Transferable Contextual Bandit for Cross-Domain Recommendation

Bo Liu[†], Ying Wei[†], Yu Zhang[†], Zhixian Yan[‡], Qiang Yang[†]
[†]the Hong Kong University of Science and Technology, Hong Kong
[‡]Cheetah Mobile USA
bliuab@cse.ust.hk, yweiad@cse.ust.hk, yuzhangcse@ust.hk
zhixian.yan@cmcm.com, qyang@cse.ust.hk

Abstract

Traditional recommendation systems (RecSys) suffer from two problems: the exploitation-exploration dilemma and the cold-start problem. One solution to solving the exploitation-exploration dilemma is the contextual bandit policy, which adaptively exploits and explores user interests. As a result, the contextual bandit policy achieves increased rewards in the long run. The contextual bandit policy, however, may cause the system to explore more than needed in the cold-start situations, which can lead to worse short-term rewards. Cross-domain RecSys methods adopt transfer learning to leverage prior knowledge in a source RecSys domain to jump start the cold-start target RecSys. To solve the two problems together, in this paper, we propose the first applicable transferable contextual bandit (TCB) policy for the cross-domain recommendation. TCB not only benefits the exploitation but also accelerates the exploration in the target RecSys. TCB's exploration, in turn, helps to learn how to transfer between different domains. TCB is a general algorithm for both homogeneous and heterogeneous domains. We perform both theoretical regret analysis and empirical experiments. The empirical results show that TCB outperforms the state-of-the-art algorithms over time.

Introduction

A personalized recommendation system (RecSys) is a vital component for online interactive services. According to user interests, an online service provider recommends items including movies, apps, articles, to achieve more user feedbacks including watches, installations, and clicks accordingly. Accurately inferring user interests and providing corresponding recommendations not only improve user satisfaction but also increase commercial revenue significantly. The exploitation-exploration dilemma and the cold-start problem are two major obstacles to the successful deployment of a RecSys (Schein et al. 2002; Li et al. 2010).

The primary objective of a RecSys is to maximize the cumulative amount of feedback which depends on user interests. The feedback, however, has high uncertainty and noise. A successful RecSys is supposed to balance between exploiting predicted user interests upon historical feedbacks and exploring uncertain user interests via recommending diverse items. Compromising between the two conflicting objectives is well known as the exploitation-exploration dilemma in RecSys.

Recently, the contextual bandit has been introduced to formulate the exploitation-exploration dilemma and achieved great success (Li et al. 2010). In a contextual bandit problem, the agent observes K arms $a_1, \cdots, a_K$ (e.g., K candidate articles) and their context $\mathbf{x}_{a_1}, \ldots, \mathbf{x}_{a_K}$ (e.g., user profile, article content, etc.). In step τ, the agent decides to pull an arm a_τ among all arms (e.g., recommending one article) and observes the corresponding reward r_{a_τ} (e.g., click or not). The reward depends on the context but is noisy, i.e., $r_{a_\tau} = f(\mathbf{x}_{a_\tau}) + \epsilon$. According to N historical observations $\{(\mathbf{x}_{a_\tau}, r_{a_\tau})\}_{\tau=1\ldots N}$, a contextual bandit policy adaptatively decides which arm to pull in each step and learns the uncertain reward function. The process is illustrated in the left of Fig. 1. To maximize the cumulative reward, the agent should pull the arms with the larger predicted reward. Because only the stochastic reward of the pulled arms are observed, the agent is also expected to pull the arms which accelerate the exploration of the uncertain reward function (Zhou 2015). Therefore, the contextual bandit policy can well balance the exploitation and exploration trade-off, and maximize the cumulative reward in the long run (Chu et al. 2011).

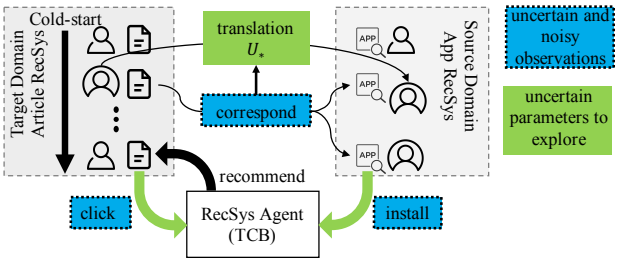


Figure 1: The process of cross-domain RecSys using TCB. In terms of contextual bandit, TCB sequentially and adaptatively recommends articles based on noisy feedbacks. In terms of transfer learning, TCB adopts the translation to leverage both source and target observations. TCB suffers from noisy observations in blue and explores uncertain parameters in green accordingly.

Unfortunately, the contextual bandit policies suffer from the cold-start problem which refers to that there exists few observations for new users, new items, or newly launched domains. In cold-start conditions, the agent tends to focus on

exploration and sacrifice short-term reward for long-term reward. The ϵ-first policy (Tran-Thanh et al. 2010), for instance, purely explores in cold-start conditions via random recommendations, which goes overboard with the exploitation and exploration balancing and results in even worse performance.

The cross-domain recommendation has been widely acknowledged as a killer method to solve the cold-start problem (Pan et al. 2010). A cross-domain RecSys leverages the prior knowledge learned from an auxiliary domain with abundant observations to improve the performance in the target domain of interest. The prior knowledge to be transferred could be either user interests or item properties. As illustrated in Fig. 1, a company is about to launch a new article RecSys, i.e., the target domain. Meantime, the company has accumulated sufficient application RecSys data. Assuming that a user's interests on applications also apply to articles, a cross-domain RecSys transfers user interests on applications as the source domain to provide better article recommendation.

In this paper, we are motivated to address the exploitation-exploration dilemma and the cold-start problem together via a new **T**ransferable **C**ontextual **B**andit (**TCB**) algorithm. The TCB harnesses the collective and mutually reinforcing power of contextual bandit and transfer learning. First, transfer learning improves the exploitation of a contextual bandit policy and accelerates its exploration in a target domain. Assuming that the user interests in applications and articles are closely related, the TCB can infer user interests based on both the source and target observations. Thus, the TCB can estimate user interests for exploitation far better than single-domain algorithms, and significantly reduce the uncertainty of the estimated reward function for exploration. Second, the contextual bandit speeds up the knowledge transfer. As shown in Fig. 1, the TCB transfers knowledge via the translation $\mathbf{U}_*$. The TCB explores not only the reward function but also how to transfer, i.e., the translation (Pan and Yang 2010). Thus the TCB can recommend those articles that help learn how to transfer the fastest. Provided that the TCB progressively and adaptively recommends an article in the target domain, it relies on the currently available source and target observations as well as the correspondence data between them. The correspondence data indicates the similarity between a source and a target observation. For example, the observations across domains produced by the same user enjoy a large similarity. Especially, the TCB is designed to handle knowledge transfer between both heterogeneous domains with the source and target contexts lying in different feature spaces and homogeneous domains in the same feature space.

