Hypothesis Transfer in Bandits by Weighted Models

Steven Bilaj(✉)[1], Sofien Dhouib[1], and Setareh Maghsudi[1]

Eberhard Karls University of Tübingen, Tübingen, Germany
{steven.bilaj, sofiane.dhouib, setareh.maghsudi}@uni-tuebingen.de

Abstract. We consider the problem of contextual multi-armed bandits in the setting of hypothesis transfer learning. That is, we assume having access to a previously learned model on an unobserved set of contexts, and we leverage it in order to accelerate exploration on a new bandit problem. Our transfer strategy is based on a re-weighting scheme for which we show a reduction in the regret over the classic Linear UCB when transfer is desired, while recovering the classic regret rate when the two tasks are unrelated. We further extend this method to an arbitrary amount of source models, where the algorithm decides which model is preferred at each time step. Additionally we discuss an approach where a dynamic convex combination of source models is given in terms of a biased regularization term in the classic LinUCB algorithm. The algorithms and the theoretical analysis of our proposed methods substantiated by empirical evaluations on simulated and real-world data.

Keywords: Multi-Armed Bandits · Linear Reward Models · Recommender Systems · Transfer Learning.

1 Introduction

The *multi-armed bandit* problem (MAB) [27,22,7] revolves about maximizing the reward collected by playing actions from a predefined set, with uncertainty and limited information about the observed payoff. At each round, the bandit player chooses an arm according to some rule that balances the exploitation of the currently available knowledge and the exploration of new actions that might have been overlooked while being more rewarding. This is known as the exploration-exploitation trade-off. MAB's find applications in several areas [6], notably in recommender systems [16,33,18,13]. In these applications, the number of actions to choose from can grow very large, and it becomes provably detrimental to the algorithm's performance to ignore any side information provided when playing an action or dependence between the arms [4]. Considering such information defines the *Stochastic Contextual Bandits* [14,16,8,1] setting, where playing an action outputs a context-dependent reward, where a context can correspond to a user's profile and/or the item to recommend in recommender system applications. Hence, less exploration is required as arms with correlating context vectors share information, thus further reducing uncertainty in the reward estimation. This ultimately led to lower regret bounds and improved performance [1].

While the stochastic contextual bandit problem solves the aforementioned issues, it disregards the possibility of learning from previously trained bandits. For instance, assume a company deploys its services in a new region. Then it would waste the information it has already learned from its previous recommending experience if it is not leveraged to accelerate the recognition of the new users' preferences. Such scenarios have motivated transfer learning for bandits [24,18,26,13], which rely on the availability of contexts of the previously learned tasks to the current learner. However, regarding a setup where context vectors correspond to items which have been selected by a user, privacy issues are encountered in healthcare applications [25,21] for instance, the aim being to recommend a treatment based on a patient's health state. Indeed, accessing the contexts of the previous tasks entails the history of users' previous activities. Moreover, in engineering applications such as scheduling of radio resources [2], storage issues [19,17,29] might arise when needing access to the context history of previous tasks. These problems would render algorithms depending on previous tasks' contexts inapplicable.

In this work, we aim to reduce exploration by exploiting knowledge from a previously trained contextual bandit accessible only through its parameters, thus accelerating learning if such model is related to the one at hand, and ultimately decreasing the regret. We extend this idea by including an arbitrary amount of models increasing the likelihood of including useful knowledge. To summarize our contributions, we propose a variation of the Linear Upper Confidence Bound ($LinUCB$) algorithm, which has access to previously trained models called source models. The knowledge transfer takes place by using an evolving convex combination of sources models and a LinUCB model, called a target model, estimated with the collected data. The combination's weights are updated according to two different weighting update strategies which minimize the required exploration factor and consecutively the upper regret bound, while also taking a lack of information into consideration. Our regret bound is at least as good as the classic LinUCB one [1], where the improvement depends on the quality of the source models. Moreover, we prove that if the source model used for transfer is not related to our problem, then it will be discarded early on and we recover the LinUCB regret rate. In other words, our algorithm is immune against negative transfer. We test our algorithm on synthetic and real data sets and show experimentally how the overall regret improves on the classic model.

The rest of the paper is organized as follows. We discuss related work in Section 2 and formulate our problem in Section 3, then we provide and analyse our weighting solution in Section 4. This is followed by an extension to the case where one has access to more than one trained model in Section 5. Finally, the performance of our algorithm is assessed in Section 6.

2 Related Work

We hereby discuss two families of contributions related to ours, namely transfer for multi-armed bandits, and hypothesis transfer learning.

Transfer for MAB's To the best of our knowledge, tUCB [5] is the first algorithm to tackle transfer in an MAB setting. Given a sequence of bandit problems picked from a finite set, it uses a tensor power method to estimate their parameters in order to transfer knowledge to the task at hand, leading to a substantial improvement over UCB. Regarding the richer contextual MAB setting, MT-LinUCB [24] reduces the confidence set of the reward estimator by using knowledge from previous episodes. More recently, transfer for MAB's has been applied to recommender systems [18,13], motivated by the cold start problem where a lack of initial information requires more exploration at the cost of higher regret. The TCB algorithm [18] assumes access to correspondence knowledge between the source and target tasks, in addition to contexts, and achieves a regret of $O(d\sqrt{n \log n})$ as in the classic LinUCB case, with empirical improvement. The same regret rate holds for the T-LinUCB algorithm [13], which exploits prior observations to initialize the set of arms, in order to accelerate the training process. The main difference of our formulation with respect to the previous ones is that we assume having access only the the preference vectors of the previously learnt tasks, without their associated contexts, which goes in line with the Hypothesis Transfer Learning setting. Even with such a restriction, we keep the LinUCB regret rate and we show that the regret is lower in the case source parameters that are close to those of the task at hand.

