same setting of α_t^u). The last column indexed by (α_t^u, α_t^a) represents the theoretical settings of those two parameters computed from the algorithms' corresponding regret analysis. As shown in the results, the empirically tuned (α^u, α^a) yields comparable performance to the theoretical values, and makes online computation more efficient. As a result, in all our following experiments we will manually set α_t^u and α_t^a.

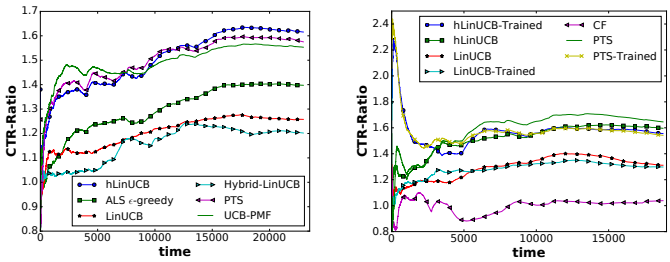
4.2 Experiments on Yahoo Today Module

In this experiment, we compared our hLinUCB algorithm with all baselines on a large-scale clickstream dataset made available by the Yahoo Webscope program[1]. This data set contains 45,811,883 user visits to Yahoo Today Module in a ten-day period in May 2009. For each visit, both the user and each of the 10 candidate articles are associated with a feature vector of six dimensions (including a constant bias feature), constructed by a conjoint analysis with a bilinear model [15]. Due to privacy constraints, only these features are available but the meaning of them is unknown. This provides us an ideal testbed to assess the value of hidden feature learning for bandit algorithms in practice. Besides, there is no user identity in this data set, which forbids us to associate the observations with individual users. To address this limitation, we first clustered all users into user groups by applying k-means algorithm on the given user features. Each observation is then assigned to its closest user group. All algorithms were executed on these identified user groups. The hidden feature dimension in hLinUCB and ALS ϵ-greedy was set to 5, and in UCB-PMF and PTS to 10.

In this experiment, the unbiased offline evaluation protocol proposed in [16] was used to compare different algorithms. Click-through-rate (CTR), which is defined as the ratio between the number of clicks an algorithm receives and the number of recommendations it makes, was used to evaluate the performance of all bandit algorithms. Average CTR (*not* the cumulated CTR) is computed in every 2000 observations for each algorithm as the performance metric. Following the same evaluation principle used in [15], we normalized the resulting CTR from different algorithms by the corresponding logged random strategy's CTR. We report the normalized CTR results from different algorithms over 160 derived user groups in Figure 2 (a). We also tested all algorithms with different number of derived user groups from k-means (from 40 to 160), and similar relative comparison results were obtained.

From Figure 2 (a) we can clearly find that hLinUCB achieved significant performance improvement comparing to other algorithms except PTS and UCB-PMF. Given these observed contextual features were originally used in Yahoo's Today Module deployment, the significant improvement from hLinUCB over Hybrid-LinUCB and LinUCB further supports the necessity of learning hidden features for contextual bandits. Compared with ALS ϵ-greedy, which uses a context-free exploration strategy, the estimation confidence based exploration strategy employed in hLinUCB leads to an improved balance between explore and exploit during online learning. On this dataset, hLinUCB outperformed PTS and UCB-PMF after running over about 6 days' observations. The good performance of these two factorization-based methods is expected: according to our observations in the simulation-based experiments, the factorization-based methods need more training data to adjust its parameters, since they cannot leverage the observed features.

[1]https://webscope.sandbox.yahoo.com/



(a) Online learning only (b) With offline training

Figure 2: Average CTR-Ratio on Yahoo Dataset

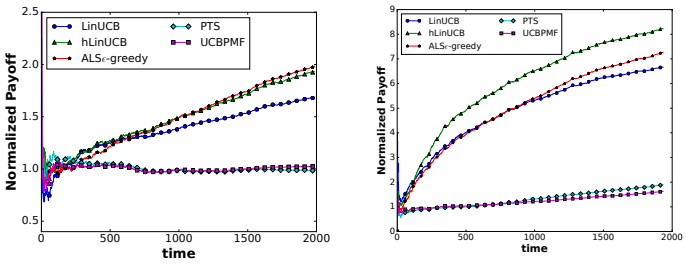
10 days' clickstream data provides them sufficient observations to accurately estimate the parameters for better performance.