The primary contributions of this paper are threefold: 1) to the best of our knowledge, TCB is the first applicable transferable contextual bandit policy for cross-domain RecSys. TCB is empirically verified using the real-world RecSys data. 2) the proposed transfer bandit policy is general for both homogeneous and heterogeneous domains; 3) the theoretical regret analysis is also provided to guard the proposed algorithm.

Related Work

In this section, we discuss the existing contextual bandit approaches and transfer learning algorithms related to TCB.

LinUCB (Li et al. 2010) firstly formulates the RecSys as a contextual bandit problem. LinUCB not only improves the reward (Li et al. 2011) in real-world online RecSys but also enjoys a sub-linear regret guarantee (Chu et al. 2011). LinUCB assumes that the expected reward is linear with respect to context. Under the same assumption, Agrawal and Goyal proposed a Thompson Sampling style method (Agrawal and Goyal 2013). Both linear policies, however, are sensitive to the dimension and quality of context. When the context is high-dimensional, Yue, Hong, and Guestrin proposed CoFineUCB to accelerate the exploration in LinUCB by introducing two levels of feature spaces, i.e., the fine and coarse feature spaces (Yue, Hong, and Guestrin 2012). To tackle not fully observable context, hLinUCB (Wang, Wu, and Wang 2016), with a significantly better regret guarantee, explicitly explores the hidden context and reward function simultaneously.

Collaborative filtering, one of the most important RecSys techniques, is also combined with contextual bandit models. COFIBA (Li, Karatzoglou, and Gentile 2016), for example, explores a particular user's interests more efficiently by leveraging the observations from all users within the same cluster. FactorUCB (Cesa-Bianchi, Gentile, and Zappella 2013; Wang, Wu, and Wang 2017) assumes the reward function as the dot product between user latent factors and item latent factors. Upon the known user relationship, FactorUCB explores both latent factors in a collaborative filtering manner. All the aforementioned contextual bandit policies, unfortunately, focus on the RecSys within a single domain and thereby tend to fail in the domains with insufficient observations.

Another line of related works is transfer learning (Pan and Yang 2010) which aims to improve the learning performance of a target domain by transferring knowledge from a source domain with sufficient observations. Heterogeneous transfer learning (HTL) advances by transferring between domains in different feature spaces. To bridge incommensurable domains, existing works fall into two categories: TLRisk (Dai et al. 2008) learns a translator from the co-occurrence data to translate features between domains; HTLIC (Zhu et al. 2011), HeMap (Shi et al. 2013), and CoHTL (Wei et al. 2016) all learn a common latent space where the two domains are comparable. Additionally, cross-domain algorithms tailor the transfer learning for RecSys. CST (Pan et al. 2010) and TCF (Pan et al. 2011), for instance, learn user latent factors and item latent factors using matrix tri-factorization. CST and TCF transfer the knowledge via shared user interests, i.e., shared user latent factors. All the transfer learning algorithms mentioned above, however, work under the supervised learning setting. Without exploration, the HTL algorithms suffer from a linear regret when facing a contextual bandit problem.

Though tUCB (Azar, Lazaric, and Brunskill 2013) and B-kl-UCB (Zhang and Bareinboim 2017) consider transfer learning for the bandit problem by transferring the estimated upper confidence bound, they differ from ours greatly. tUCB considers context-free multi-armed bandit problems which are inapplicable to real-world RecSys. Moreover, both methods require the action sets to be exactly the same across domains. In comparison, TCB works even if two domains are heterogeneous, which is frequent for cross-domain RecSys.

Methods

In this section, we first define the notations used throughout this paper. Then, in the view of transfer learning, we introduce how TCB exploits and explores, respectively. Finally, we theoretically analyze TCB's regret.

In this paper, the bold uppercase symbol ($\mathbf{A}$), bold lowercase symbol ($\mathbf{a}$), and regular symbol (x), denote the matrix, column vector, and scalar, respectively. The uppercase calligraphic symbol stands for the set, e.g., $\mathcal{A}$. We use $\mathbf{I}_d$ to denote the identity matrix with d rows and columns. Similarly, $\mathbf{0}_{d_1,d_2}$ denotes the zero matrix with d_1 rows and d_2 columns. $||\mathbf{A}||_F$ and $||\mathbf{a}||_2$ represent the Frobenius norm for a matrix and l_2 norm for a vector, respectively. For a positive definitive square matrix $\mathbf{A}$, $||\mathbf{a}||_{\mathbf{A}}$ represents the weighted l_2 norm, i.e., $||\mathbf{a}||_{\mathbf{A}} = \sqrt{\mathbf{a}^T\mathbf{A}\mathbf{a}}$. Furthermore, $\otimes$ denotes the Kronecker product, and vec$(\cdot)$ is the vectorization operator.

Exploitation in TCB

TCB assumes that N^s source observations are available, i.e., $\mathbf{O}^s_{N^s} = \{\mathbf{X}^s_{N^s}, \mathbf{r}^s_{N^s}\} = \{(\mathbf{x}^s_{a_\tau}, r^s_{a_\tau})\}_{\tau=1\dots N^s}$ where $\mathbf{X}^s_{N^s} \in \mathbb{R}^{N^s\times d^s}$. The target observations until step N^t are defined similarly as $\mathbf{O}^t_{N^t} = \{\mathbf{X}^t_{N^t}, \mathbf{r}^t_{N^t}\} = \{(\mathbf{x}^t_{a_\tau}, r^t_{a_\tau})\}_{\tau=1\dots N^t}$ where $\mathbf{X}^t_{N^t} \in \mathbb{R}^{N^t\times d^t}$. The objective of TCB is to progressively and adaptatively select the actions $\{a_1, \dots, a_{N^t}\}$ in the target domain to maximize the cumulative reward, i.e., $\sum_{\tau=1}^{N^t} r^t_{a_\tau}$. TCB learns the selection strategy from both source observations $\mathbf{O}^s_{N^s}$ and target observations $\mathbf{O}^t_{N^t}$. Without loss of generality, TCB focuses on the knowledge transfer between heterogeneous domains, which signifies that a source and a target domain have different actions or their contexts are in different feature spaces.