Hypothesis Transfer Learning Using previously learned models in order to improve learning on a new task defines the hypothesis transfer learning scenario, also known as model reuse or learning from auxiliary classifiers. Some lines of work consider building the predictor of the task at hand as the sum of a source one (possibly a weighted combination of different models) and the one learned from the available data points [30,10,28]. Such models were thoroughly analyzed in [11,12,20] by providing performance guarantees. The previously mentioned additive form of the learned model was further studied and generalized to a large family of transformation functions in [9]. In online learning, the pioneering work of [31] relies on a convex combination instead of a sum, with adaptive weights. More recently, the `Condor` algorithm [32] was proposed and theoretically analyzed to handle the concept drift scenario, relying on biased regularization w.r.t. a convex combination of source models. Our online setting involves transfer with decisions over a large set of alternatives at each time step, thus it becomes crucial to leverage transfer to improve exploration. To this end, we use a weighting scheme inspired by [31] but that relies on exploration terms rather than on how the models approximate the rewards.

3 Problem Formulation

We consider a contextual bandit setting in which at each time k, playing an action a from a set $\mathcal{A}$ results in observing a context vector $\mathbf{x}_{a_k} \in \mathbb{R}^d$ assumed to satisfy $\|\mathbf{x}_{a_k}\| \leq 1$, in addition to a reward $r(k)$. We further define the matrix induced norm: $\|\mathbf{x}\|_{\mathbf{A}} \coloneqq \sqrt{\mathbf{x}^T \mathbf{A} \mathbf{x}}$ for any vector $\mathbf{x} \in \mathbb{R}^d$ and any matrix $\mathbf{A} \in \mathbb{R}^{d \times d}$. The classical case aims to find an estimation $\hat{\boldsymbol{\theta}}$ of an optimal bandit parameter $\boldsymbol{\theta}^* \in \mathbb{R}^d$ which determines the rewards r of each arm with context vector $\mathbf{x}_a$ in a linear fashion $r = \mathbf{x}_a^T \boldsymbol{\theta}^* + \epsilon$ up to some σ-subgaussian noise ϵ. The decision at time k is made according to an upper confidence bound (UCB) associated to $\hat{\boldsymbol{\theta}}(k)$:

$$a_k = \arg\max_{a \in \mathcal{A}} \mathbf{x}_a^T \hat{\boldsymbol{\theta}}(k) + \gamma \sqrt{\mathbf{x}_a^T \mathbf{A}^{-1}(k) \mathbf{x}_a}, \tag{1}$$

where $\gamma > 0$ is a hyperparameter estimated through the derivation of the UCB later and $\mathbf{A}(k) \coloneqq \lambda \mathbf{I}_d + \sum_{k'=1}^{k} \mathbf{x}_{a_{k'}} \mathbf{x}_{a_{k'}}^T$. The latter term in the sum (1) represents the exploration term which decreases the more arms are explored. $\hat{\boldsymbol{\theta}}(k)$ is computed through regularized least-squares regression with regularization parameter $\lambda > 0$: $\hat{\boldsymbol{\theta}}(k) = \mathbf{A}^{-1}(k) \mathbf{D}^T(k) \mathbf{y}(k)$, with $\mathbf{D}(k) = [\mathbf{x}_{a_i}^T]_{i \in \{1,\dots,k\}}$ and $\mathbf{y}(k) = [r(i)]_{i \in \{1,\dots,k\}}$ as the concatenation of selected arms' context vectors and corresponding rewards respectively. We alter this decision making approach with the additional use of a previously trained source bandit. Inspired by [31], we transfer knowledge from one linear bandit model to another by a weighting approach. We denote the parameters of the source bandit by $\boldsymbol{\theta}_S \in \mathbb{R}^d$. The bandit at hand's parameters are then estimated as:

$$\hat{\boldsymbol{\theta}} = \alpha_S \boldsymbol{\theta}_S + \alpha_T \hat{\boldsymbol{\theta}}_T(k), \tag{2}$$

with weights $\alpha_S, \alpha_T \geq 0$ satisfying $\alpha_S + \alpha_T = 1$. More important is how the exploration term changes and how it affects the classic regret bound. From [1] we know that the upper bound of the immediate regret in a linear bandit algorithm directly depends on the exploration term of the UCB. We aim to reduce the required exploration with the use of the source bandits knowledge, in order to accelerate the learning process as well as reducing the upper regret bound. For the analysis we consider the pseudo-regret [3] defined as:

$$R(n) = n \max_{a \in \mathcal{A}} \mathbf{x}_a^T \theta^* - \sum_{k=1}^{n} \mathbf{x}_{a_k}^T \boldsymbol{\theta}^*. \tag{3}$$

Our goal is to prove that this quantity is reduced if the source bandit is related to the one at hand, whereas its rate is not worsened in the opposite case.

4 Weighted Linear Bandits

The model we use features dynamic weights, thus at time k, we use the following model for our algorithm:

$$\hat{\boldsymbol{\theta}}(k) = \alpha_S(k)\boldsymbol{\theta}_S + \alpha_T(k)\hat{\boldsymbol{\theta}}_T(k), \tag{4}$$

with $\hat{\boldsymbol{\theta}}_T(k)$ being updated like in the classic LinUCB case [1] and $\boldsymbol{\theta}_S$ remaining constant. To devise an update rules of the weights, we first re-write the new UCB expression as:

$$\text{UCB}(a) = \mathbf{x}_a^T \left(\alpha_S(k)\boldsymbol{\theta}_S + \alpha_T(k)\hat{\boldsymbol{\theta}}_T(k)\right) + \left(\alpha_S(k)\gamma_S + \alpha_T(k)\gamma_T\right) \|\mathbf{x}_a\|_{\mathbf{A}^{-1}}, \tag{5}$$

with $\gamma_S \geq \|\boldsymbol{\theta}^* - \boldsymbol{\theta}_S\|_{\mathbf{A}(k)}$ and $\gamma_T \geq \left\|\boldsymbol{\theta}^* - \hat{\boldsymbol{\theta}}_T(k)\right\|_{\mathbf{A}(k)}$ as confidence set bounds for the source bandit and target bandit respectively. We retrieve the classic case by setting $\alpha_S(k)$ to zero *i.e.* erasing all influence from the source. The confidence set bound γ_T has already been determined in [1].
As mentioned in section 3 we aim to reduce the required exploration in order to reduce the upper regret bound. Thus we select the weights such that the exploration term in (5) is minimized.

