pdfauthor

Reinforcement Learning for Online Information Seeking

Xiangyu Zhao, Michigan State University
Long Xia, JD.com
Jiliang Tang, Michigan State University
Dawei Yin, JD.com

Information seeking techniques, satisfying users' information needs by suggesting users personalized objects (information or services) at the appropriate time and place, play a crucial role in mitigating the information overload problem on the Web. With recent great advances in Reinforcement Learning (RL), there have been increasing interests in developing RL based information seeking techniques. These RL based techniques have two key advantages – (1) they are able to continuously update information seeking strategies according to users' real-time feedback, and (2) they can maximize the expected cumulative long-term reward from users where reward has different definitions according to information seeking applications such as click-through rate, revenue, user satisfaction and engagement. In this survey, we give an overview about reinforcement learning for information seeking on the web from methodologies to applications, review representative algorithms, and discuss some appealing research directions.

1. INTRODUCTION

The explosive growth of the world-wide web has generated massive data. As a consequence, the information overload problem has become progressively severe [Chang et al. 2006]. Thus, how to identify objects that satisfy users' information needs at the appropriate time and place has become increasingly important, which has motivated three representative information seeking mechanisms – search, recommendation, and advertising. The search mechanism outputs objects that match the query, the recommendation mechanism generates a set of items that match users' implicit preferences, and the advertising mechanism is analogous to search and recommendation expect that the objects to be presented are advertisements [Garcia-Molina et al. 2011]. Numerous efforts have been made on designing intelligent methods for these three information seeking mechanisms. However, traditional techniques often face several common challenges. First, the majority of existing methods consider information seeking as a static task and generate objects following a fixed greedy strategy. This may fail to capture the dynamic nature of users preferences (or environment). Second, most traditional methods are developed to maximize the short-term reward, while completely neglecting whether the suggested objects will contribute more in long-term reward [Shani et al. 2005]. Note that the reward has different definitions among information seeking tasks, such as click-through rate (CTR), revenue, and dwelling time.

Recent years have witnessed the rapid development of reinforcement learning (RL) techniques and a wide range of RL based applications. Under the RL schema, we tackle complex problems by acquiring experiences through interactions with a dynamic environment. The result is an optimal policy that can provide solutions to complex tasks without any specific instructions [Kaelbling et al. 1996]. Employing RL for information seeking can naturally resolve the aforementioned challenges. First, considering the information seeking tasks as sequential interactions between an RL agent (system) and users (environment), the agent can continuously update its strategies according to users' real-time feedback during the interactions, until the system converges to the optimal policy that generates objects best match users dynamic preferences. Second, the RL frameworks are designed to maximize the long-term cumulative reward from users. Therefore, the agent could identify objects with small immediate reward but making big contributions to the reward in the long run.

Given the advantages from reinforcement learning, there have been tremendous interests in developing RL based information seeking techniques. Thus, it is timely and necessary to provide an overview about information seeking techniques from a reinforcement learning perspective. In this survey, we present a comprehensive overview on state-of-the-art RL based information seeking techniques and discuss some future directions. The remaining of the survey is organized as follows. In Section 2, we introduce technical foundations of reinforcement learning based information seeking techniques. Then we review three key information seeking tasks with representative algorithms from Sections 3 to 5. Finally, we conclude the work with several future research directions.

2. TECHNICAL FOUNDATIONS

Reinforcement learning is learning how to map situations to actions [Sutton and Barto 1998]. The two fundamental elements in RL are to formulate the situations (mathematical models) and to learn the map (policy learning).

2.1 Problem Formulation

In reinforcement learning, there two main settings for problem formulations: multi-armed bandits (without state transition) and Markov decision process (with state transition).

2.1.1 *Multi-Armed Bandits.* The Multi-Armed Bandits (MAB) problem is a simple model for the exploration/exploitation trade-off [Varaiya and Walrand 1983]. Formally, a K-MAB can be defined as follows.

