INTERACTIVE ONLINE LEARNING WITH INCOMPLETE KNOWLEDGE

Qingyun Wu

A dissertation proposal submitted for the degree of Doctor of Philosophy

Dissertation Committees:

Hongning Wang (Advisor),
David Evans (Committee Chair),
Quanquan Gu,
Lihong Li,
Denis Nekipelov

July 2018
Interactive Online Learning with Incomplete Knowledge

Qingyun Wu
Department of Computer Science,
University of Virginia
qw2ky@virginia.edu

Abstract. Interactive online learning is vital in modern information service systems. Despite the recent progress in online learning, many challenges, which are brought by the complex practical scenarios of online learning, remain unsolved. The key challenges can be summarized from the following perspectives. First, from the user-user interaction perspective: to capture user heterogeneity, personalized online learning is needed, while on the other hand, the existence of user dependency calls for collaborative online learning across users, such that the learning process can be accelerated through information propagation. Second, from users’ temporal behavior perspective: information service systems are highly dynamic, which is reflected in the fact that users’ preferences change over time due to various internal or external factors [21], and item popularity vary due to fast emerging events/contents. Failing to model such dynamics may lead to sub-optimal decisions. Third, from the user-system interaction perspective: learning to interact with users and discover their preferences from repeated interactions is central in most information service systems. Instead of passively waiting for users’ feedback, proactive information acquisition should be encouraged. In addition, user feedback in online systems can be implicit. For example, user clicks, although abundant in online information service systems, can be biased and incomplete due to position bias [34].

The proposed research aims at developing online learning solutions, and more specifically multi-armed bandit solutions, to conquer the aforementioned challenges. Firstly, I propose to perform contextual bandit learning in a collaborative manner, which enables information sharing across users, and thus expedites the learning process in a dynamic environment. Secondly, I propose to study bandit learning in a more realistic non-stationary environment such that the learning algorithm can automatically detect the potential changes and adapt its decision making strategy accordingly. Finally, I propose to improve the effectiveness of user-system interaction by: (1) learning from implicit user feedback and (2), proactively choosing the most representative users or information to initiate or incentivize the interaction for the most beneficial feedback, which further improves the system’s utility in the long run. By combining the proposed research, an information service system can provide right information to the right user at the right time, so as to improve user satisfaction in the long run. This will become a new paradigm for human-machine interaction: both systems and users win in this interactive game. More importantly, the proposed solutions can be applied to a wide spectrum of applications including not only the aforementioned information service systems, but also crowdsourcing, human-machine interactions in cyber-physical systems, clinical trials in healthcare, sequential treatment design in psychology and many more.
# Table of Contents

Interactive Online Learning with Incomplete Knowledge ........................................... 1

*Qingyun Wu*

1 **Introduction** .......................................................... 1

1 Modeling User-User Dependencies ......................................................... 1
2 Modeling Users' Temporal Behaviors ..................................................... 2
3 Modeling User-System Interactions ....................................................... 3

2 **Related Works** ............................................................ 5

1 Collaborative Bandit Learning ............................................................... 5
2 Bandit Learning in A Non-Stationary Environment ....................................... 6
3 Implicit Feedback and Proactive Bandit Learning ....................................... 7

3 **Research Challenges and Proposed Research** ............................................. 8

1 Research Challenge I: Bandit Learning in a Collaborative Environment .......... 8
   1.1 Proposed Research I.A: Collaborative Contextual Bandits ....................... 8
       1.1.1 Regret Analysis ........................................................................ 9
       1.1.2 Empirical Evaluation .................................................................. 10
   1.2 Proposed Research I.B: Learning User Dependency Graph for Collaborative Bandits .. 11
   1.3 Proposed Research I.C: Factorization Bandits ..................................... 11
2 Research Challenge II: Bandit Learning in a Non-Stationary Environment .......... 12
   2.1 Proposed Research II.A: Dynamic Contextual Bandits Learning ............. 12
       2.1.1 Regret Analysis ........................................................................ 14
       2.1.2 Empirical Evaluation .................................................................. 15
   2.2 Proposed Research II.B: Context-Dependent Dynamic Ensemble of Contextual Bandits 16
   2.3 Proposed Research II.C: Collaborative Bandit Learning in a Non-Stationary Environment 18
3 Research Challenge III: Online Learning from Implicit and Incomplete Feedback .......... 19
   3.1 Proposed Research III.A: Bandit Learning with Implicit Feedback ............ 19
   3.2 Proposed Research III.B: Proactive User-System Interaction .................. 19

4 **Summary: Impact Statement of Proposed Work and Time-line** ................. 21

1 Impact Statement ................................................................................. 21
2 Proposed Time-Line ............................................................................... 21
List of Tables

List of Figures

1. Empirical evaluation on synthetic dataset and Yahoo dataset. .......................... 10
2. Illustration of dLinUCB. The master bandit model maintains the ‘badness’ estimation of slave models over time to detect changes in the environment. At each round, the most promising slave model is chosen to interact with the environment; and the acquired feedback is shared across all admissible slave models for model update. ............... 13
3. Results from synthetic dataset and Yahoo dataset. ........................................ 15
4. Performance comparison in LastFM & Delicious. ......................................... 16
5. Word cloud of tags from four identified user groups in dLinUCB on LastFM dataset. ........ 16
Chapter 1

Introduction

Satisfying users with personalized information plays a crucial role for online service providers to succeed in market. However, the rapid appearance of new information and new users together with the ever-changing nature of content popularity make traditional recommendation approaches, e.g., collaborative filtering [9, 56], incompetent. Modern information service systems now adopt online learning solutions to adaptively find good mappings between available content and users. During online learning, the need to focus on information that raises user interest and, simultaneously, the need to explore new information for globally improving user experience create an explore-exploit dilemma.

Significant research attention has been paid on multi-armed bandit algorithms [28, 42, 6, 7], which provide a principled solution of the explore-exploit dilemma. Intuitively, bandit algorithms designate a small amount of traffic to collect user feedback while improving their estimation qualities in realtime. With the available side information about users or items to be presented, contextual bandits have become a reference solution [5, 16, 44, 23]. Specifically, contextual bandits assume the expected payoff is determined by a conjecture of unknown bandit parameters and given context, which is represented as a set of features extracted from both users and recommendation candidates. Such algorithms are especially advantageous when the space of recommendation is large but the payoffs are interrelated. They have been successfully applied in many important applications, e.g., content recommendation [44, 8] and display advertising [17, 47]. However, there are many challenges hindering the application of bandit algorithms to real world problems.

1 Modeling User-User Dependencies

Due to the existence of social influence [21], e.g., content and opinions sharing among friends in a social network, exploiting the dependency among users raises new challenges and opportunities in personalized information services. For example, in many real-world applications, e.g., content recommendation in Facebook or Twitter, because of the mutual influence among friends and acquaintances, one user’s click decision on the recommended items might be greatly influenced by his/her peers. This indicates the knowledge gathered about the interest of a given user can be leveraged to improve the recommendation to his/her friends, i.e., collaborative learning. In other words, the observed payoffs from a user’s feedback might be a compound of his/her own preference and social influence he/she receives, e.g., social norms, conformity and compliance. As a result, propagating the knowledge collected about the preference of one user to his/her related peers can not only capitalize on additional information embedded in the dependency among users, which is not available in the context vectors; but also helps conquer data sparsity issue by reducing the sample complexity of preference learning (e.g., known as cold-start in recommender systems [57]). Failing to recognize such information among users will inevitably lead to a suboptimal solution.

In order to deal with the aforementioned collaborative environment, I propose a collaborative bandit learning framework that explicitly models the underlying dependency among users during online
learning. By leveraging user dependency, information sharing will be enabled while online updating. Specifically, a user graph will be constructed according to the available users’ dependency information. Each node in the graph represents a contextual bandit agent deployed for a single user and the weight on each edge indicates the influence between a pair of users. Based on this dependency structure, bandit parameters can be estimated over all the users in a collaborative manner: both context and received payoffs from one user are propagated across the whole graph in the process of online updating. The proposed collaborative bandit algorithm establishes a bridge to share information among heterogeneous users and thus reduce the sample complexity of preference learning. Rigorously theoretical analysis and extensive empirical evaluation will be provided to prove the effectiveness of the proposed solution.

The proposed collaborative bandit algorithm capitalize on the relatedness among the bandit models deployed across individual users to benefit each of those learning tasks. However, one potential limitation of the proposed collaborative bandit solution and other existing collaborative bandit learning solutions, is that one has to assume the user dependency relation that affects users’ decisions on recommendations is known beforehand. But in many real-world applications, the available relational information among users may be irrelevant or simply unavailable. Little is known about those collaborative bandit algorithms’ expected behavior in such cases. We propose to study the regret bound of collaborative bandit algorithm with erroneous dependency information and also propose to learn user dependency information on the fly from sequential user feedback.