One of the essential steps to design a transfer learning algorithm is to decide how to transfer (Pan and Yang 2010). "How to transfer" for heterogeneous transfer learning mainly aligns incommensurable feature spaces so that the transferable knowledge can be discovered. TCB adopts a translation matrix to align different feature spaces. In comparison with transfer learning methods via a common space such as CoHTL, the translation based TCB suffers from fewer sources of noise and explores fewer uncertain parameters. The optimal translation matrix $\mathbf{U}_* \in \mathbb{R}^{d^t\times d^s}$ is expected to perfectly translate a target context $\mathbf{x}^t_a$ into the source feature space, i.e. $\mathbf{x}^s_a = \mathbf{U}_*^T\mathbf{x}^t_a$. Under the assumption that the reward function is linear with regard to the context, the rewards of both domains are determined by,

$$\begin{aligned} r^t_a &= (\mathbf{x}^t_a)^T\mathbf{U}_*\boldsymbol{\theta}^s_* + \epsilon_\tau, \\ r^s_a &= (\mathbf{x}^s_a)^T\boldsymbol{\theta}^s_* + \epsilon_\tau, \end{aligned} \tag{1}$$

where ϵ_τ is $1/\sqrt{2}$-sub-Gaussian noise and $\boldsymbol{\theta}^s_*$ is the reward parameter for the source domain.

By supposing that we are given the optimal translation matrix $\mathbf{U}_*$, we estimate the reward parameter $\boldsymbol{\theta}^s_*$ from all source observations and the target observations in step τ by solving the following optimization problem similar to LinUCB:

$$\hat{\boldsymbol{\theta}}^s_\tau = \arg\min_{\boldsymbol{\theta}^s} ||\mathbf{r}^s_{N_s} - \mathbf{X}^s_{N_s}\boldsymbol{\theta}^s||_2^2 + ||\mathbf{r}^t_\tau - \mathbf{X}^t_\tau\mathbf{U}_*\boldsymbol{\theta}^s||_2^2 + ||\boldsymbol{\theta}^s||_2^2. \tag{2}$$

When τ is extremely small, all the source observations serve as the warm start and definitively benefit the exploitation and exploration of LinUCB.

Unfortunately, the optimal translation matrix $\mathbf{U}_*$ is unknown in a real-world RecSys. $\mathbf{U}_*$ is also expected to learn from all the observations. Directly optimizing Eq. 2 w.r.t. $\mathbf{U}_*$ without any constraints, however, is equivalent to the ridge regression purely on the target observations, which yields a trivial translation matrix performing poorly on knowledge transfer. Inspired by (Dai et al. 2008; Wei et al. 2016), we learn a more effective $\mathbf{U}_*$ by also leveraging the correspondence data between source and target observations. We denote the correspondence data as $\mathbf{S} \in \mathbb{R}^{N_t\times N_s}$ with $\mathbf{S}_{ij}$ indicating the relatedness between the ith target observation and the jth source observation, e.g., whether they are produced by the same user. We would believe that such correspondence data are easily accessible nowadays. $\mathbf{S}_{i,j} > 0$ if the ith target observation and the jth source observation are related and $\mathbf{S}_{i,j} = 0$ otherwise. The optimal $\mathbf{U}_*$ is also expected to align each pair of observations across domains with $\mathbf{S}_{i,j} > 0$,

$$\mathbf{x}^s_j = \mathbf{U}_*^T\mathbf{x}^t_i + \boldsymbol{\eta}, \text{ if } \mathbf{S}_{i,j} > 0, \forall i, \forall j, \tag{3}$$

where $\boldsymbol{\eta}$ denotes the translation noise, with each element as a $c_\eta/\sqrt{2}$-sub-Gaussian noise. Consequently, based on Eq. 1 and Eq. 3, the overall loss function to simultaneously estimate the reward parameter $\boldsymbol{\theta}^s_*$ and the translation matrix $\mathbf{U}_*$ in each step τ is shown as below,

$$\begin{aligned} \hat{\mathbf{U}}_\tau, \hat{\boldsymbol{\theta}}^s_\tau = \arg\min_{\mathbf{U},\boldsymbol{\theta}^s} &\beta||\mathbf{r}^s_{N_s} - \mathbf{X}^s_{N_s}\boldsymbol{\theta}^s||_2^2 + ||\mathbf{r}^t_\tau - \mathbf{X}^t_\tau\mathbf{U}\boldsymbol{\theta}^s||_2^2 \\ &+ \gamma\sum_{i=1}^{\tau}\sum_{j=1}^{N_s}\mathbf{S}_{i,j}||\mathbf{U}^T\mathbf{x}^t_i - \mathbf{x}^s_j||_2^2 + ||\boldsymbol{\theta}^s||_2^2 + ||\mathbf{U}||_F^2, \end{aligned} \tag{4}$$

where β and γ control the importance of the source domain and the translation, respectively. The first term incorporates source observations to learn the reward parameter $\boldsymbol{\theta}^s_*$, and the third term learns the translation $\mathbf{U}_*$ by encouraging the correspondence between observations across domains to be preserved. The second term emphasizes the target observations and learns both parameters. The last two terms are regularizers to avoid overfitting. For simplicity, we also write the third term as,

$$\begin{aligned} &\sum_{i=1}^{\tau}\sum_{j=1}^{N_s}\mathbf{S}_{i,j}||\mathbf{U}^T\mathbf{x}^t_i - \mathbf{x}^s_j||_2^2 = Tr(\mathbf{U}^T(\mathbf{X}^t_\tau)^T\mathbf{S}_1\mathbf{X}^t_\tau\mathbf{U}) \\ &+ Tr((\mathbf{X}^s_{N_s})^T\mathbf{S}_2\mathbf{X}^s_{N_s}) - 2Tr(\mathbf{U}^T(\mathbf{X}^t_\tau)^T\mathbf{S}\mathbf{X}^s). \end{aligned}$$

where $\mathbf{S}_1$ and $\mathbf{S}_2$ are diagonal matrices with $\mathbf{S}_1(i,i) = \sum_{j=1}^{N_s}\mathbf{S}_{i,j}$ and $\mathbf{S}_2(j,j) = \sum_{i=1}^{\tau}\mathbf{S}_{i,j}$.

Obviously, the optimization problem in Eq. 4 is not jointly convex w.r.t. $\mathbf{U}$ and $\boldsymbol{\theta}^s$. However, it is convex w.r.t. one parameter if the other is fixed. Thus, we optimize it in an alternating manner - iteratively fixing one parameter and solving the other with a closed-form solution. We give the closed-form solutions for both parameters in the following.