To understand how our algorithm performs comparing to a traditional factorization-based collaborative filtering method in an online setting, we included a Stochastic Gradient Descent based matrix factorization method with periodic model update as a new baseline, and denote it as CF. We split the data chronologically into training and testing sets. In the training phase, both CF and bandit algorithms were fed with the first two days' observations for offline model estimation. In the testing phase, daily batch update was performed for the CF baseline. All the algorithms were evaluated on the clicks starting from the third day. To make the figure clear and readable, we only demonstrated the results of hLinUCB, LinUCB, PTS and CF in Figure 2 (b). In this experiment, all the bandit algorithms have two versions: with and without offline training. It is evident that CF with periodic model update does not work in online testing, because it only exploits the immediately promising items from an inaccurate model for parameter estimation. In other words, proper exploration of currently less promising items is necessary for long-term optimality. In addition, for all the bandit algorithms, the offline pre-trained models provide some immediate benefit in online testing. For example, the performance of PTS and hLinUCB boosted on the first testing day. But they all converged to their online trained counterparts quickly afterwards. This indicates online exploration alone is sufficient to guide the algorithms to reach satisfactory performance with enough observations. Besides, since we only used two days' data for offline training and new items for recommendation kept emerging in the logged data, the utility of pre-trained models rapidly diminishes as more new observations become available. This further urges us to perform hidden feature learning in an online fashion.

4.3 Experiments on LastFM & Delicious

The LastFM dataset was extracted from the music streaming service website Last.fm (http://www.last.fm), and the Delicious data set was extracted from the social bookmark sharing service website Delicious (https://delicious.com). These two datasets were created by the Information Retrieval group at the Autonomous University of Madrid for the HetRec 2011 workshop with the goal of investigating the usage of heterogeneous information in recommendation systems[2]. The LastFM dataset contains 1,892 users and 17,632 items (artists). We used the information of "listened artists" of each user to create payoffs of recommendation candidates: if a user listened to an artist at least once, the payoff is 1, otherwise 0. The Delicious dataset contains 1,861 users and 69,226 items (URLs). We generated the payoffs using the information about the bookmarked URLs for each user: the payoff is 1 if the user bookmarked a particular URL, otherwise 0. We should note that the Delicious dataset is

[2]Datasets and their full description is available at http://grouplens.org/datasets/hetrec-2011



(a) Delicious dataset (b) LastFM dataset

Figure 3: Normalized reward on Delicious & LastFM datasets

much sparser than the LastFM dataset in terms of observations per item: they contain about the same number of users, but the number of items in Delicious is almost four times larger than that in LastFM. Therefore, these two datasets provide us complementary evaluations of online recommendation in different scenarios.

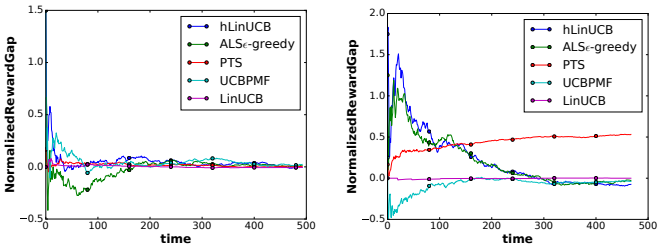
Following the same settings as in [7], we pre-processed these two datasets. First, we used all tags associated with a single item to create its TF-IDF feature vector. Then we used PCA to reduce the dimensionality of the features. In both datasets, we only took the first 25 principle components to construct the context vectors, i.e., the observed feature dimension $d = 25$. We then generated the candidate arm pool as follows: we fixed the size of candidate arm pool to 25; for a particular user u, we picked one item from those nonzero payoff items according to the whole observations in the dataset, and randomly picked the other 24 from those zero-payoff items. As a result, there is only one relevant recommendation in each arm pool. The latent dimension of hLinUCB and ALS ϵ-greedy was fixed to 5, and 10 in PTS and UCB-PMF.

We computed the *cumulated reward* for each algorithm for evaluation. To increase visibility of the demonstrated results, we normalized the cumulated reward in each algorithm by a random strategy's cumulated reward, and reported the average normalized cumulated reward in every 50 iterations. The experiment results are shown in Figure 3. Since these two datasets do not have user features, we excluded hybrid-LinUCB in this experiment.

As shown in Figure 3 (a), on Delicious dataset all algorithms performed very similarly at the beginning of online learning due to the sparse observations [7], and gradually ALS ϵ-greedy and our hLinUCB outperformed the other methods. Among all the bandit algorithms, PTS and UCB-PMF performed the worst, which were almost as bad as random. This observation verified our finding in simulation that PTS and UCB-PMF need more observations to estimate the hidden features for each item; when the observations are sparse, they suffer from inaccurate estimation of hidden features and bandit parameters. Again, due to the sparse observations many items only appeared once in a particular user, the estimation confidence based exploration strategy in hLinUCB did not lead to significant performance improvement against ALS ϵ-greedy, which uses a simple context-free exploration strategy. While on the LastFM dataset as shown in Figure 3 (b), where the observations are much more concentrated, we can clearly observe the advantage of modeling hidden features in hLinUCB against the other baselines. We also varied the initialization of hLinUCB and the setting of hidden feature dimensions on these two real-world datasets, and found it is again insensitive to initialization: the standard deviations of resulting cumulated regret were within 5% of the average cumulated regret on both datasets.