4.1 Weighting Update Strategies

We want to determine the weights after each time step such that:

$$\alpha_S, \alpha_T = \underset{\substack{\alpha'_S, \alpha'_T \geq 0 \\ \alpha'_S + \alpha'_T = 1}}{\arg\min}\ \alpha'_S\gamma_S + \alpha'_T\gamma_T. \tag{6}$$

The above minimization problem is solved for:

$$\alpha_S = \mathbb{1}_{\gamma_S \leq \gamma_T}, \quad \alpha_T = 1 - \alpha_S. \tag{7}$$

This strategy would guarantee an upper regret bound at least as good as the LinUCB bound in [1] as will be shown in the analysis section later. However, without any knowledge of the relation between source and target tasks, our upper bound on the confidence set of the source bandit is rather loose:

$$\|\boldsymbol{\theta}^* - \boldsymbol{\theta}_S\|_{\mathbf{A}(k)} = \sqrt{\lambda U^2 + \|\mathbf{D}(k)(\boldsymbol{\theta}^* - \boldsymbol{\theta}_S)\|_2^2} \quad \leq \sqrt{4\lambda + \|\overline{\mathbf{y}}(k) - \mathbf{y}_S(k)\|_2^2},$$

with $\|\theta^* - \theta_S\|_2 = U$, $\mathbf{y}_S$ as the concatenation of the source estimated rewards and $\overline{\mathbf{y}}$ as the concatenation of the observed mean rewards for each arm. Naturally after every time step, each entry in $\overline{\mathbf{y}}$ corresponding to the latest pulled arm needs to be updated to their mean value. The mean values are taken in order to cancel out the noise term in the observations. Also, we have $U \leq 2$ in case the vectors show in opposing directions and we additionally assume that $\|\boldsymbol{\theta}^*\|, \|\boldsymbol{\theta}_S\| \leq 1$. An upper bound on the confidence set γ_T of the target bandit has been determined in [1]:

$$\gamma_T = \sqrt{d\log\left(1 + \frac{k}{d\lambda}\right) + \log\left(\frac{1}{\delta^2}\right)}. \tag{8}$$

As such, γ_T grows with $\sqrt{\log(k)}$ and later on in the analysis we show if $\boldsymbol{\theta}_S \neq \boldsymbol{\theta}^*$ then an upper bound on γ_S grows with at least $\sqrt{k}$. Consequently, in theory there is some point in time where γ_S will outgrow γ_T, meaning that the source bandit will be discarded. As already mentioned, our estimation of γ_S can be loose due to our lack of information on the euclidean distance term U, thus we potentially waste a good source bandit with this strategy. Additionally we would only use one bandit at a time this way instead of the span of two bandits for example. Alternatively we can adjust the strategy in (6) by adding a regularization term in the form of KL-divergence. By substituting $\alpha_T = 1 - \alpha_S$ we get:

$$\alpha_S(k+1) = \underset{\alpha_S \in [0,1]}{\arg\min} \left\langle \begin{pmatrix} \alpha_S \\ 1-\alpha_S \end{pmatrix}, \begin{pmatrix} \gamma_S \\ \gamma_T \end{pmatrix} \right\rangle + \frac{\mathrm{KL}(\boldsymbol{\alpha} \| \boldsymbol{\alpha}(k))}{\beta}, \tag{9}$$

with $\boldsymbol{\alpha} \coloneqq (\alpha_S, 1-\alpha_S)^T$ being a vector containing both weights. The addition of the KL divergence term forces both weights to stay close to their previous value, where $\beta > 0$ is a hyper parameter controlling the importance of the regularization. Problem (9) is solved for:

$$\alpha_S(k+1) = \frac{1}{1 + \frac{1-\alpha_S(k)}{\alpha_S(k)} \exp(\beta(\gamma_S - \gamma_T))}, \tag{10}$$

which is a softened version of our solution in (7), but in this case the source bandit will not be immediately discarded if the upper bound on its confidence set becomes larger than the target bandit's.

4.2 Analysis

We are going to analyse how the upper regret bound changes, within our model in comparison to [1]. All proofs are given in the appendix. First we bound the regret for the hard update approach, not including the KL-divergence term in (7):

Theorem 1. *Let* $\{\mathbf{x}_{a_k}\}_{k=1}^N$ *be sequence in* $\mathbb{R}^d$, $U \coloneqq \|\boldsymbol{\theta}_S - \boldsymbol{\theta}^*\|$ *and* R_T *be the classic regret bound of the linear model [1]. Let* $m \coloneqq \min(\kappa, n)$ *and* $\delta \leq \exp(-2\lambda)$. *Then, with a probability at least* $1-\delta$, *the regret of the hard update approach for the weighted LinUCB algorithm is bounded as follows:*

$$R(n) \leq U\sqrt{8md\log\Big(1 + \frac{m}{d\lambda}\Big)(\lambda + m)} + R_T(n) - R_T(m) \leq R_T(n) \tag{11}$$

with κ *satisfying:*

$$\kappa = \left\lfloor 2\left[d\left(\frac{1}{U^2} - \lambda\right) + \lambda\left(\frac{2}{U^2} - \frac{1}{2}\right)\right]\right\rfloor. \tag{12}$$

The value for κ essentially gives a threshold such that we have $\gamma_S < \gamma_T$ for every $k < \kappa$. As expected, for better sources *i.e.* low values U, κ increases meaning the source is viable for more time steps. Also notable is how we see an increasing value for κ at high dimensional spaces. This is most likely due to the fact, that at higher dimensions the classic algorithm requires more time steps, in order to find a suitable estimation, thus having a larger confidence set bound. In these instances a trained source bandit would be viable early on. The regret is reduced for lower values of U and the time κ at which a source is discarded is extended. For source bandits satisfying $\|\boldsymbol{\theta}_S - \boldsymbol{\theta}^*\|_2 = 2$, we would retrieve the classic regret bound, preventing negative transfer.
Next we show what happens in case of a negative transfer for the softmax update strategy, *i.e.* the source does not provide any useful information at all and worsens the regret rate with $\gamma_S > \gamma_T$ at all time steps.