In addition to explicitly modeling user dependency information, matrix factorization based collaborative filtering [40, 39, 60] provides a way to capture the underlying dependency among users or items. The basic idea of such solutions is to characterize both recommendation items and users by vectors of latent factors inferred from historical user-item preference patterns via low-rank matrix completion [12, 13], with an assumption that only a few factors contribute to an individual’s taste [40]. In the online learning setting, I propose to perform online interactive learning by placing a factorization-based bandit algorithm on each user in the system. By sequentially learning a low-rank structure of the incrementally constructed user-item preference matrix, information sharing can be achieved on both the user side and item side.

2 Modeling Users’ Temporal Behaviors

Most existing stochastic contextual bandit algorithms assume an unknown but fixed reward mapping function [44, 23, 46, 66, 25]. In practice, this translates to the assumption that users’ preferences remain static over time. However, this assumption rarely holds in reality as users’ preferences can be influenced by various internal or external factors [21]. For example, when a sports season ends after a championship, seasonal fans might jump over to following a different sport and not have much interest in the off-season. More importantly, such changes are often not observable to the learners. If a learning algorithm fails to model or recognize the possible changes of the environment, it would constantly make suboptimal choices. In my thesis research, moving beyond a restrictive stationary environment assumption, I propose to study a more sophisticate but realistic environment setting where the reward mapping function becomes non-stationary over time. More specifically, we focus on the setting where there are abrupt changes in terms of user preferences (e.g., user interest in a recommender system) and those changes are not observable to the learner beforehand. Between consecutive change points, the reward distribution remains stationary yet unknown, i.e., a piecewise stationary environment. In my thesis, I propose a set of algorithmic solutions to conquer such a non-stationary environment.

Based on the properties of abruptly changing environment, in any stationary period between two consecutive change points, the reward estimation error of a contextual bandit model trained on the observations collected from that period should be bounded with a high probability [1, 19]. Otherwise, the model’s consistent wrong predictions can only come from the change of environment. Based on this insight, I propose to evaluate whether the stationary assumption holds by monitoring a bandit model’s
reward prediction quality over time. When a bandit model’s reward prediction quality is bad enough, a change can be considered to be detected and new bandit model should be created accordingly. To reduce variance in the prediction error from one bandit model, we ensemble a set of models, by creating and abandoning them on the fly.

In addition, in the contextual bandit setting, the changes of reward distributions caused by a non-stationary environment become context dependent. Most existing algorithms that attempt to adapt to this dynamic environment introduce a forgetting mechanism to downweight the historical observations [24, 31], or create a bandit model for each newly detected stationary period [30, 67]. These strategies, however, do not leverage past information optimally. This is because after an abrupt change in the environment, some arms’ rewards may be relatively unchanged. This can happen if the underlying change is orthogonal to the context. As a result, strategies that discount observations or abandon the ‘old’ model must regain confidence in parameters, incurring a higher regret due to redundant exploration. The possible existence of change-invariant arms in such a non-stationary environment suggests us to reuse the bandit models estimated for those earlier periods, such that more accurate reward estimation on such arms can be achieved sooner so as to obtain reduced regret in this new period. In this research, I propose to construct a dynamic context-dependent bandit ensemble method to capitalize on this unique context-dependent property in the non-stationary environment: the algorithm maintains a unique dynamic set of contextual bandit models for each specific arm instead of uniformly maintaining the same set of bandit models for all the arms. This makes it possible for those change-invariant arms to reuse old slave models while not hurting whose arms whose expected reward has changed.

Existing work on both collaborative bandit learning and multi-task learning [66, 16, 15] have suggested that sharing information among multiple learning agents can accelerate learning. Information sharing could be particularly helpful if learners operate in a changing environment, because a learner could benefit from previous experience of another learner to adapt to its new environment. In addition, sociologists have long converged that the evolution of user preferences in a social network is driven by the interplay between users’ preferences and the social network structure [64]. There are two social theories that explain this evolution: the social influence effect states that users’ future preferences are affected by the social network around them, and the homophily effect suggests that people tend to associate and bond with others that have similar preferences [4]. These social psychology perspectives further indicate the possibility and benefits of collaborative learning in a non-stationary environment. Inspired by the above insights, I propose to perform collaborative bandit learning to conquer a non-stationary environment. First of all, since different users may share the same interest/preference (not necessarily at the same time), bandit learner’s experience on some users can be reused for some other users. In addition, with mild assumptions about how interests/preferences evolve and shared among users, prior information can be utilized to help predict how users’ preference may change in a non-stationary environment.

3 Modeling User-System Interactions

Learning to interact with users and discover their preferences from repeated interactions is central in most information service systems. However various sources of bias, for example user bias and position bias, in information service systems, hinder the effectiveness of online learning solutions, which need to learn from user-system interaction. In order to address the these challenges, two research questions need to be answered: 1. How to learn from users’ implicit feedback? 2. How to encourage more informative user feedback?

Contextual bandit algorithms [5, 44, 43] provide modern information service systems an effective solution to adaptively find good mappings between available items and users from available user feedback. However, the most dominant form of user feedback in such systems is implicit feedback, such as clicks, which is known to be biased and incomplete about users’ evaluation of system’s output [38, 32]. For example, a user skips a recommended item might not be because he/she does not like the item, but he/she
just does not examine that display position, i.e., position bias [34]. Unfortunately, a common practice in contextual bandit applications simply treats no click as a form of negative feedback [44, 62, 17]. This introduces inconsistency to model update, since the skipped items might not be truly irrelevant, and it inevitably leads to suboptimal outputs of bandit algorithms over time.

Inspired by the examination hypothesis [22], in click modeling, I propose to learn contextual bandits with implicit user click feedback, and model such implicit feedback as a composition of user result examination and relevance judgment. Examination hypothesis postulates that a user clicks on a system’s returned result if and only if that result has been examined by the user and it is relevant to the user’s information need at the moment. Because a user’s examination behavior is unobserved, we propose to model users’ examination behavior as a latent variable in the bandit setting, and realize the examination hypothesis in a probabilistic model.

Interactive online learning in information service systems can be considered as a process of sequential preference elicitation. Instead of passively waiting for users’ feedback, proactive information acquisition should be encouraged to make this learning process more effective. Proactive user-system interaction not only involves encouraging the most informative actions but also the most informative users. There is a recent attempt on interactive question selection using bandit learning solution in conversational recommender systems [18]. However such solution ignores the different informativeness of users in the systems. In a collaborative environment, users are no longer independent, if a system could start the users whose feedback can mostly reduce the system’s uncertainty about the other users, less exploration would be needed to optimize the system’s utility from all the users. In this work, I propose to perform proactive information acquisition in both the action space and the user space to encourage proactive information acquisition.
Chapter 2

Related Works

Multi-armed bandit algorithms provide principled solutions to the explore/exploit dilemma, which exists in many real-world applications, such as display advertisement selection [17, 47], recommender systems [44, 8], and search engine systems [54, 69]. As opposed to the traditional $K$-armed bandit problems [7, 6, 28, 42], feature vectors in contextual bandits are created to infer the conditional expected payoff of an action [5, 19, 43, 44]. The setting for contextual bandit with linear payoffs was first introduced in [5], where the expectation of payoff for each action is assumed to be a linear function of its context vector. In the follow-up research [44, 19], LinUCB is introduced to use ridge regression to compute the expected payoff of each action and corresponding confidence interval. Later on, generalized linear models are introduced to parameterize bandit algorithms for non-linear payoffs [23]. Comparing to their context-free counterparts, contextual bandits have achieved superior performance in various application scenarios [44, 23].

1 Collaborative Bandit Learning

The idea of modeling dependency among bandits has been explored in prior research [10, 11, 16, 26, 35]. Studies in [3, 58] explore contextual bandits with assumptions about metric or probabilistic dependencies on the product space of context and actions. Hybrid-LinUCB [44] is such an instance, which uses a hybrid linear model to share observations across users. Social network structures are explored in bandit algorithms for introducing possible dependencies [11, 26]. In [10], parallel context-free $K$-armed bandits are coupled by the social network structure among the users, where the observed payoffs from neighboring nodes are shared as side-observations to help estimate individual bandits. Besides utilizing existing social networks for modeling relatedness among bandits, there is also work automatically estimates the bandit parameters together with the dependency relation among them, such as clustering the bandits via the learned model parameters during online updating [26]. Some recent work incorporates collaboration among bandits via matrix factorization based collaborative filtering techniques: Kawale et al. preformed online matrix factorization based recommendation via Thompson sampling [37, 25].

The most related work to our proposed solutions is the GOB.Lin algorithm introduced in [16]. GOB.Lin requires connected users in a network to have similar bandit parameters via a graph Laplacian based model regularization. As a result, GOB.Lin explicitly requires the learned bandit parameters across related users to be close to each other. In our algorithm, we do not have such strong assumption about each individual bandit, but we make explicit assumptions about the reward generation via an additive model: neighboring users’ judgement of the recommendations will be shared across, i.e., word-of-mouth, to explain the observed payoffs in different users. This gives us the flexibility in capturing the heterogeneity of preferences among different users in practice, and leads to both theoretically and empirically improved results.