Fix $\hat{\mathbf{U}}_\tau$:

$$\begin{aligned} \hat{\boldsymbol{\theta}}^s_\tau &= \mathbf{C}_\tau^{-1}\mathbf{d}_\tau, \\ \mathbf{C}_\tau &= \beta(\mathbf{X}^s)^T\mathbf{X}^s + \hat{\mathbf{U}}^T_\tau(\mathbf{X}^t_\tau)^T\mathbf{X}^t_\tau\hat{\mathbf{U}}_\tau + \mathbf{I}_{d_s}, \\ \mathbf{d}_\tau &= \beta(\mathbf{X}^s)^T\mathbf{r}^s + \hat{\mathbf{U}}^T_\tau(\mathbf{X}^t_\tau)^T\mathbf{r}^t. \end{aligned} \tag{5}$$

Fix $\hat{\boldsymbol{\theta}}^s_\tau$:

$$\begin{aligned}
&\text{vec}(\hat{\mathbf{U}}_\tau) = \mathbf{A}_\tau^{-1}\mathbf{b}_\tau,\\
&\mathbf{A}_\tau = \hat{\boldsymbol{\theta}}^s_\tau(\hat{\boldsymbol{\theta}}^s_\tau)^T \otimes (\mathbf{X}^t_\tau)^T\mathbf{X}^t_\tau + \gamma\mathbf{I}_{d_s} \otimes (\mathbf{X}^t_\tau)^T\mathbf{S}_1\mathbf{X}^t_\tau + \mathbf{I}_{d_t d_s},\\
&\mathbf{b}_\tau = \text{vec}((\mathbf{X}^t_\tau)^T\mathbf{r}^t_\tau(\hat{\boldsymbol{\theta}}^s_\tau)^T + \gamma(\mathbf{X}^t_\tau)^T\mathbf{S}\mathbf{X}^s_{N_s}).
\end{aligned} \tag{6}$$

For computational efficiency, we instead solve $\hat{\boldsymbol{\theta}}^s_\tau$ and $\hat{\mathbf{U}}_\tau$ using conjugate gradient descent method in our experiments.

In each step τ, a pure exploitation policy recommends the article with the largest predicted reward. Thus, the pure exploitation strategy is,

$$a_\tau = \arg\max_{a\in\mathcal{A}^t}(\mathbf{x}^t_a)^T\hat{\mathbf{U}}_\tau\hat{\boldsymbol{\theta}}^s_\tau. \tag{7}$$

Exploration in TCB

In contextual bandit problems, only the rewards of pulled arms are observed and they are stochastic. Purely exploiting following the recommendation strategy in Eq. 7, therefore, easily get stuck in the suboptimal arms and suffers from the linear regret growth which is suboptimal.

In this section, we introduce how TCB simultaneously exploits and explores. To explore, TCB is based on the Upper-Confidence-Bound (UCB) (Auer 2002) principle. UCB style methods follow three steps: first, a high probability confidence set for each uncertain parameter is constructed; second, the UCB of the expected reward is calculated for each arm; third, the action with the largest UCB is pulled in each step.

As shown in Fig. 1, there are three sources of uncertainty in our cross-domain RecSys including the correspondence data, the source rewards, and the target rewards. Accordingly, TCB explores the uncertainty of both the reward parameter $\hat{\boldsymbol{\theta}}^s_\tau$ and the translation matrix $\hat{\mathbf{U}}_\tau$. We firstly define the high probability confidence sets for both parameters in the following inequalities,

$$\begin{aligned}
&||\hat{\boldsymbol{\theta}}_\tau - \boldsymbol{\theta}^s_*||_{\mathbf{C}_\tau} \le \alpha_\theta,\\
&||\text{vec}(\hat{\mathbf{U}}_\tau) - \text{vec}(\mathbf{U}_*)||_{\mathbf{A}_\tau} \le \alpha_U
\end{aligned} \tag{8}$$

where α_θ and α_U denote the upper bounds of the confidence sets and are discussed in Lemma 1. Based on the mechanisms in Eq. 1 and Eq. 3, for $\boldsymbol{\theta}^s_*$ and $\mathbf{U}_*$ within the confidence set, we then calculate the UCB of the expected reward for each target action with the context $\mathbf{x}^t_a$:

$$\begin{aligned}
\text{UCB}(\mathbf{x}^t_a) =& (\mathbf{x}^t_a)^T\hat{\mathbf{U}}_\tau\hat{\boldsymbol{\theta}}^s_\tau\\
&+ \alpha_\theta||(\mathbf{x}^t_a)^T\hat{\mathbf{U}}_\tau||_{\mathbf{C}^{-1}_\tau} + \alpha_U||\hat{\boldsymbol{\theta}}^s_\tau \otimes \mathbf{x}^t_a||_{\mathbf{A}^{-1}_\tau}.
\end{aligned} \tag{9}$$

Finally, TCB's recommendaion strategy is

$$a_\tau = \arg\max_{a\in\mathcal{A}^t}\text{UCB}(\mathbf{x}^t_a). \tag{10}$$

In Eq. 9, the first term of UCB calculates the expected reward which encourages more exploitation. The second term explores the reward parameter $\hat{\boldsymbol{\theta}}^s_\tau$. TCB, therefore, tends to select the actions that help learn the reward function quickly. More informative source observations incur shrinkage of the second term and thereby accelerate the exploration. The third term explores the translation matrix $\hat{\mathbf{U}}_\tau$, i.e., how to transfer. Thus, TCB favors the actions that help learn the translation quickly. In short, UCB in Eq. 9 promotes transfer learning and the bandit policy to benefit each other. Additionally, the second and third terms of UCB are in shrinkage as TCB accumulates more target observations. As a result, TCB explores more in the beginning and exploits more in later stages. We present the details of the TCB policy in Algorithm 1.

It is worth noting that TCB can also handle knowledge transfer between homogeneous domains. When $\hat{\mathbf{U}}_\tau$ is fixed to be $\mathbf{I}_{d_t}$, TCB degenerates to LinUCB with warm start. When two domains lie the same feature space but different distributions, TCB utilizes the translation to align distributions.

Algorithm 1 Transferable Contextual Bandit

Input: Source Observations $\mathbf{O}^s_{N^s}$, β, γ, α_U and α_θ.
Initialize: $\hat{\mathbf{U}}_1 = \mathbf{0}_{d^t,d^s}$, $\hat{\boldsymbol{\theta}}^s_1 = \mathbf{0}_{d^s,1}$.
for $\tau = 1 \dots N^t$ in the target domain **do**
  Observe the action set $\mathcal{A}^t_\tau$ and context $\mathbf{x}^t_{a\in\mathcal{A}^t_\tau}$;
  Calculate the UCB for $a \in \mathcal{A}^t_\tau$ according to Eq. 9;
  Pull arm $a_\tau = \arg\max_{a\in\mathcal{A}_\tau}\text{UCB}(\mathbf{x}^t_a)$;
  Observe reward $r^t_{a_\tau}$;
  while Until Convergence **do**
    With new observation $(\mathbf{x}^t_{a_\tau}, r^t_{a_\tau})$;
    Update $\hat{\boldsymbol{\theta}}^s_{\tau+1}$ and $\mathbf{C}_{\tau+1}$ according to Eq. 5;
    Update $\hat{\mathbf{U}}_{\tau+1}$ and $\mathbf{A}_{\tau+1}$ according to Eq. 6;
  end while
end for