Theorem 2. *Let $\{\mathbf{x}_{a_k}\}_{k=1}^N$ be sequence in $\mathbb{R}^d$ and the minimal difference between confidence set bounds given as $\Delta_{\min} = \min_{k\in\{0,\dots,N\}}(\gamma_S(k) - \gamma_T(k))$, with $\gamma_S > \gamma_T$ for all time steps and the initial target weight denoted by $\alpha_T(0)$. Then with probability of at least $1-\delta$ an upper regret bound $R(n)$ in case of a negative transfer scenario is given by:*

$$R(n) \leq \frac{(1-\alpha_T(0))}{e\beta\alpha_T(0)(1-\exp(-\beta\Delta_{\min}))} + R_T(n) \tag{13}$$

Theorem 2 shows that in case of a negative transfer, the upper regret bound is increased by at most a constant term and vanishes in the case of $\beta \to \infty$ retrieving the hard update rule.

5 Weighted Linear Bandits with Multiple Sources

Up until now we only used a single source bandit, but our model can easily be extended to an arbitrary amount of different sources. Assuming we have M source bandits $\{\theta_{S,j}\}_{j=1}^M$, we define $\hat{\theta}$ as:

$$\hat{\boldsymbol{\theta}} = \sum_{j=1}^{M} \alpha_{S,j}\boldsymbol{\theta}_{S,j} + \alpha_T\hat{\boldsymbol{\theta}}_T, \tag{14}$$

with $\alpha_{S,j}, \alpha_T \geq 0\ \forall 1 \leq j \leq M$ and $\alpha_T + \sum_{j=1}^M \alpha_{S,j} = 1$. With this each source bandit yields its own confidence set bound $\gamma_{S,j}$. Similarly to (5) we retrieve for the UCB with multiple sources:

$$\text{UCB}(a) = \mathbf{x}_a^T\left(\sum_{j=1}^{M}\alpha_{S,j}(k)\boldsymbol{\theta}_{S,j} + \alpha_T(k)\hat{\boldsymbol{\theta}}_T(k)\right) + \boldsymbol{\alpha}^T(k)\boldsymbol{\gamma}\|\mathbf{x}_a\|_{\mathbf{A}^{-1}(k)}, \tag{15}$$

with $\boldsymbol{\alpha}(k) = (\alpha_{S,1}(k), ..., \alpha_{S,M}(k), \alpha_T(k))^T$ and $\boldsymbol{\gamma} = (\gamma_{S,1}, ..., \gamma_{S,M}, \gamma_T)^T$. As for the weight updates the same single source strategies apply *i.e.* the minimization of the exploration term in the UCB function:

$$\boldsymbol{\alpha}(k+1) = \underset{\boldsymbol{\alpha} \in \mathfrak{P}_{M+1}}{\arg\min}\, \boldsymbol{\alpha}^T(k)\boldsymbol{\gamma} + \frac{1}{\beta}\mathrm{KL}(\boldsymbol{\alpha}||\boldsymbol{\alpha}(k)), \tag{16}$$

where $\mathfrak{P}_{M+1}$ is the $(M+1)-$dimensional probability simplex. The solution of the previous problem is:

$$\alpha_{S,m}(k+1) = \frac{\alpha_{S,m}(k)\exp\left(-\beta\gamma_{S,m}\right)}{\sum_{j=1}^{M}\alpha_{S,j}(k)\exp(-\beta\gamma_{S,j}) + \alpha_T(k)\exp(-\beta\gamma_T)}. \tag{17}$$

This is basically the solution of (10) generalized to multiple sources. In the decisions making it favours the bandit with the lowest upper bound γ of their confidence set. When we take the limit $\beta \to \infty$ in (16) the KL-divergence term vanishes and we retrieve the hard case:

$$\alpha_{S,j} = \mathbb{1}_{\gamma_{S,j} = \min(\min_i \gamma_{S,i}, \gamma_T)} \tag{18}$$

which forces the weights to satisfy $\alpha_{S,m}, \alpha_T \in \{0,1\}$ for every source index and for all time steps. Thus decision making is done by selecting one single bandit in each round with the lowest value of their respective confidence set bound γ. The regret of hard update strategy for multiple sources is given by the following theorem:

Theorem 3. *Let* $\{\mathbf{x}_{a_k}\}_{k=1}^N$ *be sequence in* $\mathbb{R}^d$ *and* $\min_m \|\boldsymbol{\theta}_{S,m} - \boldsymbol{\theta}^*\| = U_{\min}$ *and the classic regret bound of the linear model up to time step n given by* $R_T(n)$ *[1]. Let* $m \coloneqq \min(\kappa, n)$ *and* $\delta \leq \exp(-2\lambda)$*. Then with probability of at least* $1-\delta$ *the regret of the hard update approach for the weighted LinUCB algorithm with multiple sources is bounded by:*

$$R(n) \leq 4U_{\min}\sqrt{\kappa d \log(1+\kappa/(d\lambda))(\lambda+\kappa)} - R_T(m) + R_T(n) \leq R_T(n), \tag{19}$$

with κ *as:*

$$\kappa = \left\lfloor 2\left[d\left(\frac{1}{U_{\min}^2} - \lambda\right) + \lambda\left(\frac{2}{U_{\min}^2} - \frac{1}{2}\right)\right]\right\rfloor.$$

depending on $U_{\min}$ the multiple source approach benefits from the additional information as the upper bound corresponds to the best source overall. In case of the softmax update strategy, we need to show how the regret changes in case of a negative transfer scenario, *i.e.* the confidence set bounds of any source bandit is larger than the target bound at any time.