There are some recent developments that focus on online collaborative filtering with multi-armed bandit algorithms. [70] studies interactive collaborative filtering via probabilistic matrix factorization.
Both context-free and contextual bandit algorithms are introduced to perform online item selection based on the factorization results. [37] performs online low-rank matrix completion, where the explore/exploit balance is achieved via Thompson sampling. [53] introduces a UCB-like strategy to perform interactive collaborative filtering. 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. [46] performs co-clustering on users and items for collaborative filtering, where confidence bound on reward estimation is used to decide the clustering structures. However, because of the ad-hoc combinations of collaborative filtering methods and bandit methods in the aforementioned studies, limited theoretical understanding is available in those solutions. In this work, we provide a rigorous regret bound analysis of the developed factorization-based bandit algorithm, and demonstrate the algorithm’s convergence property under different conditions. Moreover, our online factorization solution is general enough to incorporate several recent advances in factorization techniques, such as feature-based latent factor models [2, 55] and modeling mutual dependency among users [51, 50], which further improve the proposed algorithm’s convergence rate during interactive online learning with users.

2 Bandit Learning in A Non-Stationary Environment

There are some existing works studying the non-stationary bandit problems. A typical non-stationary environment setting is the abruptly changing environment, or piecewise stationary environment, in which the environment undergoes abrupt changes at unknown time points but remains stationary between two consecutive change points. To deal with such an environment, Hartland et al. [31] proposed the $\gamma$–Restart algorithm, in which a discount factor $\gamma$ is introduced to exponentially decay the effect of past observations. Garivier and Moulines [24] proposed a discounted-UCB algorithm, which is similar to the $\gamma$–Restart algorithm in discounting the historical observations. They also proposed a sliding window UCB algorithm, where only observations inside a sliding window are used to update the bandit model. Yu and Mannor [67] proposed a windowed mean-shift detection algorithm to detect the potential abrupt changes in the environment. An upper regret bound of $O(\Gamma T \log(T))$ is proved for the proposed algorithm, in which $\Gamma T$ is the number of ground-truth changes up to time $T$. However, they assume that at each iteration, the agent can query a subset of arms for additional observations. Slivkins and Upfal [59] considered a continuously changing environment, in which the expected reward of each arm follows Brownian motion. They proposed a UCB-like algorithm, which considers the volatility of each arm in such an environment. The algorithm restarts in a predefined schedule to account for the change of reward distribution.

Most existing solutions for non-stationary bandit problems focus on context-free scenarios, which cannot utilize the available contextual information for reward modeling. Ghosh et al. proposed an algorithm in [27] to deal with environment misspecification in contextual bandit problems. Their algorithm comprises a hypothesis test for linearity followed by a decision to use either the learnt linear contextual bandit model or a context-free bandit model. But this algorithm still assumes a stationary environment, i.e., neither the ground-truth linear model nor unknown models are changing over time. Liu et al. [48] proposed to use cumulative sum and Page-Hinkley Test to detect sudden changes in the environment. An upper regret bound of $O(\sqrt{\Gamma T T \log T})$ is proved for one of their proposed algorithms. However, this work is limited to a simplified Bernoulli bandit environment. Recently, Luo et al [49] studied the non-stationary bandit problem and proposed several bandit algorithms with statistical tests to adapt to changes in the environment. They analyzed various notions of regret including interval regret, switching regret, and dynamic regret. Hariri et al. [30] proposed a contextual Thompson sampling algorithm with a change detection module, which involves iteratively applying a combination of cumulative sum charts and bootstrapping to capture potential changes of user preference in interactive recommendation. But no theoretical analysis is provided about this proposed algorithm.
3 Implicit Feedback and Proactive Bandit Learning

As having been extensively studied in click modeling of user search results [20], various factors affect users’ click decisions, and among them result examination plays a central role [34, 22]. Unfortunately, most applications of bandit algorithms simply treat user clicks as feedback for model update [62, 17, 68], where no click on a selected result is considered as negative feedback. This inevitably leads to inaccurate model update and sub-optimal arm selection.

There is a line of research that develops click model based bandit algorithms for learning to rank problems. For example, by assuming that skipped documents are less attractive than later clicked ones in a ranked list, Kveton et al. [41] develop a cascading bandit model to learn from both clicks and skips in search results. To enable learning from multiple clicks in the same result ranking list, they adopt the dependent click model [29] to infer user satisfaction after a sequence of clicks [36], and later further extend to broader types of click models [71]. However, such algorithms aim at estimating the best ranking of results in a per-query basis, without specifying any ranking function nor generalizing to unseen queries. This limits their application scenario in practice. Another line of related research is bandit learning with latent variables. Maillard and Mannor studied the problem of latent bandit [52], which assumes reward distributions are clustered and the clusters are determined by some latent variables. They only studied the problem in a context-free setting, and a very weak performance guarantee is provided when the reward distribution is unknown in those clusters. Kawale et al. developed a Thompson sampling scheme for online matrix-factorization [37]. Latent features are extracted via an online low-rank matrix completion based on samples selected from Thompson sampling on the fly. Due to the ad-hoc combination of factorization method and bandit method, little theoretical analysis is provided. Wang et al. studied the problem of latent feature learning for contextual bandits [62]. They extended arms’ context vectors with latent features under a linear reward structure, and applied the upper confidence bound principle over coordinate descent to iteratively estimate the hidden features and model parameters. The linear reward structure prohibits it from recognizing the nonlinear dependency between result examination and relevance judgment in click feedback.

In terms of proactive bandit learning, recently, Christakopoulou et al. [18] proposed a bandit-based preference elicitation framework to identify which questions to ask a new user to quickly learn user preference. To the best of our knowledge, there is no existing work considers active feedback acquisition from the combinatory space of candidate actions and users.
Chapter 3

Research Challenges and Proposed Research

1 Research Challenge I: Bandit Learning in a Collaborative Environment

1.1 Proposed Research I.A: Collaborative Contextual Bandits

In bandit problems, an online learning agent takes sequential actions according to immediate feedback from the environment (i.e., the users of a system) to maximize its cumulated reward (i.e., positive feedback from users). Contextual bandits further assume the payoff of each action with respect to different users are governed by an noisy payoff function with respect to the context vectors, e.g., linear payoff function [44, 5].

Our proposed solution is built on contextual bandits. In particular, we assume the collaborative relationship among users can be encoded as a weighted graph \( G = (V, E) \). Each node \( v_i \in \{V_1, ..., V_N\} \) in \( G \) hosts a bandit parameterized by \( \theta_i \) for user \( i \); and the edges in \( E \) represent the affinity relation over pairs of users. This graph can be described as an \( N \times N \) stochastic matrix \( W \). In this matrix, each element \( w_{ij} \) is nonnegative and proportional to the influence that user \( j \) has on user \( i \) in determining the payoffs of different arms. \( W \) is column-wise normalized such that \( \sum_{j=1}^{N} w_{ij} = 1 \) for \( i \in \{1, ..., N\} \). In this work, we assume \( W \) is time-invariant and known to the learner beforehand. Based on the graph \( G \), collaboration among bandits happens when determining the payoff of a particular arm with respect to a given user. To denote this, we define a \( d \times N \) matrix \( \Theta \), which consists of parameters from all the bandits in the graph:

\[
\Theta = (\theta_1, ..., \theta_N)
\]

Accordingly, we define a context feature matrix \( X_t = (x_{a_t,1}, ..., x_{a_t,N}) \), where the \( i \)th column is the context vector \( x_{a_t,i} \), for arm \( a \) at trial \( t \) selected for user \( i \). The collaboration among bandits characterized by the influence matrix \( W \) results in a new bandit parameter matrix \( \bar{\Theta} = \Theta W \), which determines the payoff \( r_{a_t,u_t} \) of arm \( a_t \) for user \( u_t \) at trial \( t \) by:

\[
r_{a_t,u_t} - \text{diag}_{u_t}(X_t^T \Theta W) \sim N(0, \sigma^2)
\]

where \( \text{diag}_{u_t}(X) \) is the operation returning the \( u_t \)-th element in the diagonal of matrix \( X \). Eq (1) postulates our additive assumption about reward generation in this collaborative environment: the reward \( r_{a_t,u_t} \) is not only determined by user \( u_t \)'s own preference on the arm \( a_t \) (i.e., \( w_{u_t,u_t}x^T_{a_t,u_t} \theta_{u_t} \)), but also by the judgement from the neighbors who have influence on \( u_t \) (i.e., \( \sum_{j \neq u_t} w_{u_t,j}x^T_{a_t,j} \theta_j \)). This enables us to distinguish a user’s intrinsic preference of the recommended content from his/her neighbors’ influence, i.e., personalization. In addition, the linear payoff assumption in our model is to simplify the discussion in this paper; and it can be relaxed via a generalized linear model [23] to deal with nonlinear rewards.