Regret Analysis

The overall loss function in Eq. 4 is not jointly convex if both parameters are optimized together. Since analyzing the convergence to the true parameters in such non-convex problems is beyond the focus of this paper, we conduct regret analysis on TCB by solving $\mathbf{U}$ and $\boldsymbol{\theta}^s$ separately in Eq. 11. The true parameter of $\mathbf{U}_*$ is known when solving $\hat{\boldsymbol{\theta}}^s_\tau$, and vice versa.

$$\begin{aligned}
\hat{\boldsymbol{\theta}}^s_\tau =& \arg\min_{\boldsymbol{\theta}^s}\beta||\mathbf{r}^s_{N_s} - \mathbf{X}^s_{N_s}\boldsymbol{\theta}^s||_2^2\\
&+ ||\mathbf{r}^t_\tau - \mathbf{X}^t_\tau\mathbf{U}_*\boldsymbol{\theta}^s||_2^2 + ||\boldsymbol{\theta}^s||_2^2,\\
\hat{\mathbf{U}}_\tau =& \arg\min_{\mathbf{U}}||\mathbf{r}^t_\tau - \mathbf{X}^t_\tau\mathbf{U}\boldsymbol{\theta}^s_*||_2^2\\
&+ \gamma\sum_{i=1}^{\tau}\sum_{j=1}^{N_s}\mathbf{S}_{i,j}||\mathbf{U}^T\mathbf{x}^t_i - \mathbf{x}^s_j||_2^2 + ||\mathbf{U}||_F^2.
\end{aligned} \tag{11}$$

Lemma 1 illustrates that the confidence sets of $\hat{\boldsymbol{\theta}}^s_\tau$ and $\hat{\mathbf{U}}_\tau$ grow sublinearly with respect to the number of steps N_t.

Lemma 1 *Suppose* $||\mathbf{U}_*||_F \le c_u$, $||\boldsymbol{\theta}^s_*||_2 \le c_\theta$, $||\mathbf{x}^s||_2 \le 1$ *and* $||\mathbf{x}^t||_2 \le 1$. *If each target observation has one corresponding source observation, then with probability* $1-\delta$

$$\begin{aligned}
&||\hat{\boldsymbol{\theta}}_{N_t} - \boldsymbol{\theta}^s_*||_{\mathbf{C}_{N_t}}\\
\le& c_\theta + \beta\sqrt{\log\frac{3\det(\mathbf{C}_{N_t})^{1/2}}{\det(\hat{\mathbf{U}}^T_{N_t}(\mathbf{X}^t_\tau)^T\mathbf{X}^t_\tau\hat{\mathbf{U}}_{N_t} + \mathbf{I}_{d_s})^{1/2}\delta}}\\
&+ (1 + c_\theta c_\eta)\sqrt{\log\frac{3\det(\mathbf{C}_{N_t})^{1/2}}{\det(\beta(\mathbf{X}^s)^T\mathbf{X}^s + \mathbf{I}_{d_s})^{1/2}\delta}}\\
\le& c_\theta + (1 + c_\theta c_\eta + \beta)\sqrt{d_s\log\frac{1+N_t}{\delta}}
\end{aligned} \tag{12}$$

And

$$
\begin{aligned}
&||vec(\hat{\mathbf{U}}_\tau) - vec(\mathbf{U}_*)||_{\mathbf{A}_\tau} \\
\leq & c_U + \gamma c_\eta \sqrt{\log \frac{3\det(\mathbf{A}_{N_t})^{1/2}}{\det(\hat{\boldsymbol{\theta}}_{N_t}(\hat{\boldsymbol{\theta}}_{N_t})^T \otimes (\mathbf{X}_\tau^t)^T\mathbf{X}_\tau^t + \mathbf{I}_{d_t d_s})^{1/2}\delta}} \\
& + (1 + c_\eta c_\theta)\sqrt{\log \frac{3\det(\mathbf{A}_{N_t})^{1/2}}{\det(\gamma\mathbf{I}_{d_s} \otimes ((\mathbf{X}_\tau^t)^T\mathbf{S}_1\mathbf{X}_\tau^t) + \mathbf{I}_{d_t d_s})^{1/2}\delta}} \\
\leq & c_U + (1 + c_\eta c_\theta + c_\eta\gamma)\sqrt{d_s d_t \log \frac{1+N_t}{\delta}} \qquad (13)
\end{aligned}
$$

With Lemma 1, we further present the the upper bound of TCB's cumulative regret in the next. As in Eq. 14, the regret quantifies a policy's effectiveness by the difference between the expected reward of the optimal arm a_τ^* and that of the pulled arm a_τ (Burtini, Loeppky, and Lawrence 2015).

$$
R(N_t) = \sum_{\tau=1}^{N_t}(\mathbb{E}(r_{a_\tau^*}^t) - \mathbb{E}(r_{a_\tau}^t)) \qquad (14)
$$

Theorem 1 *Under the same assumptions as Lemma 1, with probability* $1-\delta$*, TCB's cumulative regret satisfies*

$$
\begin{aligned}
R(N_t) \leq & 2\alpha_U\sqrt{N_t \log \frac{\det(\mathbf{A}_{N_t})}{\det(\gamma\mathbf{I}_{d_s} \otimes ((\mathbf{X}_{N_t}^t)^T\mathbf{S}_1\mathbf{X}_{N_t}^t) + \mathbf{I}_{d_t d_s})}} \\
& + 2\alpha_\theta\sqrt{N_t \log \frac{\det(\mathbf{C}_{N_t})}{\det(\beta(\mathbf{X}^s)^T\mathbf{X}^s + \mathbf{I}_{d_s})}} \qquad (15) \\
\leq & 2\alpha_U\sqrt{N_t d_t \log(1+N_t)} + 2\alpha_\theta\sqrt{N_t d_s \log(1+N_t)}
\end{aligned}
$$

The detailed proofs are in the supplementary material[1]. According to Lemma 1 and Theorem 1, we conclude that TCB enjoys the same order of sublinear regret as the standard LinUCB, i.e. $O(\sqrt{N_t}\log(N_t))$ (Abbasi-Yadkori, Pál, and Szepesvári 2011; Li et al. 2010).