Theorem 4. *Let $\{\mathbf{x}_{a_k}\}_{k=1}^N$ be sequence in $\mathbb{R}^d$, a total of M source bandits being available indexed by j and the minimal difference between confidence set bounds set as $\Delta_{\min,j} = \min_{k\in\{0,\dots,N\}}(\gamma_{S,j}(k) - \gamma_T(k))$ for every source j with $\gamma_{S,j} > \gamma_T$ $\forall j$ at every time step. Additionally the initial target weight is denoted by $\alpha_T(0)$. Then with probability $1-\delta$ an upper regret bound $R(n)$ in case of a negative transfer scenario is given by:*

$$R(n) \leq \frac{(1-\alpha_T(0))}{e\beta M \alpha_T(0)} \sum_{j=1}^{M} \frac{1}{(1-\exp(-\beta\Delta_{\min,j}))} + R_T \tag{20}$$

In comparison to the single source result, the additional constant is averaged over all sources. Depending on the quality, it can be beneficial to include more source bandits as potentially bad sources would be mitigated.

Algorithm 1: Weighted LinUCB

Initialize: $\hat{\boldsymbol{\theta}}_T(0)$ from $\mathcal{U}([0,1]^d)$, $\alpha_{S,j}(0) = (1-\alpha_T(0))/M = \frac{1}{2M}$, $U_j > 0$ $\gamma_{S,j} > 0$ $\forall j \in \{1,...,M\}$, $\delta \in [0,1]$, $\gamma_T > 0$, $\lambda > 0$, $\beta > 0$, $\mathbf{A}(0) = \lambda\mathbf{I}$, $\mathbf{b}(0) = \mathbf{0}$;
for $k = 0...N$ **do**
- Pull arm $a_k = \arg\max_a \text{UCB}(a)$ taken from (15);
- Receive estimated rewards from sources and real rewards: $r_{S,j}(k)|_{j\in\{0,\dots,M\}}, r(k)$;
- $\mathbf{A}(k+1) = \mathbf{A}(k) + \mathbf{x}_{a_k}\mathbf{x}_{a_k}^T$;
- $\mathbf{b}(k+1) = \mathbf{b}(k) + r(k)\mathbf{x}_{a_k}$;
- $\hat{\boldsymbol{\theta}}_T(k+1) = \mathbf{A}^{-1}(k+1)\mathbf{b}(k+1)$;
- Store rewards $r_{S,j}(k)|_{j\in\{0,\dots,M\}}, r(k)$ in vectors $\mathbf{y}_{S,j}(k)|_{j\in\{0,\dots,M\}}, \mathbf{y}(k)$ respectively;
- Calculate $\overline{\mathbf{y}}(k)$ from $\mathbf{y}(k)$ such that each entry r corresponding to the latest arm a_k pulled is updated to the mean reward $\overline{r}$ of the respective arm;
- Update $U_j = \max_{i\in\{0,\dots,k\}} \frac{|\overline{r}(i) - r_{S,j}(i)|}{\|\mathbf{x}_{a_i}\|}$ for every j;
- $\gamma_{S,j} = \sqrt{\lambda U_j + \|\mathbf{y}_{S,j}(k) - \overline{\mathbf{y}}(k)\|}$;
- $\gamma_T = \sqrt{\lambda} + \sqrt{\log \frac{\|\mathbf{A}(k)\|}{\lambda^d \delta^2}}$;
- update source weights $\alpha_{S,j}(k+1)$ according either to softmax rule in (17):
- or to the hard update rule in (18);
- update target weight as:
- $\alpha_T(k+1) = 1 - \sum_{j=1}^{M} \alpha_{S,j}(k+1)$;

For the practical implementation we use $\gamma_T = \sqrt{\lambda} + \sqrt{\log \frac{\|\mathbf{A}(k)\|}{\lambda^d \delta^2}}$ which is also taken from [1] and gives a tighter confidence set bound on the target estimator. Also we give an estimation for U_j by taking the maximum value of the lower bound induced by the Cauchy-Schwartz inequality $U_j = \|\boldsymbol{\theta}_{S,j} - \boldsymbol{\theta}^*\| \geq \max_{i\in\{0,\dots,k\}} \frac{|\overline{r}(i) - r_{S,j}(i)|}{\|\mathbf{x}_{a_i}\|}$ at each time step.

5.1 Biased Regularization

In [32] a similar approach of model reuse was used in a concept drift scenario for linear classifiers via biased regularization. In [12] the risk generalization analysis for this approach was delivered in a supervised offline learning setting. Their mathematical formulation is stated as following: A classifier is about to be trained given a target training set $(\mathbf{D}, \mathbf{y})$ and a source hypothesis $\boldsymbol{\theta}_{src}$, which is specifically used for a biased regularization term. In contrast to our approach the weighting is only applied the source model, giving an alternate solution to the target classifier. Adapted to a linear bandit model, the optimization problem can be formulated as:

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \|\mathbf{D}\boldsymbol{\theta} - \mathbf{y}\|^2 + \lambda\|\boldsymbol{\theta} - \boldsymbol{\theta}_{src}\|^2. \tag{21}$$

$\boldsymbol{\theta}_{src}$ is a convex combination of an arbitrary amount of given source models $\{\boldsymbol{\theta}_j\}_{j\in\{1,\dots,M\}}$:

$$\boldsymbol{\theta}_{src} = \sum_{j=1}^{M} \alpha_j \boldsymbol{\theta}_j, \tag{22}$$

As in our model, these weights are not static and are updated after each time step. The update strategy is not chosen to minimize the upper regret bound but can be chosen such that the convex combination is as close as possible to the optimal bandit parameter. The UCB function is then simply given by:

$$\mathrm{UCB}(a) = \mathbf{x}_a^T \hat{\boldsymbol{\theta}} + \gamma \|\mathbf{x}_a\|_{\mathbf{A}^{-1}(k)}, \tag{23}$$

with $\gamma = \sqrt{d\log\big(1 + \frac{k}{d\lambda}\big) + \log\big(\frac{1}{\delta^2}\big)} + \sqrt{\lambda}\|\boldsymbol{\theta}_{src} - \boldsymbol{\theta}^*\|_2$ and the solution to (21):

$$\hat{\boldsymbol{\theta}} = \mathbf{A}^{-1}\mathbf{D}^T\mathbf{y} - (\mathbf{A}^{-1}\mathbf{D}^T\mathbf{D} - \mathbf{I})\boldsymbol{\theta}_{src}. \tag{24}$$

At some point in time we expect the weights to converge to a single source bandit closest to the optimal bandit. But contrary to our original model it is not possible for the model to discard all sources once the target estimation yield better upper bounds for their confidence sets. The upper regret bound is similar to the classic bound with the difference being in one term.