With the collaborative assumption about the expected payoffs defined in Eq (1), we appeal to ridge regression for estimating the unknown bandit parameter \( \theta \) for each user. In particular, we simultaneously estimate the global bandit parameter matrix \( \Theta \) for all the users as follows,

\[
\bar{\Theta} = \arg \min_{\Theta} \frac{1}{2} \sum_{t=1}^{T} (\text{vec}(X_{a_t,u_t})^T \text{vec}(\Theta W) - r_{a_t,u_t})^2 + \frac{\lambda}{2} \text{tr}(\Theta^T \Theta)
\]
where \( \lambda \in [0, 1] \) is a trade-off parameter of L2 regularization in ridge regression, and vec(\cdot) is the matrix vectorize operation. \( \mathbf{X}_{a_t,u} \) is a special case of \( \mathbf{X}_{a} \); only the column corresponding to the user \( u_t \) at time \( t \) is set to \( \mathbf{x}_{a_t,u_t} \), and all the other columns are set to zero. This corresponds to the situation that at trial \( t \) the learner only needs to interact with one user.

Since the objective function defined in Eq (2) is quadratic with respect to \( \Theta \), we have a closed-form estimation of \( \Theta \) as vec(\( \Theta_t \)) = \( \mathbf{A}_t^{-1} \mathbf{b}_t \), in which \( \mathbf{A}_t \) and \( \mathbf{b}_t \) are computed as, \( \mathbf{A}_t = \lambda \mathbf{I} + \sum_{t'=1}^{t} \text{vec}(\mathbf{X}_{a_{t'},u_{t'}} \mathbf{W}^T) \text{vec}(\mathbf{X}_{a_{t'},u_{t'}} \mathbf{W}^T)^T \) and \( \mathbf{b}_t = \sum_{t'=1}^{t} \text{vec}(\mathbf{X}_{a_{t'},u_{t'}} \mathbf{W}^T) r_{a_{t'},u_{t'}} \) where \( \mathbf{I} \) is a \( dN \times dN \) identity matrix. The effect of collaboration among bandits is clearly depicted in the above estimation of \( \Theta \). Matrix \( \mathbf{A}_t \) and vector \( \mathbf{b}_t \) store global information shared among all the bandits in the graph.

More specifically, the context vector \( \mathbf{x}_{a_t,u_t} \) and payoff \( r_{a_t,u_t} \) observed in user \( u_t \) at trial \( t \) are propagated through the whole graph via the relational matrix \( \mathbf{W} \). To understand this, note that vec(\( \mathbf{X}_{a_t,u_t} \mathbf{W}^T \)) is a dense vector with projected context vectors on every user, while the original \( \mathbf{X}_{a_t,u_t} \) is a sparse vector with observations only at active users \( u_t \). Because of this information sharing, at certain trial \( t \), although some users might generate any observation yet (i.e., cold-start), they can already start from a non-random initialization of their bandit parameters \( \theta_t \). The estimated bandit parameters \( \hat{\Theta}_t \) predict the expected payoff of a particular arm for each user according to the observed context feature matrix \( \mathbf{X}_t \).

Our collaborative assumption in Eq (1) implies that \( r_{a_t,u_t} \) across users are independent given \( \mathbf{X}_t \) and \( \mathbf{W} \). As a result, for any \( \sigma \), i.e., the standard deviation of Gaussian noise in Eq (1), the following inequality holds with probability at least \( 1 - \delta \),

\[
|r_{a_t,u_t} - r_{a_t,u_t}| \leq \alpha_t \sqrt{\text{vec}(\mathbf{X}_{a_t,u} \mathbf{W}^T)^T \mathbf{A}_t^{-1} \text{vec}(\mathbf{X}_{a_t,u} \mathbf{W}^T)}
\]

where \( \alpha_t = \sqrt{dN \ln \left( 1 + \frac{\sum_{t'=1}^{T} \sum_{u=1}^{N} \mathbf{w}_{a_t,u}^2}{\lambda dN} \right)} \). The proof of this inequality can be found in Lemma 1 of [66]. The inequality in Eq (3) gives us a reasonably tight upper confidence bound (UCB) for the expected payoff of a particular arm over all users in the graph \( G \), from which a UCB-style action-selection strategy can be derived. In particular, at trial \( t \), we choose an arm for user \( u_t \) by

\[
a_{t,u_t} = \arg \max_{a \in \mathcal{A}} \left( \text{vec}(\mathbf{X}_{a,u_t})^T \text{vec}(\hat{\Theta}_t \mathbf{W}) + \alpha_t \sqrt{\text{vec}(\mathbf{X}_{a,u_t} \mathbf{W}^T)^T \mathbf{A}_t^{-1} \text{vec}(\mathbf{X}_{a,u_t} \mathbf{W}^T)} \right)
\]

We name this resulting algorithm as Collaborative Linear Bandit, or CoLin in short. One potential issue with CoLin is its computational complexity: matrix inverse has to be performed on \( \mathbf{A}_t \) at every trial. First, because of the rank one update of matrix \( \mathbf{A}_t \), quadratic computation complexity is possible via applying the Sherman-Morrison formula. Second, we may compute \( \mathbf{A}_t^{-1} \) in a mini-batch manner to further reduce computation with some extra penalty in regret. We will leave this as our future research.

### 1.1.1 Regret Analysis
We provide detailed regret analysis of our proposed CoLin algorithm. We focus on the accumulated pseudo regret, which is formally defined as, \( \mathbf{R}(T) = \sum_{t=1}^{T} \mathbb{E}[r_{a_t} - \mathbb{E}[r_{a_t}]] \), where \( a_t^* \) is the best arm to select according to the oracle of this problem, and \( a_t \) is the arm selected by the algorithm to be evaluated.

**Theorem 1.** With probability at least \( 1 - \delta \), the cumulated regret of CoLin algorithm satisfies,

\[
\mathbf{R}(T) \leq 2\alpha_T \sqrt{2dNT \ln \left( 1 + \frac{\sum_{t=1}^{T} \sum_{j=1}^{N} \mathbf{w}_{a_t,j}^2}{\lambda dN} \right)}
\]

The detailed proof of this theorem is provided in the Appendix of [66]. As shown in Theorem 1, the graph structure plays an important role in the upper regret bound of our CoLin algorithm. Consider two extreme...
cases. First, when $W$ is an identity matrix, i.e., no influence among users, the upper regret bound degenerates to $O(N\sqrt{T\ln \frac{1}{\delta}})$. Second, when the graph is fully connected and uniform, i.e., $\forall i, j, w_{ij} = \frac{1}{N}$, such that users have homogeneous influence among each other, and the upper regret bound of CoLin decreases to $O(N\sqrt{T\ln \frac{1}{\delta}})$. That means via collaboration, CoLin achieves an $O(\sqrt{T\ln N}$) regret reduction for every single user in the graph comparing to the independent case.

### 1.1.2 Empirical Evaluation

We performed empirical evaluations of our CoLin algorithm against several state-of-the-art contextual bandit algorithms, including $N$ independent LinUCB [44], hybrid LinUCB with user features [44], GOB.Lin [16], and online cluster of Bandits (CLUB) [26]. We tested all the algorithms on a synthetic data set via simulations, and a large collection of click stream from Yahoo! Today Module dataset [44].

![Empirical evaluation on synthetic dataset and Yahoo dataset](image)

**Fig. 1.** Empirical evaluation on synthetic dataset and Yahoo dataset.

In simulation, we generate $N$ users, each of which is associated with a $d$-dimensional parameter vector $\theta^*$, i.e., $\Theta^* = (\theta^*_1, \ldots, \theta^*_N)$. $\Theta^*$ is treated as the ground-truth bandit parameters for reward generation, and they are unknown to bandit algorithms. We then construct the golden relational stochastic matrix $W$ for the graph of users by defining $w_{ij} \propto \langle \theta^*_i, \theta^*_j \rangle$, and normalize each column of $W$ by its L1 norm. The resulting $W$ is disclosed to the bandit algorithms. In the end, we generate a size-$K$ action pool $A$. Each action $a$ in $A$ is associated with a $d$-dimensional feature vector $x_a$. To simulate the collaborative reward generation process among users, we first compute $\tilde{\Theta}^* = \Theta^* W$ and then compute the payoff of action $a$ for user $i$ at trial $t$ as $r_{a_{i,t}} = \text{diag}(X_i^t \tilde{\Theta}^*) + \epsilon_t$, where $\epsilon_t \sim N(0, \sigma^2)$. As we can find in Figure 1 (a), simply running $N$ independent LinUCB algorithm gives us the worst regret, which is expected. Hybrid LinUCB, which exploits user dependency via a set of hybrid linear models over user features performed better than LinUCB, but still much worse than CoLin. Although GOB.Lin also exploits the graph structure when estimating the bandit parameters, its assumption about the dependency among bandits is too restrictive to well capture the information embedded in the interaction with users.