We further discuss the influences of knowledge transfer on TCB. According to Eq. 15, when the largest eigenvalue of $(\mathbf{X}^s)^T\mathbf{X}^s$ is nonzero, TCB's regret decreases by a constant. Alternatively speaking, transfer learning improves the contextual bandit policy if informative source data are provided. Similarly, in Eq. 12, the confidence set of $\hat{\boldsymbol{\theta}}_\tau^s$ shrinks with the eigenvalues of the source observations. Thus, transfer learning accelerates the exploration of the reward function.

Transfer learning always assumes that a source domain and a target domain are related but different. In our work, we describe the difference between domains with a random vector $\boldsymbol{\eta}$. As we mentioned before, each element of $\boldsymbol{\eta}$ is $c_\eta/\sqrt{2}$-sub-Gaussian. Intuitively, a random vector with a smaller c_η is expected if two domains are near. A smaller c_η will introduce faster shrinkage on the confidence sets of both $\boldsymbol{\theta}_*^s$ and $\mathbf{U}_*$ according to Lemma 1, and achieve a better regret according to Theorem 1. In summary, the exploration of TCB becomes much more efficient when a pair of source and target domains are more related.

[1]Supplementary material is available at `http://www.cse.ust.hk/~bliuab`

Experiments

In this section, we evaluate TCB's empirical performance using one synthetic and two real-world datasets. We compare TCB with four representative baselines, including HTL (Wei et al. 2016), LinUCB (Li et al. 2010), hLinUCB (Wang, Wu, and Wang 2016), and FactorUCB (Wang, Wu, and Wang 2017). HTL is the pure exploitation strategy with transfer learning as shown in Eq. 7. LinUCB, hLinUCB, and FactorUCB are the contextual bandit baselines without transfer. LinUCB is the most widely used contextual bandit baseline with the assumption of a linear reward function. hLinUCB[2] explicitly models the hidden context and claims to be robust to the cold-start problem. FactorUCB learns the latent user factors and latent item factors as well as their UCB in a collaborative filtering manner. FactorUCB requires the knowledge of user clustering. Thus, we only compare with FactorUCB in the Delicious dataset.

For all experiments, we fix TCB's hyperparameter as $\beta = 0.1$ and $\gamma = 0.1$ such that all terms in Eq. 4 are in the same order of magnitude. To be consistent with the canonical LinUCB, we set the regularization hyperparameter of all methods to be one and exploration hyperparameter to be 0.2. We mainly investigate TCB's performance in the cold-start situation. Thus, we plot the cumulative reward within 1,000 steps in the synthetic experiments and 5,000 steps in the real-world experiments. All reported results are averaged by 10-times random experiments.

Synthethic Experiment

In this part, we build the synthetic environment to verify TCB's effectiveness. With the oracle knowledge of the reward function, we measure the performance with cumulative regret as shown in Eq 14.

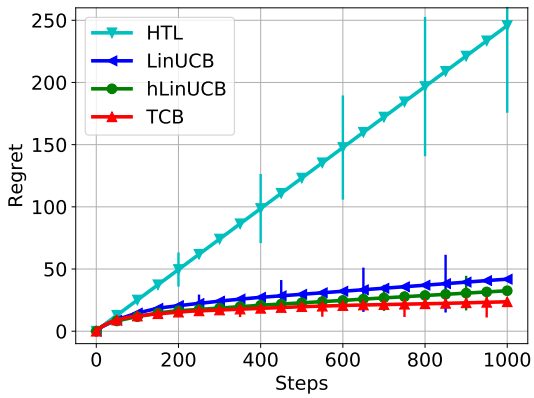
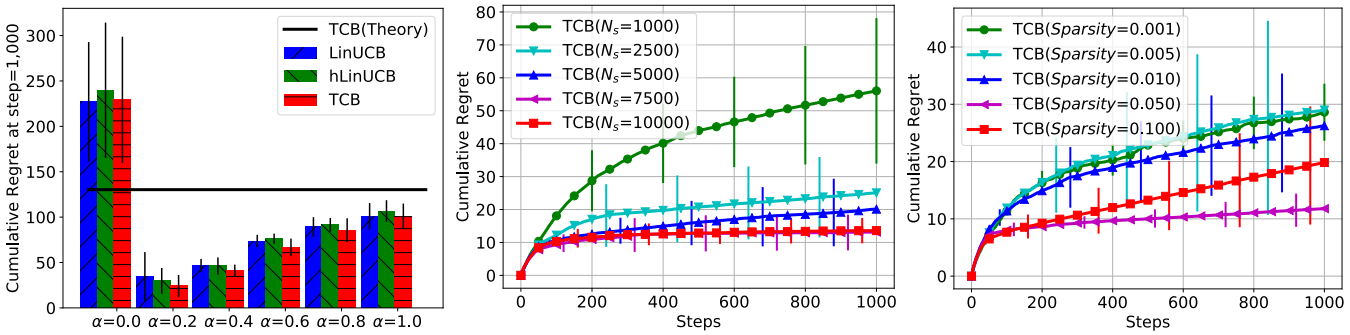


Figure 2: Cumulative regret w.r.t. the steps in synthetic experiments

We firstly generate the source and target context by randomly drawing each dimension from a Gaussian distribution, i.e., $\mathcal{N}(0,\sigma)$ and $\sigma \sim \mathcal{U}(0,1)$. Similarly, the optimal translation matrix $\mathbf{U}_*$ and the reward parameter $\boldsymbol{\theta}_*^s$ are randomly drawn from a uniform distribution $\mathcal{U}(0,1)$. To be consistent

[2]hLinUCB and FactorUCB are both available at `github.com/huazhengwang/BanditLib`



(a) Regret of three policies with respect to the varying α

(b) Regret of TCB with the different number of source observations

(c) Regret of TCB using the similarity matrix of different sparsity

with Lemma 1, we normalize the data such that the norm of each context, of the translation, and of the reward parameter are upper bounded by one. We further use the Gaussian similarity to measure the correspondence between jth source context and ith target context, i.e., $\mathbf{S}_{i,j} = e^{-||\mathbf{U}_*^T \mathbf{x}_i^t - \mathbf{x}_j^s||_2^2}$. We keep the top 1% largest similarity and set the remainings to be zero. In each step of the simulation, all compared policies face 5,000 actions and suffer from the same reward noise $\epsilon \sim \mathcal{N}(0, 1)$. Additionaly, we set the $d_s = 60$, $d_t = 50$, and $N_s = 10,000$. On average, TCB costs 1.38 seconds to decide one action and update the UCB.

In Fig. 2, we plot the cumulative regret w.r.t. the number of steps. Apparently, HTL, as a pure exploitation strategy, suffers from the linearly growing regret which is the worst. TCB and HTL both transfer the knowledge from the same source observations. In comparison, TCB explores the uncertainty of the translation and the reward function. On behalf of the exploration, TCB learns the translation and the reward function more effectively and efficiently, thereby achieving the improved transferring performance.