Theorem 5. *Let $\{\mathbf{x}_{a_k}\}_{k=1}^N$ be sequence in $\mathbb{R}^d$ and the upper bound of the biggest euclidean distance between any of the M source bandit indexed by m and optimal bandit parameter given by $\max_m \|\boldsymbol{\theta}_{S,m} - \boldsymbol{\theta}^*\| \leq U_{\max}$, then with probability of at least $1-\delta$ the regret of the biased LinUCB algorithm with multiple sources is upper bounded by:*

$$R(n) \leq \sqrt{8nd\log(\lambda + n/d)}\left(\sqrt{d\log\Big(1 + \frac{n}{d\lambda}\Big) + \log\Big(\frac{1}{\delta^2}\Big)} + \sqrt{\lambda}U_{\max}\right) \tag{25}$$

Since we are looking for an upper bound, U is dominated by the largest euclidean distance between the optimal bandit parameter and all given source bandits. Theorem 5 differs from the classic case in the regularization related parameters where we have $\sqrt{\lambda}U_{\max}$ instead of $\sqrt{\lambda}\|\boldsymbol{\theta}^*\|$. For sources with low values of U, we improve the overall regret.

6 Experimental Results

We test the presented algorithms, *i.e.* the weighted model algorithm as well as the biased regularization algorithm, for single source and multiple source transfers on synthetic and real data sets. The plots include the results from the classical LinUCB approach as well as the EXP4 approach from [15] with target and source models acting as expert, for comparison purposes. Additionally to the regret plots we also showcase the mean of the target weight as a function of time to see how the relevancy of the target estimation evolved.

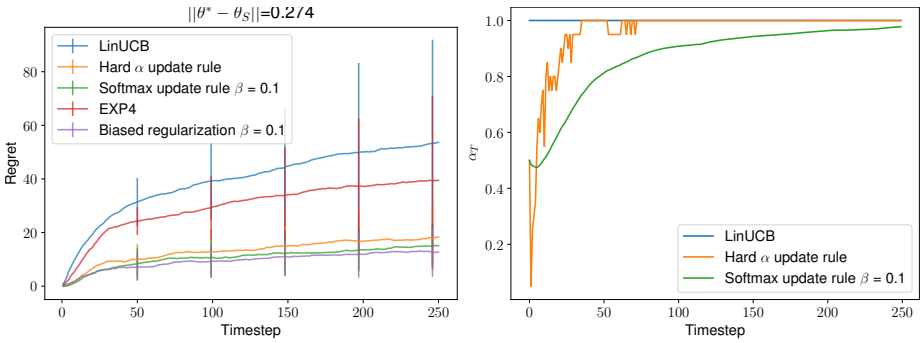
6.1 Synthetic Data Experiments

Our synthetic experiments follow a similar approach to [18]. The target context feature vectors $\mathbf{x}_a$ are drawn from a multivariate Gaussian with variances sampled from a uniform distribution. We chose the number of dimensions $d = 20$ and the number of arms to be 1000. Our optimal target bandit parameter is sampled from a uniform distribution and scaled such that $\|\boldsymbol{\theta}^*\| \leq 1$, thus the rewards are implicitly initialized as well with $r = \mathbf{x}_a^T\boldsymbol{\theta}^* + \epsilon$, with some Gaussian noise $\epsilon \sim \mathcal{N}(0, \sigma^2)$ and $\sigma = 1/\sqrt{2\pi}$. The source bandit parameters $\boldsymbol{\theta}_{S,m}$ are initialized by adding a random noise vector $\boldsymbol{\eta}_m$ to the optimal target bandit parameters for every source bandit to be generated $\boldsymbol{\theta}_{S,m} = \boldsymbol{\theta}^* + \boldsymbol{\eta}_m$. This way we ensure that there is actual information of the target domain in the source bandit parameter. We could also scale $\boldsymbol{\eta}_m$ to determine how much information the respective source yields about the target domain. The regularization parameter was constantly chosen to be $\lambda = 1$ and the initial weights are equally distributed among all available bandit parameters: $\alpha_T = \alpha_{S,m} = \frac{1}{M+1}$. The shown results are the averaged values over 20 runs.

As we showed in Section 4 the upper regret bound is lower for $\beta \to \infty$ *i.e.* the hard update rule which ignores the KL-divergence in the optimization, but we see overall better results than in the classic case with the softmax update strategy as well. The inclusion of eight more source bandits in Figure 2 improves the sources slightly, though it should be mentioned that all sources generated were similar in quality. Thus we would expect higher improvements in the regret when including significantly better sources. The EXP4 algorithm on the other hand does not perform as well when increasing the number of experts.

6.2 Real Data Experiments

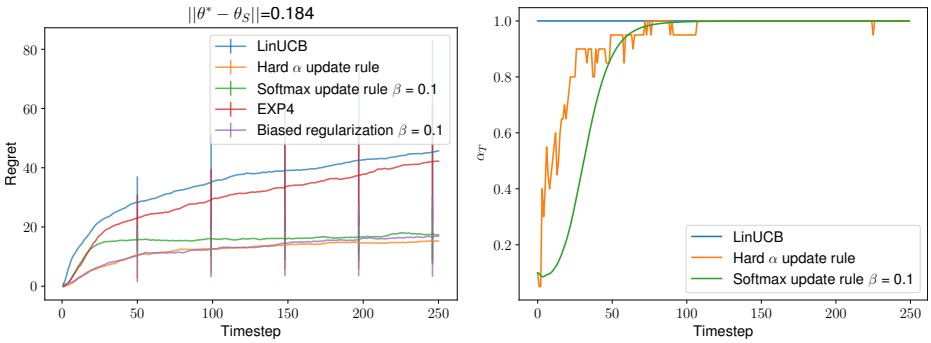
The real data sets used for our purposes are taken from the MovieLens sets. Their data include an assemble of thousands of users and corresponding traits



(a) Regret evolution plot labeled by confidence set bound.