We also tested the proposed algorithm on a large collection of ten days’ real traffic data from Yahoo! Today Module [44] using the unbiased offline evaluation protocol proposed in [45]. However, this dataset does not contain any user identity. This 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 for the graph of users by defining $w_{ij} \propto \langle u_i, u_j \rangle$. The CoLin and GOB.Lin algorithms are then executed over those identified user groups. For the LinUCB baseline, we tested two variants: one is individual LinUCBs running over the identified user groups and it is denoted as M-LinUCB; another one is a uniform LinUCB shared by all the users, i.e., it does not distinguish individual users, and thus it is denoted as Uniform-LinUCB. Click-through-rate...
(CTR) was used to evaluate the performance of all bandit algorithms on Yahoo dataset. Average CTR is computed in every 2000 observations (not the cumulated CTR) for each algorithm based on the unbiased offline evaluation protocol proposed in [45, 44]. Following the same evaluation principle used in [44], we normalized the resulting CTR from different bandit algorithms by the corresponding logged random strategy’s CTR. We report the normalized CTR results from different contextual bandit algorithms over 160 derived user groups in Figure 1 (b). CoLin outperformed all baselines on this real-world data set, except CLUB on the first day. As we can find CLUB achieved the best CTR on the first day; but as some popular news articles became out-of-date, CLUB cannot correctly recognize their decreased popularity, and thus provided degenerated recommendations. But in CoLin, because of collaborative preference learning, it better controlled the exploration-exploitation trade-off and thus timely recognized the change of items’ popularity.

1.2 Proposed Research I.B: Learning User Dependency Graph for Collaborative Bandits

In proposed research I.A, we have assumed that the mutual influence matrix $W$ is time-invariant and known to the system beforehand. This limits the applications of proposed algorithm when such information is not available. We plan to estimate $W$ from observed payoffs: users’ social connections (e.g., links in social networks, if observable) will be treated as prior information for estimating $W$; and coordinate descent can be used to iteratively refine $W$ and $\Theta$ for optimizing cumulated reward from all the users. The learned $W$ could then better reflect the dependency among users in determining the rewards. However, because the coupling of $W$ and $\Theta$ estimations, it complicates the confidence estimation in the online learning process with conventional bandit solutions and raises new scientific challenges in regret analysis. We plan to conduct the regret analysis based on the nature of coordinate descent: when estimating $\Theta$, we have an estimated noisy version of $W$, which has an unknown but constant discrepancy to the true $W$; then, when estimating $W$, we have an estimated noisy version of $\Theta$, which has an unknown but constant disparity to the true $\Theta$. In each of the two phrases, the regret bound can be derived based on the estimation of another parameter, and the overall regret is an integration of them across phases. And because the estimation quality of $\Theta$ and $W$ improves over iterations, we should expect an overall better than linear regret.

1.3 Proposed Research I.C: Factorization Bandits

Matrix factorization based collaborative filtering has become a standard practice in recommender systems [40, 39, 60] to capture the potential dependency among users or items. The basic idea of such solutions is to characterize both recommendation items and users by vectors of latent factors inferred from historical user-item preference patterns via low-rank matrix completion [12, 13], with an assumption that only a few factors contribute to an individual’s taste [40].

Some preliminary attempts have been made to perform online matrix factorization for collaborative filtering. Basically, multi-armed bandit algorithms [7, 5] are employed to control the exploration of currently less promising recommendations for user feedback, and 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 various factorization methods [70, 53, 37], given they only provide an index of candidate items for feedback acquisition. As a result, little is known about whether such combinations would lead to a converging recommendation performance nor would it ensure long-term optimality in theory, i.e., regret bound analysis.

I propose to address the aforementioned challenges by performing online interactive recommendation by placing a factorization-based bandit algorithm on each user in the system. Low-rank matrix completion is performed over an incrementally constructed user-item preference matrix, where an upper confidence bound (UCB) based item selection strategy is developed to balance the exploit/explore
trade-off during online feedback acquisition. To better conquer cold-start in recommendation, two special treatments are devised. First, observable contextual features are integrated with the estimated latent factors during matrix factorization. This improves recommendation when the number of candidate items is large, but the payoffs are interrelated, i.e., context-aware. Second, the dependence among users (e.g., social influence) is introduced to our bandit algorithm through a collaborative reward generation assumption [66]. It enables information sharing among the neighboring users while online learning, so as to help reduce the overall regret.

2 Research Challenge II: Bandit Learning in a Non-Stationary Environment

In a non-stationary environment, the reward distribution over arms varies over time because of the changes in the environment’s bandit parameter $\theta$. We consider abrupt changes in the environment [24, 30, 31], i.e., the ground-truth parameter $\theta$ changes arbitrarily at arbitrary time, but remains constant between any two consecutive change points:

$$r_0, r_1, \ldots, r_{t_{c_1}-1}, r_{t_{c_1}}, r_{t_{c_1}+1}, \ldots, r_{t_{c_2}-1}, \ldots, r_{t_{C_T}}, r_{t_{C_T}+1}, \ldots, r_T$$

where the change points $\{t_{c_j}\}_{j=1}^{T_f-1}$ of the underlying reward distribution and the corresponding bandit parameters $\{\theta_{c_j}\}_{j=0}^{T_f-1}$ are unknown to the learner. We only assume there are at most $T_f - 1$ change points in the environment up to time $T$, with $T_f \ll T$. To simplify the discussion, linear structure in $f(x_t, r_t)$ is postulated, but it can be readily extended to more complicated dependency structures, such as generalized linear models [23], without changing the design of our algorithm. Specifically, we have $r_t = f_{\theta_t}(x_{a_t}) = x_{a_t}^T \theta_t^* + \eta_t$, in which $\eta_t$ is Gaussian noise drawn from $\mathcal{N}(0, \sigma^2)$, and the superscript $*$ in $\theta_t^*$ means it is the ground-truth bandit parameter in the environment.

2.1 Proposed Research II.A: Dynamic Contextual Bandits Learning

Based on the piece-wise stationary property of the non-stationary environment, in any stationary period between two consecutive change points, the reward estimation error of a contextual bandit model trained on the observations collected from that period should be bounded with a high probability [1, 19]. Otherwise, the model’s consistent wrong predictions can only come from the change of environment. Based on this insight, we can evaluate whether the stationary assumption holds by monitoring a bandit model’s reward prediction quality over time. To reduce variance in the prediction error from one bandit model, we ensemble a set of models, by creating and abandoning them on the fly.

Specifically, we propose a hierarchical bandit algorithm, in which a master multi-armed bandit model operates over a set of slave contextual bandit models to interact with the changing environment. The master model monitors the slave models’ reward estimation error over time, which is referred to as ‘badness’ in this paper, to evaluate whether a slave model is admissible for the current environment. Based on the estimated ‘badness’ of each slave model, the master model dynamically discards out-of-date slave models or creates new ones. At each round $t$, the master model selects a slave model with the smallest lower confidence bound (LCB) of ‘badness’ to interact with the environment, i.e., the most promising slave model. The obtained observation $(x_{a_t}, r_{a_t})$ is shared across all admissible slave models to update their model parameters. The process is illustrated in Figure 2.

In order to guarantee the detectability of the changes, and reflects our insight how to detect them on the fly, We impose the following assumption about the non-stationary environment,

Assumption 1. For any two consecutive change points $t_{c_j}$ and $t_{c_{j+1}}$ in the environment, there exists $\Delta_j > 0$, such that when $t \geq t_{c_{j+1}}$ at least $\rho (0 < \rho \leq 1)$ portion of all the arms satisfy,

$$|x_{a_t}^T \theta_{c_{j+1}}^* - x_{a_t}^T \theta_{c_j}^*| > \Delta_j$$

(5)
Any contextual bandit algorithm [44, 23, 46, 66] can serve as our slave model. Due to the simplified linear reward assumption, we choose LinUCB [44] for the purpose in this paper; but our proposed algorithm can be readily adapted to any other choices of the slave model. This claim is also supported by our later regret analysis. As a result, we name our algorithm as Dynamic Linear Bandit with Upper Confidence Bound, or dLinUCB in short.

In the following, we first briefly describe our chosen slave model LinUCB. Then we formally define the concept of ‘badness’, based on which we design the strategy for creating and discarding slave bandit models. Lastly, we explain how slave models are selected and updated.

**Slave bandit model: LinUCB.** Each slave LinUCB model maintains all historical observations that the master model has assigned to it. Based on the assigned observations, a slave model \( m \) gets an estimate of user preference \( \theta_i(m) = \mathbf{A}_i^{-1}(m)\mathbf{b}_i(m) [44] \), in which \( \mathbf{A}_i(m) = \lambda \mathbf{I} + \sum_{t \in \mathcal{I}_{m,t}} \mathbf{x}_t \mathbf{x}_t^T \), \( \mathbf{I} \) is a \( d \times d \) identity matrix, \( \lambda \) is the coefficient for \( L2 \) regularization; \( \mathbf{b}_i(m) = \sum_{t \in \mathcal{I}_{m,t}} \mathbf{x}_t r_{a_t} \), and \( \mathcal{I}_{m,t} \) is an index set recording when the observations are assigned to the slave model \( m \) up to time \( t \). According to [1], with a high probability \( 1 - \delta \), the expected reward estimation error of model \( m \) is upper bounded:

\[
|\tilde{r}_a(m) - \mathbb{E}[r_a]| \leq B_t(m, a), \text{ in which } B_t(m, a) = \left( \sigma^2 \sqrt{d \ln(1 + \frac{1}{\lambda^2 \mathbf{x}_a^T \mathbf{x}_a})} + \sqrt{\lambda} \right) \|\mathbf{x}_a\| \mathbf{A}_i^{-1}(m).
\]

Based on the upper confidence bound principle [6], a slave model \( m \) takes an action using the following arm selection strategy:

\[
a_t(m) = \arg \max_{a \in \mathcal{A}} \left( \mathbf{x}_a^T \theta_i(m) + B_t(m, a) \right)
\]

**Slave model creation and abandonment.** For each slave bandit model \( m \), we define a binary random variable \( e_i(m) \) to indicate whether the slave model \( m \)'s prediction error at time \( i \) exceeds its confidence bound,

\[
e_i(m) := 1 \{ |\tilde{r}_i(m) - r_i(m)| > B_t(m, a_t) + \epsilon \}
\]

where \( \epsilon = \sqrt{2 \sigma \text{erf}^{-1}(\delta_1)} \) and \( \text{erf}^{-1}(\cdot) \) is the inverse of Gauss error function. \( \epsilon \) represents the high probability bound of Gaussian noise in the received feedback.