In comparison with LinUCB and hLinUCB which do not transfer, we see that TCB's regret converges significantly faster. We, therefore, attribute TCB's superiority to the knowledge transfer. TCB leverages the knowledge from the source observations to obtain the more accurate parameters and tighter upper confidence bound. Additionally, across ten-times random experiments, TCB achieves more stable and robust regrets according to the smaller variance. In conclusion, Fig. 2 consolidates our claim that TCB's exploration and knowledge transfer mutually benefit each other.

The hyperparameters α_θ and α_U balance the exploitation and exploration and directly decide the performance of TCB. In Fig. 3a, we investigate how $\alpha_U = \alpha_\theta = \alpha$ influences the cumulative regret. According to Fig 3a, for all $\alpha > 0$, TCB consistently outperforms LinUCB and hLinUCB, which further consolidates TCB's superiority. When $\alpha = 0$, all bandit approaches degenerate to the pure exploitation and suffer from the inferior regret and larger variance. When $\alpha > 0.2$, all methods explore more. More exploration, however, does not compensate the sacrificed short-term rewards, and achieves worse cumulative regret. Moreover, TCB(Theory) sets α_U and α_θ according to Lemma 1. Obviously, TCB with manually tuned α outperforms the TCB(Theory). As a result, $\alpha = 0.2$ is a fair choice for all methods.

The source observations and the correspondence data are the essential components of TCB. Theorem 1 requires that each target observation has one corresponding source observation, which is not easy to hold in a real-world RecSys. Thus, we investigate the effects of the source observations and the correspondence data on TCB's performance.

In Fig. 3b, we examine how TCB's regret changes w.r.t. the varying number of source observations. Apparently, TCB's regret continues to deteriorate with the decreasing number of source observations. When only 1,000 source observations are available, TCB performs even worse than LinUCB and hLinUCB without transfer. The potential explanation is that TCB explores both the translation and the reward function the combination of which are more uncertain. The insufficient source observations, however, cannot accelerate the exploration of TCB enough, thereby leading to the worse performance. On the other hand, Fig. 3b signifies that TCB is capable of utilizing more and more source data.

The correspondence data $\mathbf{S}$ is the guidance for exploring the translation matrix $\mathbf{U}_*$. In Fig. 3c, we present TCB's cumulative regret when $\{0.1\%, 0.5\%, 1\%, 5\%, 10\%\}$ entries of $\mathbf{S}$ are non-zero. Given very sparse correspondence data, the exploration and learning of $\mathbf{U}_*$ turn slow due to the insufficient constraints. When TCB observes denser correspondence data, its regret firstly decreases significantly and then increases. For sparsity$= 0.1$, TCB projects each target context to be similar to 10% source observations, so that the target context becomes indistinguishable in the source feature space, thereby leading to worse regret. Finally, to further analyze TCB's performance, we investigate how TCB's regret changes with respect to the dimension of the context in Fig. 6 in the supplementary material.

Real-world Experiments

In this section, we empirically compare TCB with the state-of-the-art baselines using the **Cheetah** data and the public **Delicious**[3] data. Without the knowledge of true reward function, we compare all methods using the cumulative reward. Due to the privacy issues, all reported cumulative rewards are normalized by a random policy.

[3]Dataset is available at `grouplens.org/datasets/hetrec-2011`

The Cheetah dataset is from the real-world RecSys supplied by Cheetah Mobile Inc. The app and the article RecSys are the source and the target domains, respectively. The user profiles and app attributes together serve as the context for app source RecSys. We combine the user profiles and article titles to be the target context. We further adopt principal components analysis (PCA) to reduce the source and the target context to 50 and 30 dimensions, respectively. Therefore, in Cheetah dataset, TCB faces not only different actions but also heterogeneous feature spaces of two domains.

In each step τ of our experiment, we randomly select ten articles from all as the candidate set. We guarantee that the serving user reads exactly one candidate. All compared methods are required to choose one recommendation from the candidate set. If the serving user reads the recommended article, the agent receives the reward as one, and otherwise zero. We serve all users in chronological order. In the same way, we accumulate the source app observations using a random policy. Finally, we construct the correspondence data $\mathbf{S}$ for TCB and HTL. The ith target observation corresponds to the jth source observation if they satisfy the following two criteria: first, the ith target and jth source observations serve the same user; second, the user feedbacks are the same for the ith target and jth source observations. For example, one user installs the jth app and reads the ith article, then $\mathbf{S}_{ij} = 1$. We summarize the Cheetah data in Table 1.

Table 1: Statistics of Cheetah Data

# of unique users	# of correspondence per user	# of users with correspondence
4,000	1.077	1,395
# of unique articles	# of source observations	time span
3,706	11,770	21 Days

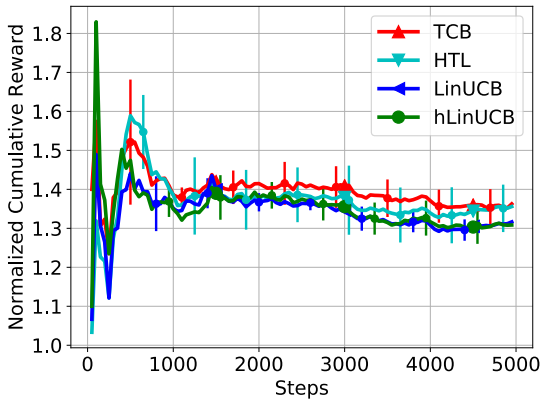


Figure 4: Normalized Reward on Cheetah Data w.r.t. steps

In Fig. 4, we empirically verify the necessity of knowledge transfer to improve the contextual bandit methods. When the target observations are fewer than 500, calculating the cumulative reward is high-variance and noisy. As a result, all methods are not distinguishable in this period. When the step is larger than 500, on behalf of transfer learning, TCB and HTL consistently outperform LinUCB and hLinUCB. At a later stage, all algorithms still suffer from the cold-start problem. TCB and HTL, however, learns the reasonable translation from the correspondence data. In the translated feature space, TCB and HTL estimate the reward function more accurately using both source and target observations. In comparison, LinUCB and hLinUCB only learn from the insufficient target observations and achieve inferior rewards.

By comparing TCB with HTL in Fig. 4, we can empahsize the importance of exploration to transfer learning. TCB, unfortunately, is outperformed by HTL in the early period. The potential reason is that TCB sacrifices the short term reward to explore both the translation and the reward function. In the long run, due to the exploration, TCB estimates the reward function far more efficiently than HTL without exploration. As a result, when more observations are accumulated, TCB quickly bypasses HTL and shows consistent advantages.