(b) Evolution of the target weight α_T.

Fig. 1: Regret and weight evolution for single source transfer scenario on synthetic data sets. The blue lines showcase the classic LinUCB results. The vertical lines indicate the standard deviation.



(a) Regret evolution plot labeled by the lowest confidence set bound of all available sources.

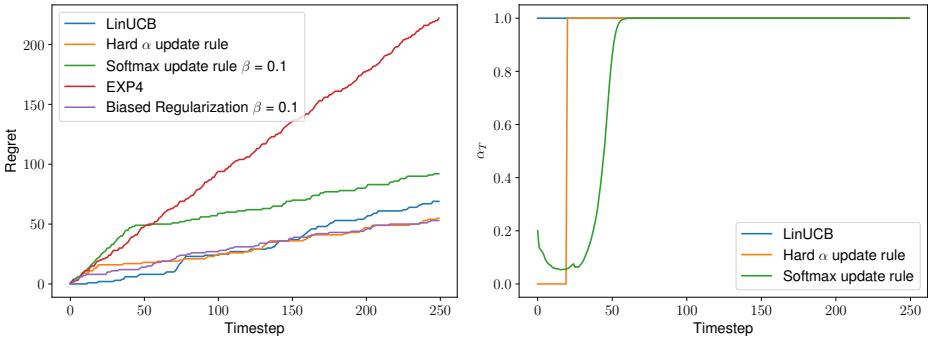
(b) Evolution of the target weight α_T. Since multiple sources are present, the initial weight is reduced

Fig. 2: Regret and weight evolution for multiple source transfer scenario (9 sources) on synthetic data sets. The blue lines showcase the classic LinUCB results. The vertical lines indicate the standard deviation.

such as age, gender and profession as well as thousands of movies and their genres. Every user has a rating from 1 to 5 given to at least 20 different movies. The movies, rated by a user, function as the available arms for that particular user. The information of the movies apart from the title itself are solely given by their genres. Each movie may have up to three different genres and there are 18 different genres in total. Arms, which are linked to the movies, have context vectors depending on the movies genre only. We design 18-dimensional context vector with each dimension representing a genre. If the movie is associated with a particular genre, the respective dimensional feature is set as $x_i = \frac{1}{\sqrt{S}}$ with S as the total number of genres the movie is associated with. This way we guarantee that every context vector is bounded by 1. the reward of an arm in our bandit setting is simply given by the user rating.
For our purposes we require source bandits for the transfer learning to take place. Therefore we pretrained a bandit for every single user, given all of the movie information, with the classic LinUCB algorithm and stored the respective parameters. This way every single user can function as a potential source for a different user. With all of the users available we grouped them according their age, gender and profession. We enforce every user to only act as source to other users with similar traits. This stems from a general assumption that people with matching traits may also have similar interests. This is a very general assumption made but given all of the information, it is the easiest way to find likely useful sources for every user. In Figure 3 the results for two individuals of two different groups of users respectively are showcased. Instead of only using one source, we used the multiple source strategy and made use of every user of the same group the individuals are located in, since this way we have a higher chance to find good sources. Even though the real data is far from guaranteed to have a linear reward structure, as well as the fact that important information on the arms' contexts are not available, since ratings usually not only depend on the movie genre, we find satisfying results with converging regrets as well as improved learning rates when including sources.

7 Discussion and Outlook

This work shows that our approach to make use of information from different tasks, without having actually access to concrete data points, is efficient, given the improved regrets. We have proven an upper regret bound of our weighted LinUCB algorithm with the hard update strategy at least as good as the classic LinUCB bound with a regret rate of $O(d\sqrt{n \log n})$, and a converging sublinear negative-transfer term when using the softmax update strategy. Further argument for the utility of our model was given with synthetic and real data experiments. The synthetic data sets showed promising results especially with the softmax update strategy, even without having a guaranteed improved regret bound. The softmax approach uses a convex combination of models, which might be more practical than using one model at a time especially when it comes to high quality sources. This further raises the question whether different weight-



(a) Regret evolution plot with user data taken from the group of 35 to 44 years old female lawyers.

(b) Target weight evolution plot with the respective algorithms labeled with user data taken from the group of 35 to 44 years old female lawyer.

Fig. 3: Regret evolution for multiple source transfer scenario on real data sets taken from Movielens data. A group of users are shown with one bandit trained for a random user of each group, while the rest of the users act as source to the respective user. The blue lines showcase the classic LinUCB results.

ing update rules, which yield solutions consisting of a span of source models, might be more efficient for transfer. The inclusion of multiple sources further improved the results, indicating that using information from multiple different tasks is more effective then just one, which aligns with our theoretical result in Theorem 3. The real-world data experiments showed improvements as well, even when considering that the rewards did not necessarily follow a linear model and that the available features for the context vector were rather sparse, the transfer of information from similar users almost always led to lower regrets.

In upcoming projects we intend to adapt our approach to non-linear models such as kernelized bandits, since the convex weighting is not limited to just linear models, as well as give a proper regret bound for the softmax update strategy. There is potential in using our transfer model to non stationary bandits, such that each prior estimation of the bandit parameter may act as source for the current setting, thus making use of the information collected in prior instances of the bandit setting. In this case we would need to make assumptions of the change rate of the tasks after a certain amount of time steps. Previous algorithms on non-stationary bandits [23] perform weighting on data points and discard them after some time steps, without evaluating the benefit of the data beforehand. In our setting, previously trained bandit parameters would be used according to their performance.