According to the concentration inequality in Chernoff Bound, if the environment stays stationary since the slave model \( m \) has been created, we have \( \mathbb{P}(e_i(m) = 1) \leq \delta_1 \), where \( \delta_1 \in (0, 1) \) is a hyper-parameter in \( B_t(m, a) \). Therefore, if we observe a sequence of consistent prediction errors from the slave model \( m \), it strongly suggests a change of environment, so that this slave model should be abandoned from the admissible set. Moreover, we introduce a size-\( \tau \) sliding window to only accumulate the most recent observations when estimating the expected error in slave model \( m \).
We define $\hat{e}_t(m) := \frac{\sum_{i=t-\tau(m)}^{t} e_i(m)}{\tau(m)}$, which estimates the ‘badness’ of slave model $m$ within the most recent period $\tau$ to time $t$, i.e., $\tau(m) = \min\{t - t_m, \tau\}$, in which $t_m$ is when model $m$ was created. Combining the concentration inequality in Chernoff Bound, we have the assertion that if in the period $[t - \tilde{\tau}(m), t]$ the stationary hypothesis is true, for any given $\delta_1 \in (0, 1)$ and $\delta_2 \in (0, 1)$, with a probability at least $1 - \delta_2$, the expected ‘badness’ of slave model $m$ satisfies,

$$\hat{e}_t(m) \leq E[e_t(m)] + \sqrt{\frac{\ln(1/\delta_2)}{2\tau(m)}} \leq \delta_1 + \sqrt{\frac{\ln(1/\delta_2)}{2\tau(m)}}$$

Eq (8) provides a tight bound to detect changes in the environment. If the environment is unchanged, within a sliding window the estimation error made by an up-to-date slave model should not exceed the right-hand side of Eq (8) with a high probability. Otherwise, the stationary hypothesis has to be rejected and thus the slave model $m$ should be discarded. Accordingly, if none of the slave models in the admissible bandit set satisfy this condition, a new slave bandit model should be created for this new environment. Specifically, the master bandit model controls the slave model creation and abandonment in the following way.

- **Model abandonment:** when the slave model $m$’s estimated ‘badness’ exceeds its upper confidence bound defined in Eq (8), i.e., $\hat{e}_t(m) > \delta_1 + \sqrt{\frac{\ln(1/\delta_2)}{2\tau(m)}}$, it will be discarded and removed from the admissible slave model set.

- **Model creation:** When no slave model’s estimated ‘badness’ is within its expected confidence bound, i.e., no slave model satisfies $\hat{e}_t(m) \leq \delta_1 + \sqrt{\frac{\ln(1/\delta_2)}{2\tau(m)}}$, a new slave model will be created. $\delta_1 \in [0, 1]$ is a parameter to control the sensitivity of dLinUCB, which affects the number of maintained slave models. When $\delta_1 = \delta_1$, the threshold of creating and abandoning a slave model matches and the algorithm only maintains one admissible slave model. When $\delta_1 < \delta_1$ multiple slave models will be maintained. The intuition is that an environment change is very likely to happen when all active slave models face a high risk of being out-of-date (although they have not been abandoned yet).

**Slave model selection and update.** At each round, the master bandit model selects one active slave bandit model to interact with the environment, and updates all active slave models with the acquired feedback accordingly.

### 2.1.1 Regret Analysis

**Theorem 2.** When Assumption 1 is satisfied with $\Delta \geq 2\sqrt{\lambda} + 2\epsilon$, if $\delta_1$ and $\tau$ in the proposed Algorithm are set according to Lemma 1 in [65], and $\delta_2$ is set to $\delta_2 \leq \frac{1}{2S_{\max}}$, with probability at least $(1 - \delta_1)(1 - \delta_2)(1 - \delta_2)$, the accumulated regret of dLinUCB satisfies,

$$R(T) \leq 2T_{\text{Lin}}(S_{\max}) + T\tau + \sqrt{\frac{4}{1 - \delta_2}}$$

Where $S_{\max}$ is the length of the longest stationary period up to $T$.

Briefly speaking, the proof of Theorem 2 is based on the following analysis: If change points can be perfectly detected, the regret of dLinUCB can be bounded by $\sum_{t=0}^{T-1} R_{\text{Lin}}(S_{c_t})$. However, additional regret may accumulate if early or late detection happens. With the following two Lemmas, we can bound the probability of early detection and late detection, based on which we can further bound the possible additional regret from early detection, and that from late detection.

Theorem 2 indicates with our model update and abandonment mechanism, each slave model in dLinUCB is ‘admissible’ in terms of upper regret bound. In the following, we further prove that maintaining multiple slave models and selecting them according to their LCB of ‘badness’ can further improve the regret bound.
2.1.2 Empirical Evaluation

We tested our proposed dLinUCB on a synthetic dataset and three real-world datasets, including Yahoo dataset, LastFM dataset and Delicious Dataset. The proposed algorithm is compared with the following baselines: LinUCB [44], a standard stationary linear contextual bandit algorithm and AdTS [30], which is a state-of-the-art contextual bandit algorithms for piece-wise stationary environment.

In simulation, we generate a size-$K$ ($K = 1000$) arm pool $\mathcal{A}$, in which each arm $a$ is associated with a $d$-dimensional feature vector $x_a \in \mathbb{R}^d$ with $\|x_a\|_2 \leq 1$. Similarly, we create a set of ground-truth bandit parameters $\theta^* \in \mathbb{R}^d$ with $\|\theta^*\|_2 \leq 1$, which are not disclosed to the learners. The ground-truth reward $r_a$ is corrupted by Gaussian noise $\eta \sim N(0, \sigma^2)$ before being fed back to the learners. The standard deviation of Gaussian noise $\sigma$ is set to 0.05 by default. To simulate an abruptly changing environment, after every $S$ rounds, we randomize $\theta^*$ with respect to the constraint that at least $\rho$ portion of arms in $\mathcal{A}$ satisfy $|x_a^T \theta^*_{t_j} - x_a^T \theta^*_{x_j+1}| > \Delta$. And by default, we set $S$ to 800, $\Delta$ to 0.7, and $\rho$ to 0.8. Accumulated regret is used to evaluate different algorithms and is reported in Figure 3. The bad performance of LinUCB illustrates the necessity of modeling the non-stationarity of the environment – its regret only converges in the first stationary period, and it suffers from an almost linearly increasing regret, which is expected according to our theoretical analysis. AdTS is able to detect and react to the changes in the environment, but it is slow in doing so and therefore suffers from a linear regret at the beginning of each stationary period before converging. dLinUCB, on the other hand, can quickly identify the changes and create corresponding slave models to capture the new reward distributions, which makes the regret of dLinUCB converge much faster in each detected stationary period. In Figure 3 we use the black and blue vertical lines to indicate the actual change points and the detected ones by dLinUCB respectively. It is clear that dLinUCB detects the changes almost immediately every time.

We compared all the algorithms on the large-scale clickstream dataset made available by the Yahoo Webscope program as described in proposed research I.A. Unbiased offline evaluation protocol proposed in [45] is used to compare different algorithms. CTR is used as the performance metric of all bandit algorithms. Following the same evaluation principle used in [44], we normalized the resulting CTR from different algorithms by the corresponding logged random strategy’s CTR. Since this dataset does not provide user identities, we followed [66] to cluster users into $N$ user groups and assume those in the same group share the same bandit parameter. From Figure 3 (b), we can find that both the personalized and non-personalized variants of dLinUCB achieved significant improvement compared with all baselines. It is worth noticing that uniform-dLinUCB obtained around 50% improvement against uniform-LinUCB, 15% against N-LinUCB, and 25% against CLUB. Clearly assuming all the users share the same preference over the recommendation candidates is very restrictive, which is confirmed by the improved performance from the personalized version over the non-personalized version of all bandit algorithms.