Finally, we verify that TCB can also be applied to the homogeneous problem using the public Delicious dataset. We perform the experiments in the same setting as (Cesa-Bianchi, Gentile, and Zappella 2013). After preprocessing, we obtain 1,867 users, 57,784 URLs, and the corresponding 25-dimensional textual context. In each step τ, the agent recommends one URL among 25 candidates. The agent receives the reward as one if the user bookmarks the URL. For TCB, we design the source and the target RecSys to have mutual exclusive URLs. Specifically, the URL is the source if it owns a tag that occurs more than 80 times, otherwise target. Therefore, the source and target URLs are described in the same textual feature space but different distributions of text. The source observations and the correspondence are constructed in the same way as Cheetah data. hLinUCB and FactorUCB consider the user clustering for collaborative filtering. For a fair comparison, we carry on the independent TCB, HTL, and LinUCB on each user clustering.

According to Fig. 5, TCB's two-way advantage is again proved in the homogeneous problem. For one thing, by comparing TCB with hLinUCB and FactorUCB, we conclude that, for the cold-start problem, knowledge transfer from an auxiliary domain is more effective than collaborative filtering within a single domain. For another, TCB's knowledge transfer is more efficient than HTL.

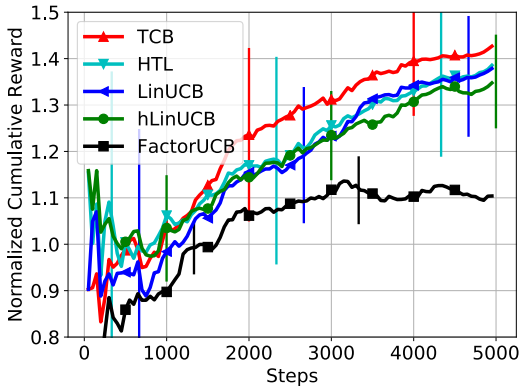


Figure 5: Normalized Reward on Delicious Data w.r.t. steps

Conclusions

In this paper, we propose a transferable contextual bandit policy named TCB for the cross-domain recommendation. TCB adopts transfer learning to optimize the cumulative reward in the target RecSys. On behalf of transfer learning, TCB benefits the exploitation and accelerates the exploration in the target RecSys. On behalf of the contextual bandit policy, TCB efficiently explores how to transfer and how to recommend. The theoretical regret analysis and empirical experiments verify TCB's superiority. In the future, we plan to speed up TCB in the high-dimensional context and deploy it in a real online recommendation system.

Acknowledgements

We are supported by National Grant Fundamental Research (973 Program) of China under Project 2014CB340304 and Hong Kong CERG projects 16211214, 16209715 and 16244616. Yu Zhang is supported by NSFC (61473087, 61673202) and the Natural Science Foundation of Jiangsu Province (BK20141340).

References

Abbasi-Yadkori, Y.; Pál, D.; and Szepesvári, C. 2011. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems*, 2312–2320.

Agrawal, S., and Goyal, N. 2013. Thompson sampling for contextual bandits with linear payoffs. In *Proceedings of the 30th International Conference on Machine Learning*, 127–135.

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

Azar, M. G.; Lazaric, A.; and Brunskill, E. 2013. Sequential transfer in multi-armed bandit with finite set of models. In *Advances in Neural Information Processing Systems 26*, 2220–2228.

Burtini, G.; Loeppky, J.; and Lawrence, R. 2015. A survey of online experiment design with the stochastic multi-armed bandit. *CoRR* abs/1510.00757.

Cesa-Bianchi, N.; Gentile, C.; and Zappella, G. 2013. A gang of bandits. In *Advances in Neural Information Processing Systems 26*, 737–745.

Chu, W.; Li, L.; Reyzin, L.; and Schapire, R. E. 2011. Contextual bandits with linear payoff functions. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, 208–214.

Dai, W.; Chen, Y.; Xue, G.; Yang, Q.; and Yu, Y. 2008. Translated learning: Transfer learning across different feature spaces. In *Advances in Neural Information Processing Systems 21*, 353–360.

Li, L.; Chu, W.; Langford, J.; and Schapire, R. E. 2010. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th International Conference on World Wide Web*, 661–670. ACM.

Li, L.; Chu, W.; Langford, J.; and Wang, X. 2011. Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. In *Proceedings of the Forth International Conference on Web Search and Web Data Mining*, 297–306.

Li, S.; Karatzoglou, A.; and Gentile, C. 2016. Collaborative filtering bandits. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, 539–548.

Pan, S. J., and Yang, Q. 2010. A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.* 22(10):1345–1359.

Pan, W.; Xiang, E. W.; Liu, N. N.; and Yang, Q. 2010. Transfer learning in collaborative filtering for sparsity reduction. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence,*.

Pan, W.; Liu, N. N.; Xiang, E. W.; and Yang, Q. 2011. Transfer learning to predict missing ratings via heterogeneous user feedbacks. In *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011*, 2318–2323.

Schein, A. I.; Popescul, A.; Ungar, L. H.; and Pennock, D. M. 2002. Methods and metrics for cold-start recommendations. In *SIGIR 2002: Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 253–260.

Shi, X.; Liu, Q.; Fan, W.; and Yu, P. S. 2013. Transfer across completely different feature spaces via spectral embedding. *IEEE Trans. Knowl. Data Eng.* 25(4):906–918.

Tran-Thanh, L.; Chapman, A. C.; de Cote, E. M.; Rogers, A.; and Jennings, N. R. 2010. Epsilon-first policies for budget-limited multi-armed bandits. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence,*.

Wang, H.; Wu, Q.; and Wang, H. 2016. Learning hidden features for contextual bandits. In *Proceedings of the 25th ACM International Conference on Information and Knowledge Management*, 1633–1642.

Wang, H.; Wu, Q.; and Wang, H. 2017. Factorization bandits for interactive recommendation. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, 2695–2702.

Wei, Y.; Zhu, Y.; Leung, C. W.; Song, Y.; and Yang, Q. 2016. Instilling social to physical: Co-regularized heterogeneous transfer learning. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, 1338–1344.

Yue, Y.; Hong, S. A.; and Guestrin, C. 2012. Hierarchical exploration for accelerating contextual bandits. In *Proceedings of the 29th International Conference on Machine Learning*.

Zhang, J., and Bareinboim, E. 2017. Transfer learning in multi-armed bandits: A causal approach. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, 1340–1346.

Zhou, L. 2015. A survey on contextual multi-armed bandits. *CoRR* abs/1508.03326.

Zhu, Y.; Chen, Y.; Lu, Z.; Pan, S. J.; Xue, G.; Yu, Y.; and Yang, Q. 2011. Heterogeneous transfer learning for image classification. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*.