Another unique advantage of hLinUCB that is worth our attention is its ability to address *cold-start*, a serious challenge in online recommendation [21]. In hLinUCB, because the estimation of $\mathbf{v}_{a_t,t}$ depends on all the related users' estimated $\boldsymbol{\theta}_{u,t}$ (details can be found in Algorithm 1), user feedback on the recommended items is prorogated across users via the learned hidden features. As a result, users presented with overlapped items can benefit from each other's feedback in bandit parameter estimation. We evaluated this collaborative effect in hLinUCB and all other bandit algorithms on the LastFM and Delicious datasets. In this experiment, users were randomly separated into two groups denoted as U_1 and U_2, and items were also randomly split into two groups denoted as V_1 and V_2. We reserved the observations in the user-item combination of (U_2, V_2) as our testing set (i.e., also known as out of matrix recommendation). Then we created two training sets: one was constructed from the observations in the combinations of (U_1, V_1) and (U_2, V_1), and another set is based on the observations in the combinations of (U_1, V_1), (U_2, V_1) and (U_1, V_2). The difference between these two training sets is: in the first training set no information is disclosed about the items in group V_2; but in the second training set, the hidden features of items in V_2 learnt from the interactions with users in U_1 will help recommend items from V_2 to users in U_2. Intuitively, a bandit algorithm with explicit modeling of hidden features should benefit from information propagation in the chain of $U_2 \to V_1 \to U_1 \to V_2$; and therefore, the recommendations in the testing set of (U_2, V_2) become a *warm-start*.

We computed the improvement of different bandit algorithms' cumulated reward on these two training sets (*warm-start* v.s., *cold-start*), and reported the results in Figure 4. We can clearly notice that the difference in LinUCB between these two training settings is zero all the time. This is expected because in LinUCB users maintain their own bandit models and nothing is shared across users. However, in hLinUCB and ALS ϵ-greedy that explicitly estimate the hidden features, we can observe clear utility of information prorogation in alleviating cold-start. The improvement was considerably large at the beginning, especially on the LastFM dataset, and then gradually converged to zero. The large improvement at the beginning of online learning reveals the benefit of collaboration via the learnt hidden features. Although users in U_2 did not interact with any item in V_2 before, the observations in (U_1, V_2) served as a bridge for information sharing. Without such information propagation, the bandit algorithms could still achieve similar performance in the end, but it requires much more observations over time with the cost of decreased user satisfaction. The performance improvement in all bandit algorithms on Delicious dataset is not as significant as that in LastFM, because of the data sparsity issue again: the overlapped observations between users in U_1 and U_2 are four times less than those on LastFM. This makes the learnt hidden features for items in V_2 less accurate.



(c) Delicious dataset (d) LastFM dataset

Figure 4: The effectiveness of collaboration

The above experiments unveil the source where hLinUCB improves over LinUCB: the learnt hidden features help propagate observations from active users and popular items to less active users and less popular items. To further verify this conclusion, we carefully investigated the experiment results in Figure 3, and found that

more than 68.2% of active users (who are ranked at top 50% based on the number of observations) on the LastFM dataset received improved recommendations from hLinUCB, while only 20.7% from LinUCB; and 48.9% of active users benefit from hLinUCB comparing to that of 31.1% from LinUCB on the Delicious dataset.

5. CONCLUSIONS

In this paper, we studied the problem of hidden feature learning for contextual bandit algorithms. Traditional contextual bandit algorithms assume the features that govern the reward generation are entirely accessible to the learner. When some important features are missing due to sampling or privacy constraints, systematic bias will be introduced into online learning. As our solution, hidden features are explicitly modeled in the proposed contextual bandit algorithm. We rigorously prove that the developed bandit algorithm with hidden features achieves a sublinear upper regret bound with high probability; otherwise, a linear regret is inevitable. Extensive experimental comparisons on both simulations and large-scale real-world datasets verified the effectiveness of the proposed algorithm.

Our current solution assumes the knowledge of hidden feature's dimension. This is admittedly hard to achieve in practice. It is important to explore how to determine the dimension of hidden features during online learning. In addition, our regret analysis is based on the assumption of proper initialization of coordinate descent. It is necessary to explore other techniques or optimization procedures for model parameter estimation and derive the corresponding provable arm selection strategies.