The LastFM dataset is extracted from the music streaming service Last.fm, and the Delicious dataset is extracted from the social bookmark sharing service Delicious. They were made available on the HotRec 2011 workshop. The LastFM dataset contains 1892 users and 17632 items (artists). The Delicious dataset contains 1892 users and 17632 items (artists). The LastFM dataset is extracted from the music streaming service Last.fm, and the Delicious dataset is extracted from the social bookmark sharing service Delicious. They were made available on the HotRec 2011 workshop. The LastFM dataset contains 1892 users and 17632 items (artists).
dataset contains 1861 users and 69226 items (URLs). Following the settings in [16], we pre-processed these two datasets in order to fit them into the contextual bandit setting. We followed [31] to simulate a non-stationary environment: we ordered observations chronologically inside each user, and built a single hybrid user by merging different users. Hence, the boundary between two consecutive batches of observations from two original users is treated as the preference change of the hybrid user. We created a semi-oracle algorithm named as OracleLinUCB, which knows where the boundary is in the environment and resets LinUCB at each change point. Normalized rewards from these two datasets are reported in Figure 4 (a) & (b), in which the vertical lines are the actual change points in the environment and the detected points by dLinUCB. Since OracleLinUCB knows where the change is ahead of time, its performance can be seen as optimal. On LastMF, the observations are denser per user group, so that dLinUCB can almost always correctly identify the changes and achieve quite close performance to this oracle. We qualitatively studied those created slave models to investigate what kind of stationarity they have captured. On the LastFM dataset, each user is associated with a list of tags he/she gave to the artists. The tags are usually descriptive and reflect users’ preference on music genres or artist styles. In each slave model, we use all the tags from the users being served by this model to generate a word cloud. Figure 5 are four representative groups identified on LastFM, which clearly correspond to four different music genres – rock music, metal music, pop music and hip-hop music. dLinUCB recognizes those meaningful clusters purely from user click feedback.

2.2 Proposed Research II.B: Context-Dependent Dynamic Ensemble of Contextual Bandits

In a contextual bandit setting, the reward changes caused by environment changes are context dependent. To realize the unique property that in a contextual bandit setting the changes of reward distribution are context-dependent, we categorize the arms into two types, change-invariant and change-sensitive, between any two stationary periods. Specifically, for stationary periods i and j with their ground-truth bandit parameters \( \theta^*_i \) and \( \theta^*_j \), we refer to arm \( a \) that satisfies \( |x^T_a \theta^*_i - x^T_a \theta^*_j| \leq \Delta_L \) as a change-invariant arm; otherwise as a change-sensitive arm. \( \Delta_L \) is a parameter to relax the requirement that the change has to be completely orthogonal to the context vector. This makes our definition more general than those in [67, 48] for recognizing the existence of change-invariant arms with infinitesimal reward shift.
The possible existence of change-invariant arms in this environment setting suggests us to reuse the bandit models estimated for those earlier periods, such that more accurate reward estimation on such arms can be achieved sooner so as to obtain reduced regret in this new period. However, as the changes of environment are unknown to the learner, three challenges have to be addressed: 1) how to recognize the change-invariant arms at current iteration; 2) when to create new bandit models to account for the change-sensitive arms in the new environment; and 3) which arm to choose given multiple bandit models might exist at the same time. To avoid any potential ambiguity in our later discussion, we refer to the contextual bandit models created for reward estimation as bandit experts.

We address the first two challenges by creating a companion bandit model for each bandit expert to monitor its reward estimation quality. We refer to this companion bandit model as a bandit auditor. In particular, each bandit auditor evaluates whether the monitored bandit expert is admissible to make an accurate reward estimation for a given arm with respect to potential environment changes. An admissible bandit expert indicates that with a high probability, 1) no change has happened since the creation of this bandit expert, or 2) the environment changed but the arm is change-invariant between this bandit expert’s estimated reward distribution and the current period’s underlying reward distribution. When no admissible bandit expert exists for a given arm, it is thus highly likely to be a change-sensitive arm in a new environment, and a new bandit expert is needed. To address the third challenge, at each round of interaction, an arm is chosen by the upper confidence bound of reward estimation based on an ensemble of all its admissible bandit experts. The acquired feedback is used to update the corresponding bandit experts and their auditors.

**Bandit Expert.** Define \( t_m \) as the time when bandit expert \( m \) is created. Each bandit expert \( m \) maintains an estimated bandit parameter \( \theta_t(m) \) for the stationary period at \( t_m \). Define \( I_t^m \) as a set of timestamps when observation \((x_{a,t}, r_{a,t,i})\) is assigned to the bandit expert \( m \) for model update till time \( t \). \( r_{a,i} \) is the observed reward on arm \( a \) at time \( i \). Because of our linear reward structure, \( \theta_t(m) \) can be readily estimated by \( \hat{\theta}_t(m) = A_t^{-1}(m) b_t(m) \), in which \( A_t(m) = \lambda I + \sum_{i \in I_t^m} x_{a,i} x_{a,i}^T \), \( b_t(m) = \sum_{i \in I_t^m} x_{a,i} r_{a,i} \).

**Bandit Auditor.** Denote the reward estimation error of bandit expert \( m \) on arm \( a \) at time \( t \) as \( e_{a,t}(m) = \hat{r}_{a,t}(m) - r_{a,t} \), in which \( \hat{r}_{a,t}(m) = x_{a,t}^T \hat{\theta}_t(m) \). We have \( \mathbb{E}[e_{a,t}(m)] = x_{a,t}^T (\theta_t(m) - \Theta_s) \), which is referred as ‘badness’ of \( m \) on arm \( a \) at time \( t \) and leads to a linear structure for badness estimation. We create a new bandit model with the target parameter for estimation as \( \beta^*_t(m) = \hat{\theta}_t(m) - \Theta_s \), and refer to it as the bandit auditor of bandit expert \( m \). We maintain and update the bandit auditors in a similar manner as we have taken in bandit experts. Denote \( I_t^s \) as a set of timestamps when observation \((x_{a,i}, e_{a,i}(m))\) is assigned to the bandit auditor for bandit expert \( m \) up to time \( t \). The bandit auditor estimates \( \beta^*_t(m) \) by \( \hat{\beta}_t(m) = C_t^{-1}(m) d_t(m) \), in which \( C_t(m) = \lambda I + \sum_{i \in I_t^s} x_{a,i} x_{a,i}^T \), and \( d_t(m) = \sum_{i \in I_t^s} x_{a,i} e_{a,i}(m) \). Intuitively, the bandit auditor for \( m \) evaluates whether an arm \( a \) at time \( t \) is change-invariant to the reward distribution specified by \( \beta^*_t \) and \( \Theta_s \). The definition of badness requires us to update \( e_{a,t}(m) \) of all observations in \( I_t^s \) whenever \( \hat{\theta}_t(m) \) is updated or \( \Theta_s \) is changed. But as the environment change is unknown to the learner, we decide to only accumulate the most recent \( \tau \) observations in \( I_t^s \) for auditor update. We prove a high probability bound of each bandit auditor’s badness estimation as \( |e_{a,t}(m) - \mathbb{E}[e_{a,t}(m)]| \leq B_{a,t}^\beta(m), \) where \( B_{a,t}^\beta(m) = \left( \sigma^2 \sqrt{d \ln(\frac{\lambda + |I_t^s(m)|}{\lambda \Delta^2})} + \sqrt{\lambda} \right) \|x_a\| C_t^{-1}(m) \).

**Bandit Expert Selection.** Based on our badness definition, we have \( |e_{a,t}(m) - \mathbb{E}[e_{a,t}(m)]| + B_{a,t}^\beta(m) \leq |x_a^T \hat{\theta}_t(m) - x_a^T \Theta_s| + |x_a^T \Theta_s - x_a^T \Theta^*_s| + B_{a,t}^\beta(m) \). The first part on the right-hand side of this inequality is the reward estimation quality of bandit expert \( m \), which can be bounded by \( |x_a^T \hat{\theta}_t(m) - x_a^T \Theta_{s,t}| \leq B_{a,t}^\beta(m, a) \), where \( B_{a,t}^\beta(m, a) = \left( \sigma^2 \sqrt{d \ln(\frac{\lambda + |I_t^s(m)|}{\lambda \Delta^2})} + \sqrt{\lambda} \right) \|x_a\| A_t^{-1}(m) \). We should note that this bound is different from those for classical linear bandit algorithms. The second part on the right-hand
side can be bounded by $\Delta_L$, if no change has happened since $t_m$ or the arm $a$ is change-invariant between $t_m$ and $t$. As a result, the condition $|\hat{e}_{a,t}(m)| < B_{a,t}^0(m) + B_{a,t}^{\beta}(m) + \Delta_L$ determines if the bandit expert $m$ is admissible to arm $a$ at time $t$. When there is no admissible bandit expert for arm $a$, a new bandit expert needs to be created to account for the change of environment.

**Arm Selection and Model Update.** We appeal to the Upper Confidence Bound (UCB) principle [44, 1] to select an arm from the candidate arm pool. With the above bandit expert selection strategy, for arm $a$ at time $t$, we might collect a set of admissible bandit experts $M_{a,t}$, and each admissible bandit expert $m$ gives us an upper confidence bound of reward estimation for arm $a$: $\text{UCB}_{a,t}(m) = x^T_{a,t} \hat{\theta}_t(m) + B_{a,t}^0(m)$.

Therefore, we need to ensemble $\text{UCB}_{a,t}(m)$ from multiple bandit experts. We propose two strategies for this operation, each of which has its own advantages.