Acknowledgements This work was supported by Grant 01IS20051 from the German Federal Ministry of Education and Research (BMBF). S. Maghsudi is a member of the Machine Learning Cluster of Excellence, EXC number 2064/1

– Project number 390727645. The authors thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for supporting Steven Bilaj.

References

1. Abbasi-Yadkori, Y., Pál, D. & Szepesvári, C. Improved Algorithms for Linear Stochastic Bandits. *Advances In Neural Information Processing Systems*. (2011)
2. Amrallah, A., Mohamed, E., Tran, G. & Sakaguchi, K. Radio Resource Management Aided Multi-Armed Bandits for Disaster Surveillance System. *Proc. 2020 International Conference On Emerging Technologies For Communications (ICETC2020), Virtual, K1-4*. (2020)
3. Audibert, J., Munos, R. & Szepesvári, C. Exploration–exploitation tradeoff using variance estimates in multi-armed bandits. *Theoretical Computer Science*. (2009)
4. Auer, P., Cesa-Bianchi, N., Freund, Y. & Schapire, R. The nonstochastic multi-armed bandit problem. *SIAM Journal On Computing*. (2002)
5. Azar, M., Lazaric, A. & Brunskill, E. Sequential transfer in multi-armed bandit with finite set of models. *Advances In Neural Information Processing Systems*. (2013)
6. Bouneffouf, D., Rish, I. & Aggarwal, C. Survey on applications of multi-armed and contextual bandits. *2020 IEEE Congress On Evolutionary Computation (CEC)*. (2020)
7. Bush, R. & Mosteller, F. A stochastic model with applications to learning. *The Annals Of Mathematical Statistics*. (1953)
8. Chu, W., Li, L., Reyzin, L. & Schapire, R. Contextual Bandits with Linear Payoff Functions. *AISTATS*. (2011)
9. Du, S., Koushik, J., Singh, A. & Póczos, B. Hypothesis transfer learning via transformation functions. *Advances In Neural Information Processing Systems*. (2017)
10. Duan, L., Tsang, I., Xu, D. & Chua, T. Domain adaptation from multiple sources via auxiliary classifiers. *Proceedings Of The 26th Annual International Conference On Machine Learning*. (2009)
11. Kuzborskij, I. & Orabona, F. Stability and hypothesis transfer learning. *International Conference On Machine Learning*. (2013)
12. Kuzborskij, I. & Orabona, F. Fast rates by transferring from auxiliary hypotheses. *Machine Learning*. (2017)
13. Labille, K., Huang, W. & Wu, X. Transferable Contextual Bandits with Prior Observations. *Pacific-Asia Conference On Knowledge Discovery And Data Mining*. (2021)
14. Langford, J. & Zhang, T. The epoch-greedy algorithm for multi-armed bandits with side information. *Advances In Neural Information Processing Systems*. (2007)
15. Lattimore, T. & Szepesvári, C. Bandit Algorithms. (2020)
16. Li, L., Chu, W., Langford, J. & Schapire, R. A contextual-bandit approach to personalized news article recommendation. *Proceedings Of The 19th International Conference On World Wide Web*. (2010)
17. Liau, D., Song, Z., Price, E. & Yang, G. Stochastic multi-armed bandits in constant space. *International Conference On Artificial Intelligence And Statistics*. (2018)
18. Liu, B., Wei, Y., Zhang, Y., Yan, Z. & Yang, Q. Transferable contextual bandit for cross-domain recommendation. *Proceedings Of The AAAI Conference On Artificial Intelligence*. (2018)

19. Maiti, A., Patil, V. & Khan, A. Multi-Armed Bandits with Bounded Arm-Memory: Near-Optimal Guarantees for Best-Arm Identification and Regret Minimization. *Advances In Neural Information Processing Systems*. **34** (2021)
20. Perrot, M. & Habrard, A. A theoretical analysis of metric hypothesis transfer learning. *International Conference On Machine Learning*. (2015)
21. Ras, Z., Wieczorkowska, A. & Tsumoto, S. Recommender Systems for Medicine and Music. (2021)
22. Robbins, H. Some aspects of the sequential design of experiments. *Bulletin Of The American Mathematical Society*. (1952)
23. Russac, Y., Vernade, C. & Cappé, O. Weighted linear bandits for non-stationary environments. *Advances In Neural Information Processing Systems*. (2019)
24. Soare, M., Alsharif, O., Lazaric, A. & Pineau, J. Multi-task linear bandits. *NIPS2014 Workshop On Transfer And Multi-task Learning: Theory Meets Practice*. (2014)
25. Stark, B., Knahl, C., Aydin, M. & Elish, K. A literature review on medicine recommender systems. *International Journal Of Advanced Computer Science And Applications*. (2019)
26. Suk, J. & Kpotufe, S. Self-Tuning Bandits over Unknown Covariate-Shifts. *Algorithmic Learning Theory*. (2021)
27. Thompson, W. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*. (1933)
28. Tommasi, T., Orabona, F. & Caputo, B. Learning Categories From Few Examples With Multi Model Knowledge Transfer. *IEEE Transactions On Pattern Analysis And Machine Intelligence*. (2014)
29. Xu, X. & Zhao, Q. Memory-Constrained No-Regret Learning in Adversarial Multi-Armed Bandits. *IEEE Transactions On Signal Processing*. (2021)
30. Yang, J., Yan, R. & Hauptmann, A. Cross-domain video concept detection using adaptive svms. *Proceedings Of The 15th ACM International Conference On Multimedia*. (2007)
31. Zhao, P., Hoi, S., Wang, J. & Li, B. Online Transfer Learning. *Artificial Intelligence*. (2014)
32. Zhao, P., Cai, L. & Zhou, Z. Handling concept drift via model reuse. *Machine Learning*. (2020)
33. Zhou, Q., Zhang, X., Xu, J. & Liang, B. Large-scale bandit approaches for recommender systems. *International Conference On Neural Information Processing*. (2017)