6. ACKNOWLEDGMENTS

We thank all the anonymous reviewers for their helpful comments. This project was supported by the National Science Foundation under grant IIS-1553568.

7. REFERENCES

[1] Y. Abbasi-yadkori, D. Pál, and C. Szepesvári. Improved algorithms for linear stochastic bandits. In *NIPS*, pages 2312–2320. 2011.

[2] D. Agarwal and B.-C. Chen. Regression-based latent factor models. In *Proceedings of the 15th ACM SIGKDD*, pages 19–28. ACM, 2009.

[3] P. Auer. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3:397–422, 2002.

[4] P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Mach. Learn.*, 47(2-3):235–256, May 2002.

[5] R. M. Bell and Y. Koren. Lessons from the netflix prize challenge. *ACM SIGKDD Explorations Newsletter*, 9(2):75–79, 2007.

[6] D. Bouneffouf, A. Bouzeghoub, and A. L. Gançarski. A contextual-bandit algorithm for mobile context-aware recommender system. In *Neural Information Processing*, pages 324–331. 2012.

[7] N. Cesa-Bianchi, C. Gentile, and G. Zappella. A gang of bandits. *Proceedings of NIPS*, 2013.

[8] O. Chapelle and L. Li. An empirical evaluation of thompson sampling. In *NIPS*, pages 2249–2257, 2011.

[9] W. Chu, L. Li, L. Reyzin, and R. E. Schapire. Contextual bandits with linear payoff functions. In *International Conference on Artificial Intelligence and Statistics*, pages 208–214, 2011.

[10] S. Filippi, O. Cappe, A. Garivier, and C. Szepesvári. Parametric bandits: The generalized linear case. In *NIPS*, pages 586–594, 2010.

[11] L. Hong, A. S. Doumith, and B. D. Davison. Co-factorization machines: modeling user interests and predicting individual decisions in twitter. In *Proceedings of the sixth ACM WSDM*, pages 557–566. ACM, 2013.

[12] J. Kawale, H. H. Bui, B. Kveton, L. Tran-Thanh, and S. Chawla. Efficient thompson sampling for online matrix-factorization recommendation. In *NIPS*, pages 1297–1305, 2015.

[13] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, (8):30–37, 2009.

[14] J. Langford and T. Zhang. The epoch-greedy algorithm for multi-armed bandits with side information. In *NIPS*, pages 817–824, 2008.

[15] L. Li, W. Chu, J. Langford, and R. E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of 19th WWW*, pages 661–670. ACM, 2010.

[16] L. Li, W. Chu, J. Langford, and X. Wang. Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. In *Proceedings of 4th WSDM*, pages 297–306. ACM, 2011.

[17] W. Li, X. Wang, R. Zhang, Y. Cui, J. Mao, and R. Jin. Exploitation and exploration in a performance based contextual advertising system. In *Proceedings of 16th SIGKDD*, pages 27–36. ACM, 2010.

[18] A. K. Menon and C. Elkan. A log-linear model with latent features for dyadic prediction. In *2010 IEEE ICDM*, pages 364–373. IEEE, 2010.

[19] A. Nakamura. A ucb-like strategy of collaborative filtering. In *ACML*, 2014.

[20] S. Rendle. Factorization machines with libfm. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 3(3):57, 2012.

[21] A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock. Methods and metrics for cold-start recommendations. In *Proceedings of 25th SIGIR*, pages 253–260. ACM, 2002.

[22] X. Su and T. M. Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in artificial intelligence*, 2009:4, 2009.

[23] A. Uschmajew. Local convergence of the alternating least squares algorithm for canonical tensor approximation. *SIAM Journal on Matrix Analysis and Applications*, 33(2):639–652, 2012.

[24] C.-C. Wang, S. R. Kulkarni, and H. V. Poor. Bandit problems with side observations. *Automatic Control, IEEE Transactions on*, 50(3):338–355, 2005.

[25] H. Wang, A. Dong, L. Li, Y. Chang, and E. Gabrilovich. Joint relevance and freshness learning from clickthroughs for news search. In *Proceedings of the 21st WWW*, pages 579–588. ACM, 2012.

[26] Y. Yue and C. Guestrin. Linear submodular bandits and their application to diversified retrieval. In *NIPS*, pages 2483–2491, 2011.

[27] X. Zhao, W. Zhang, and J. Wang. Interactive collaborative filtering. In *Proceedings of the 22nd CIKM*, pages 1411–1420. ACM, 2013.

APPENDIX

Due to space limit, we can only describe the proof scratch in our theoretical analysis of hLinUCB's upper regret bound.