**Option 1:** Average ensemble. We compute $\text{UCB}_{a,t} = \frac{1}{|M_{a,t}|} \sum_{m \in M_{a,t}} \text{UCB}_{a,t}(m)$. This ensemble helps reduce variance in reward estimation among the admissible bandit experts.

**Option 2:** Lower Confidence Bound (LCB) of badness. We first select a bandit expert from $M_{a,t}$ according to the LCB of its auditor’s estimated badness: $\tilde{m}_{a,t} = \arg \min_{m \in M_{a,t}} (\hat{e}_{a,t}(m) - B_{a,t}^\beta(m))$; and then compute the UCB of arm $a$ using the selected bandit expert $\tilde{m}_{a,t}$. The LCB of badness balances exploration and exploitation in badness estimation across admissible bandit experts, as the badness is also estimated on the fly in each bandit auditor.

Once the feedback $(x_{a_t}, r_{a_t})$ is obtained from the environment on the selected arm $a_t$, the bandit experts and auditors in this arm’s admissible model set will be updated.

### 2.3 Proposed Research II.C: Collaborative Bandit Learning in a Non-Stationary Environment

Sharing information among learning agents is beneficial in many learning scenarios, including collaborative bandit learning [66, 16], multi-task learning [14, 15] and general online learning [72]. In a collaborative non-stationary environment, users are connected and influenced by each other through network, and at the same time users’ preference evolve over time under the social influence. In this case, bandit learning for one user could potentially benefit from previous experience of another learner, which indicates that dynamic bandit learning for users independently may waste such information and lead to sub-optimal solution. In this research, I propose to utilize information sharing among learning agents of different users to better conquer the non-stationary environment. Based on the bandit expert, bandit auditor, and expert badness concept in proposed research II.A and II.B, we enable information sharing by creating a globally shared bandit expert pool for connected users. The globally shared bandit expert pool contains all bandit experts of the connected users on different detected stationary periods. Each bandit expert is associated with a set of localized bandit auditors (localized for concerned each user). At each round of interaction, for a particular user, with the help of the localized bandit auditors, the algorithm decides whether each of the bandit expert in the global expert pool is admissible to make a reward prediction for this concerned user. In this case, past experience from different users can be shared through the globally shared bandit experts. New bandit experts will be added in the globally shared expert pool as new environment is detected for each individual user. The localized bandit auditors monitors whether the learnt experience is helpful in the potentially changing environment. In addition, with a mild assumption about how user preference changes are triggered by the social connection, prior information can be further utilized to predict the popularity of the globally shared bandit experts, which can help the bandit expert selection for each individual user.
3 Research Challenge III: Online Learning from Implicit and Incomplete Feedback

3.1 Proposed Research III.A: Bandit Learning with Implicit Feedback

Contextual bandit algorithms [5, 44, 43] provide modern information service systems an effective solution to adaptively find good mappings between items and users according to user feedback. However, the most dominant form of user feedback in such systems is implicit feedback, such as clicks, which is known to be biased and incomplete about users’ evaluation of system’s output [38, 32]. In this work, we propose to learn contextual bandits with user click feedback, and model such implicit feedback as a composition of user result examination and relevance judgment. Examination hypothesis [22], which is a fundamental assumption in click modeling, postulates that a user clicks on a system’s returned result if and only if that result has been examined by the user and it is relevant to the user’s information need at the moment. Because a user’s examination behavior is unobserved, we model it as a latent variable, and realize the examination hypothesis in a probabilistic model. We define the conditional probabilities of result examination and relevance judgment via logistic functions over the corresponding contextual features. To perform model update, we take a variational Bayesian approach to develop closed form approximation to the posterior distribution of model parameters on the fly. This approximation also paves the way for an efficient Thompson sampling strategy for arm selection in bandit learning.

Specifically, we model user examination through a binary latent variable \( E^a_t \) and assume that the context vector \( x^a_{C,t} \) of arm \( a \) can be decomposed into \( (x^a_{C,t}, x^a_{E,t}) \), where the dimension of \( x^a_{C,t} \) and \( x^a_{E,t} \) are \( d_C \) and \( d_E \) respectively. Accordingly, users’ result examination and relevance judgment decisions are assumed to be governed by a conjecture of \( (x^a_{C,t}, x^a_{E,t}) \) and the bandit parameter \( \theta^* = (\theta^*_C, \theta^*_E) \). In the rest of this section, when no ambiguity is introduced, we drop the index \( a \) to simplify the notations. As a result, we make the following generative assumption about an observed click \( C_t \) on arm \( a_t \),

\[
\begin{align*}
\mathbb{P}(C_t = 1 | E_t = 0, x_{C,t}) &= 0 \\
\mathbb{P}(C_t = 1 | E_t = 1, x_{C,t}) &= \rho(x^T_{C,t} \theta^*_C) \\
\mathbb{P}(E_t = 1 | x_{E,t}) &= \rho(x^T_{E,t} \theta^*_E)
\end{align*}
\]

where \( \rho(x) = \frac{1}{1 + e^{-x}} \). Based on this assumption, we have \( \mathbb{E}[C_t | x_t] = \rho(x^T_{C,t} \theta^*_C) \rho(x^T_{E,t} \theta^*_E) \). As a result, the observed click feedback \( C_t \) is a sample from this generative process. On the algorithm side, the learner will to estimate the bandit parameters \( \theta^*_C \) and \( \theta^*_E \) based on its interactively obtained click feedback \( \{C_i\}_{i=1}^t \) over time.

3.2 Proposed Research III.B: Proactive User-System Interaction

In our proposed CoLin algorithm and other existing collaborative bandit learning solutions [16], the order of users for a system to interact with is pre-defined. The system only needs to choose the most informative action for feedback during online learning. However, given users are no longer independent in collaborative learning, if a system could start with the users whose feedback can mostly reduce the system’s uncertainty about the other users, less exploration would be needed to optimize the system’s utility or all users. To the best of our knowledge, there is no existing work considers such active feedback acquisition in online learning problems. Our insight to solve this problem is based on the UCB criterion exploited in our CoLin algorithm: we should explore the combinatorial space of candidate actions and users that maximizes the expected reward with respect to a fixed confidence level of estimated bandit parameters.

Intuitively, within a fixed confidence set, this new algorithm prefers the combination of user and action that leads to a maximized payoff. As more feedback becomes available about such combinations,
the confidence set of bandit parameter estimation shrinks and eventually converges to the true parameters with high probability. The scientific challenge is to prove the resulting upper regret bound of the new algorithm: we are expecting another regret reduction. And starting from this new UCB-style algorithm for user selection, we plan to further refine the bandit algorithm by considering more practical constraints in user selection, e.g., urgency of request and trustworthiness of feedback, to provide more practical solutions of proactive user selection.
Chapter 4

Summary: Impact Statement of Proposed Work and Time-line

1 Impact Statement

The proposed research tries to solve several key challenges of online learning in real world systems comprehensively, including challenges from user-user interaction, challenges from users’ temporal behavior and challenges from user-system interaction. The significance of proposed research is that they incorporate sophisticated statistical modeling into online learning of user behavior and preference, which extends the application of bandit learning algorithms to more complex practical scenarios. Rigorous theoretical analysis and extensive empirical evaluation already indicate the proposed approaches’ applicability in various contexts. This line of research will establish a new foundation of contextual bandit algorithms and trigger new algorithmic and theoretical development of interactive online learning. By combining the proposed research, an information service system can provide right information to the right user at the right time, and improve users satisfaction in the long run. This will shift the paradigm for human-machine interaction from passive response to proactive learning. More generally, the proposed research can benefit a wide spectrum of important real-world applications, including research result ranking [54], personalized recommendation [44], crowdsourcing [33], clinical trials in healthcare [61], sequential treatment design in psychology and many more.

2 Proposed Time-Line

For research challenge I, I have finished the proposed research I.A [66] and I.C [63]. Some preliminary theoretical analysis that supporting the proposed research I.B have been conducted. I plan to finish I.B by December 2018.

For research challenge II, I have finished the proposed research II.A [65]. For proposed research II.B, solution has been formulated, and preliminary regret analysis and empirical evaluation have been performed accordingly. Further improvement might be needed according to the feedback from peer-reviewed conferences. I plan to finish proposed research II.B by October 2018. Preliminary attempts about the algorithm design have been conducted about proposed research II.C. I plan to finish the algorithm design and empirical evaluation of it by December 2018 and finish the theoretical analysis of it by February 2019 or May 2019 depending on the progress.

For proposed research III.A, preliminary algorithm design, empirical evaluation and theoretical analysis have been conducted together with the other collaborators. Further improvement will be conducted if needed. I plan to finish this line of proposed research by February 2019. For proposed research III.B, I plan to start working on it by February 2019, finish the algorithm design by May 2019, and finish the empirical evaluation and theoretical analysis by October 2019.

I plan to start writing thesis from October 2019 and finish thesis writing by December 2019.
Bibliography


N. Hariri, B. Mobasher, and R. Burke. Adapting to user preference changes in interactive recommendation.