Proof of Lemma 1:

PROOF. By taking the gradient of the objective function defined in Eq (3) with respect to $\boldsymbol{\theta}$ and $\mathbf{v}$ and applying our model assumption specified in Eq (2), we have,

$$\mathbf{A}_{u,t}(\hat{\boldsymbol{\theta}}_{u,t}-\boldsymbol{\theta}^*)=\sum_{t'=1}^{t}(\mathbf{x}_{a_{t'}},\hat{\mathbf{v}}_{a_{t'},t'})\big((\mathbf{v}^*_{a_{t'}}-\hat{\mathbf{v}}_{a_{t'},t'})^\mathsf{T}\boldsymbol{\theta}^{*\mathbf{v}}_u\big)+\sum_{t'=1}^{t}(\mathbf{x}_{a_{t'}},\hat{\mathbf{v}}_{a_{t'},t'})\eta_{t'}-\lambda_1\boldsymbol{\theta}^*$$

in which $\eta_{t'}$ is the Gaussian noise at time t' in reward generation. Therefore, we can bound the function norm of the difference between $\hat{\boldsymbol{\theta}}_{u,t}$ and $\boldsymbol{\theta}^*_u$ by,

$$\begin{aligned}\|\hat{\boldsymbol{\theta}}_{u,t}-\boldsymbol{\theta}^*_u\|_{\mathbf{A}_{u,t}}=&\|\sum_{t'=1}^{t}(\mathbf{x}_{a_{t'}},\hat{\mathbf{v}}_{a_{t'}})\big((\mathbf{v}^*_{a_{t'}}-\hat{\mathbf{v}}_{a_{t'},t'})^\mathsf{T}\boldsymbol{\theta}^{*\mathbf{v}}_u\big)\\&+\sum_{t'=1}^{t}(\mathbf{x}_{a_{t'}},\hat{\mathbf{v}}_{a_{t'},t'})\eta_{t'}-\lambda_1\boldsymbol{\theta}^*_u\|_{\mathbf{A}^{-1}_{u,t}}\\ \le&\|\sum_{t'=1}^{t}(\mathbf{x}_{a_{t'}},\hat{\mathbf{v}}_{a_{t'},t'})\eta_{t'}\|_{\mathbf{A}^{-1}_{u,t}}+\frac{LS}{\sqrt{\lambda_1}}\sum_{t'=1}^{t}\|\mathbf{v}^*_{a_{t'}}-\hat{\mathbf{v}}_{a_{t'},t'}\|_2+\sqrt{\lambda_1}S\end{aligned}$$

where the first term on the right-hand side of the inequality is bounded by the property of self-normalized vector-valued martingales [1], because $\mathbf{x}_{a_t}$ and $\mathbf{v}_{a_t}$ have finite L2 norm and η_t has a finite variance. For the second term, if the regularization parameter λ_1 is sufficiently large, the Hessian matrix of Eq (3) is positive definite at the optimizer based on the property of alternating least square [23]. The estimation of $\boldsymbol{\theta}_u$ and $\mathbf{v}_a$ is thus q-linearly convergent to the optimizer. This indicates for every $\epsilon_1>0$, we have

$$\|\hat{\mathbf{v}}_{a,t+1}-\mathbf{v}^*_a\|_2\le(q_1+\epsilon_1)\|\hat{\mathbf{v}}_{a,t}-\mathbf{v}^*_a\|_2$$

where $0<q_1<1$. As a conclusion, we have for any $\delta>0$, with probability at least $1-\delta$,

$$\|\hat{\boldsymbol{\theta}}_{u,t}-\boldsymbol{\theta}^*_u\|_{\mathbf{A}_{u,t}}\le+\frac{2SL^2}{\sqrt{\lambda_1}}\frac{(q_1+\epsilon_1)(1-(q_1+\epsilon_1)^t)}{1-(q_1+\epsilon_1)}+\sqrt{d\ln\big(\frac{\lambda_1 d+tL^2}{\lambda_1 d\delta}\big)}+\sqrt{\lambda_1}S$$

The same proof techniques apply to the proof of $\|\hat{\mathbf{v}}_{a,t}-\mathbf{v}^*_a\|_{\mathbf{C}_{a,t}}$ □

Proof of Theorem 1:

PROOF. According to the regret definition in Eq (1), the regret at time t can be written as,

$$\begin{aligned}R_t=&r_{a^*_t,u}-r_{a_t,u}=(\mathbf{x}_{a^*_t},\mathbf{v}^*_{a^*_t})^\mathsf{T}\boldsymbol{\theta}^*_u-(\mathbf{x}_{a_t},\mathbf{v}^*_{a_t})^\mathsf{T}\boldsymbol{\theta}^*_u\\ \le&(\mathbf{x}_{a_t},\hat{\mathbf{v}}_{a_t,t})^\mathsf{T}\hat{\boldsymbol{\theta}}_{u,t}+\alpha^u_t\|\mathbf{x}_{a_t},\hat{\mathbf{v}}_{a_t,t}\|_{\mathbf{A}^{-1}_{u,t}}+\alpha^a_t\|\hat{\boldsymbol{\theta}}^{\mathbf{v}}_{u,t}\|_{\mathbf{C}^{-1}_{a_t,t}}\\&+\alpha^a_t\|\boldsymbol{\theta}^{*\mathbf{v}}_u-\hat{\boldsymbol{\theta}}^{\mathbf{v}}_{u,t}\|_{\mathbf{C}^{-1}_{a^*_t,t}}-(\mathbf{x}_{a_t},\mathbf{v}^*_{a_t,t})^\mathsf{T}\boldsymbol{\theta}^*_u\\ =&(\mathbf{x}_{a_t},\hat{\mathbf{v}}_{t,a})^\mathsf{T}(\hat{\boldsymbol{\theta}}_{u,t}-\boldsymbol{\theta}^*_u)+\alpha^u_t\|\mathbf{x}_{a_t},\hat{\mathbf{v}}_{a_t,t}\|_{\mathbf{A}^{-1}_{u,t}}+\alpha^a_t\|\hat{\boldsymbol{\theta}}^{\mathbf{v}}_{u,t}\|_{\mathbf{C}^{-1}_{a_t,t}}\\&+\alpha^a_t\|\boldsymbol{\theta}^{*\mathbf{v}}_u-\hat{\boldsymbol{\theta}}^{\mathbf{v}}_{u,t}\|_{\mathbf{C}^{-1}_{a^*_t,t}}+(\hat{\mathbf{v}}_{a_t,t}-\mathbf{v}^*_{a^*_t})^\mathsf{T}\boldsymbol{\theta}^{*\mathbf{v}}_u\\ \le&\|\hat{\boldsymbol{\theta}}_{u,t}-\boldsymbol{\theta}^*_u\|_{\mathbf{A}_{u,t}}\|\mathbf{x}_{a_t},\hat{\mathbf{v}}_{a_t,t}\|_{\mathbf{A}^{-1}_{u,t}}+\alpha^u_t\|\mathbf{x}_{a_t},\hat{\mathbf{v}}_{a_t,t}\|_{\mathbf{A}^{-1}_{u,t}}\\&+\alpha^a_t\|\hat{\boldsymbol{\theta}}^{\mathbf{v}}_{u,t}\|_{\mathbf{C}^{-1}_{a_t,t}}+\|\hat{\mathbf{v}}_{a_t,t}-\mathbf{v}^*_{a_t}\|_{\mathbf{C}_{a_t,t}}\|\boldsymbol{\theta}^{*\mathbf{v}}_u\|_{\mathbf{C}^{-1}_{a_t,t}}\\&+\alpha^a_t\|\boldsymbol{\theta}^{*\mathbf{v}}_u-\hat{\boldsymbol{\theta}}^{\mathbf{v}}_{u,t}\|_{\mathbf{C}^{-1}_{a^*_t,t}}\\ \le&2\alpha^u_t\|\mathbf{x}_{a_t},\hat{\mathbf{v}}_{a_t,t}\|_{\mathbf{A}^{-1}_{u,t}}+2\alpha^a_t\|\hat{\boldsymbol{\theta}}^{\mathbf{v}}_{u,t}\|_{\mathbf{C}^{-1}_{a_t,t}}\\&+\alpha^a_t\|\boldsymbol{\theta}^{*\mathbf{v}}_u-\hat{\boldsymbol{\theta}}^{\mathbf{v}}_{u,t}\|_{\mathbf{C}^{-1}_{a_t,t}}+\alpha^a_t\|\boldsymbol{\theta}^{*\mathbf{v}}_u-\hat{\boldsymbol{\theta}}^{\mathbf{v}}_{u,t}\|_{\mathbf{C}^{-1}_{a^*_t,t}}\end{aligned}$$

where the first inequality is based on the following two inequalities. First, according to the arm selection strategy if arm a is chosen at trial t, we have

$$\begin{aligned}&(\mathbf{x}_{a_t},\hat{\mathbf{v}}_{a_t,t})^\mathsf{T}\hat{\boldsymbol{\theta}}_{u,t}+\alpha^u_t\|\mathbf{x}_{a_t},\hat{\mathbf{v}}_{a_t,t}\|_{\mathbf{A}^{-1}_{u,t}}+\alpha^a_t\|\hat{\boldsymbol{\theta}}^{\mathbf{v}}_{u,t}\|_{\mathbf{C}^{-1}_{a_t,t}}\\&\ge(\mathbf{x}_{a^*_t},\hat{\mathbf{v}}_{a^*_t,t})^\mathsf{T}\hat{\boldsymbol{\theta}}_{u,t}+\alpha^u_t\|(\mathbf{x}_{a^*_t},\hat{\mathbf{v}}_{a^*_t,t})\|_{\mathbf{A}^{-1}_{u,t}}+\alpha^a_t\|\hat{\boldsymbol{\theta}}^{\mathbf{v}}_{u,t}\|_{\mathbf{C}^{-1}_{a^*_t,t}}\end{aligned}$$

Second, using Cauchy-Schwarz inequality, we get,

$$\begin{aligned}&(\mathbf{x}_{a^*_t},\hat{\mathbf{v}}_{a^*_t,t})^\mathsf{T}\hat{\boldsymbol{\theta}}_{u,t}+\alpha^u_t\|(\mathbf{x}_{a^*_t},\hat{\mathbf{v}}_{a^*_t,t})\|_{\mathbf{A}^{-1}_{u,t}}+\alpha^a_t\|\hat{\boldsymbol{\theta}}^{\mathbf{v}}_{u,t}\|_{\mathbf{C}^{-1}_{a^*_t,t}}-(\mathbf{x}_{a^*_t},\mathbf{v}^*_{a^*_t})^\mathsf{T}\boldsymbol{\theta}^*_u\\&-(\mathbf{x}_{a^*_t},\hat{\mathbf{v}}_{a^*_t,t})^\mathsf{T}\boldsymbol{\theta}^*_u+(\mathbf{x}_{a^*_t},\hat{\mathbf{v}}_{a^*_t,t})^\mathsf{T}\boldsymbol{\theta}^*_u-(\mathbf{x}_{a^*_t},\mathbf{v}^*_{a^*_t})^\mathsf{T}\boldsymbol{\theta}^*_u\\ =&(\mathbf{x}_{a^*_t},\hat{\mathbf{v}}_{a^*_t,t})^\mathsf{T}(\hat{\boldsymbol{\theta}}_{u,t}-\boldsymbol{\theta}^*_u)+(\mathbf{0},(\hat{\mathbf{v}}_{a^*_t,t}-\mathbf{v}^*_{a^*_t}))^\mathsf{T}\boldsymbol{\theta}^*_u\\&+\alpha^u_t\|(\mathbf{x}_{a^*_t},\hat{\mathbf{v}}_{a^*_t,t})\|_{\mathbf{A}^{-1}_{u,t}}+\alpha^a_t\|\hat{\boldsymbol{\theta}}^{\mathbf{v}}_{u,t}\|_{\mathbf{C}^{-1}_{a^*_t,t}}\\ \ge&-\|\hat{\boldsymbol{\theta}}_{u,t}-\boldsymbol{\theta}^*_u\|_{\mathbf{A}_{u,t}}\|(\mathbf{x}_{a^*_t},\hat{\mathbf{v}}_{a^*_t,t})\|_{\mathbf{A}^{-1}_{u,t}}-\|(\hat{\mathbf{v}}_{a^*_t,t}-\mathbf{v}^*_{a^*_t})\|_{\mathbf{C}_{a^*_t,t}}\|\boldsymbol{\theta}^{*\mathbf{v}}_u\|_{\mathbf{C}^{-1}_{a^*_t,t}}\\&+\alpha^u_t\|(\mathbf{x}_{a^*_t},\hat{\mathbf{v}}_{a^*_t,t})\|_{\mathbf{A}^{-1}_{u,t}}+\alpha^a_t\|\hat{\boldsymbol{\theta}}^{\mathbf{v}}_{u,t}\|_{\mathbf{C}^{-1}_{a^*_t,t}}\\ \ge&-\alpha^u_t\|(\mathbf{x}_{a^*_t},\hat{\mathbf{v}}_{a^*_t,t})\|_{\mathbf{A}^{-1}_{u,t}}-\alpha^a_t\|\boldsymbol{\theta}^{*\mathbf{v}}_u\|_{\mathbf{C}^{-1}_{a^*_t,t}}\\&+\alpha^u_t\|(\mathbf{x}_{a^*_t},\hat{\mathbf{v}}_{a^*_t,t})\|_{\mathbf{A}^{-1}_{u,t}}+\alpha^a_t\|\hat{\boldsymbol{\theta}}^{\mathbf{v}}_{u,t}\|_{\mathbf{C}^{-1}_{a^*_t,t}}\\ =&-\alpha^a_t\|\boldsymbol{\theta}^{*\mathbf{v}}_u-\hat{\boldsymbol{\theta}}^{\mathbf{v}}_{u,t}\|_{\mathbf{C}^{-1}_{a^*_t,t}}\end{aligned}$$

in which α^u_t is the upper bound of $\|\hat{\boldsymbol{\theta}}_{u,t}-\boldsymbol{\theta}^*_u\|_{\mathbf{A}_{u,t}}$ and α^a_t is the upper bound of $\|\hat{\mathbf{v}}_{a,t}-\mathbf{v}^*_a\|_{\mathbf{C}_{a,t}}$.

Putting all these together, the the accumulated regret of hLinUCB at time T can be derived as,

$$\begin{aligned}&\mathbf{R}(T)=\sum_{t=1}^{T}R_t\\ \le&\sqrt{T\sum_{t=1}^{T}4(\alpha^u_t)^2\|\mathbf{x}_{a_t},\hat{\mathbf{v}}_{a_t,t}\|^2_{\mathbf{A}^{-1}_{u,t}}}+\sqrt{T\sum_{t=1}^{T}4(\alpha^u_t)^2\|\hat{\boldsymbol{\theta}}^{v}_{u,t}\|^2_{\mathbf{C}^{-1}_{a_t,t}}}\\&+\sum_{t=1}^{T}\alpha^a_t\|\boldsymbol{\theta}^{*\mathbf{v}}_u-\hat{\boldsymbol{\theta}}^{\mathbf{v}}_{u,t}\|_{\mathbf{C}^{-1}_{a_t,t}}+\sum_{t=1}^{T}\alpha^a_t\|\boldsymbol{\theta}^{*\mathbf{v}}_u-\hat{\boldsymbol{\theta}}^{\mathbf{v}}_{u,t}\|_{\mathbf{C}^{-1}_{a^*_t,t}}\\ \le&2\alpha^u_T\sqrt{T\sum_{t=1}^{T}\|\mathbf{x}_{a_t},\hat{\mathbf{v}}_{a_t,t}\|^2_{\mathbf{A}^{-1}_{u,t}}}+2\alpha^a_T\sqrt{T\sum_{t=1}^{T}\|\hat{\boldsymbol{\theta}}^{v}_{u,t}\|^2_{\mathbf{C}^{-1}_{a_t,t}}}\\&+2\alpha^a_T\frac{1}{\sqrt{\lambda_2}}\sum_{t=1}^{T}\|\boldsymbol{\theta}^{*\mathbf{v}}_u-\hat{\boldsymbol{\theta}}^{\mathbf{v}}_{u,t}\|_2\end{aligned}$$

Based on Lemma 11 in [1] and our previous proof, the first and second terms on the right-hand side of above inequality can be bounded by,

$$\begin{aligned}&2\alpha^u_T\sqrt{T\sum_{t=1}^{T}\|\mathbf{x}_{a_t},\hat{\mathbf{v}}_{a_t,t}\|^2_{\mathbf{A}^{-1}_{u,t}}}+2\alpha^a_T\sqrt{T\sum_{t=1}^{T}\|\hat{\boldsymbol{\theta}}^{v}_{u,t}\|^2_{\mathbf{C}^{-1}_{a_t,t}}}\\ \le&2\alpha^u_T\sqrt{2dT\ln(1+\frac{det(\mathbf{A}_{u,t})}{det(\lambda_1\mathbf{I})})}+2\alpha^a_T\sqrt{2lT\ln(1+\frac{det(\mathbf{C}_{a,t})}{det(\lambda_2\mathbf{I})})}\\ \le&2\alpha^u_T\sqrt{2dT\ln(1+\frac{TL}{\lambda_1 d})}+2\alpha^a_T\sqrt{2lT\ln(1+\frac{TS}{\lambda_2 l})}\end{aligned}$$

For the third term of the upper bound in $\mathbf{R}(T)$, according to the q-linear convergence property, we have,

$$\begin{aligned}2\alpha^a_T\frac{1}{\sqrt{\lambda_2}}\sum_{t=1}^{T}\|\boldsymbol{\theta}^{*\mathbf{v}}_u-\hat{\boldsymbol{\theta}}^{\mathbf{v}}_{u,t}\|_2\le&2\alpha^a_T\frac{S}{\sqrt{\lambda_2}}\sum_{t=1}^{T}(q_2+\epsilon_2)^t\\ \le&2\alpha^a_T\frac{1}{\sqrt{\lambda_2}}\frac{(q_2+\epsilon_2)(1-(q_2+\epsilon_2)^T)}{1-(q_2+\epsilon_2)}\end{aligned}$$

□