Definition 2.1. A K-MAB is a 3-tuple $\langle A, R, \pi \rangle$, where A is the set of actions (arms) and $|A| = K$, $r = R(a)$ is the reward distribution when performing action a, and policy π describes probability distribution over the possible actions.

An arm with the highest expected reward is called the *best arm* (denoted as a_*) and its expected reward r_* is the *optimal reward.* An algorithm for MAB, at each time step t, samples an arm a_t and receives a reward r_t. When making its selection, the algorithm depends on the history (i.e., actions and rewards) up to the time t-1. The contextual bandit model (a.k.a. associative bandits or bandits with side information) is an extension of MAB that takes additional information into account [Auer et al. 2002; Lu et al. 2010].

2.1.2 *Markov Decision Process.* Markov decision process (MDP) is a classical formalization of sequential decision making, which is a mathematically idealized form of reinforcement learning problem [Bellman 2013]. We define an MDP as follows.

Definition 2.2. A Markov Decision Process is a 5-tuple $\langle S, A, T, R, \pi \rangle$, where S is a set of states, A is a discrete set of actions, T is the state transition function $s_{t+1} = T(s_t, a_t)$, which specifies a function mapping a state s_t into a new state s_{t+1} in response to the selected action a_t, $r = R(s, a)$ is the reward distribution when performing action a in state s, and policy $\pi(a|s)$ describes the behaviors of an agent which is a probability distribution over the possible actions.

The agent and environment interact at each of a sequence of discrete time steps $t = \{0, 1, 2, \ldots\}$. Consequently, a sequence or *trajectory* is generated as $\{s_0, a_0, r_1, \cdots, s_t, a_t, r_{t+1}, \cdots\}$. In general, we seek to maximize the *expected discounted return*, where the return G_t is defined as: $G_t = \sum_{k=0}^{\infty} \gamma^k r_{r+k+1}$, where γ $(0 \leq \gamma \leq 1)$ is the *discounted rate*. The Partially Observable Markov Decision Process (POMDP) is an extension of MDP to the case where the state of the system is not necessarily observable [Åström 1965; Smallwood and Sondik 1973; Sondik 1978; Kaelbling et al. 1998].

2.1.3 *Multi-Agent Setting.* The generalization of the Markov Decision Process to the Multi-Agent case is the stochastic game [Bowling and Veloso 2002; Shoham et al. 2003; Busoniu et al. 2008] as:

Definition 2.3. A multi-agent game is a tuple $\langle S, A_1, \ldots, A_n, T, R_1, \ldots, R_n, \pi_1, \ldots, \pi_n \rangle$, where n is the number of agents, S is the discrete set of environment states, A_i is the discrete set of actions for the agent i, T is the state transition probability function, R_i is the reward function of agent i, and π_i is the policy adopted by agent i.

In the multi-agent game, the state transition is the result of the joint actions of all the agents $\mathbf{a}_t = [a_{1,t}^T, \ldots, a_{n,t}^T]^T$, where $a_{i,t} \in A_i$ denotes the action taken by agent i at time step t. The reward $r_{i,k+1}$ also depends on the joint action. If $\pi_1 = \cdots = \pi_n$, i.e., all the agents adopt the same policy to maximize the same expected return, the multi-agent game is fully cooperative. If $n = 2$ and $\pi_1 = -\pi_2$, i.e., the two agents have opposite policies, the game is fully competitive. Mixed games are stochastic games that are neither fully cooperative nor fully competitive.

2.2 Policy learning

Reinforcement Learning is a class of learning problems in which the goal of an agent (or multi-agent) to find the *policy* to optimize some measures of its long-term performance. RL solutions can be categorized in different ways. Here we investigate them from two perspectives: whether the full model is available and the way of finding the optimal policy.

2.2.1 *Model-based v.s. Model-free.* Reinforcement learning algorithms, which explicitly learn system models and use them to solve MDP problems, are model-based methods. Model-based RL has a strong influence from the control theory and is often explained in terms from different disciplines. These methods include popular algorithms such as the Dyna [Sutton 1991], Prioritized Sweeping [Moore and Atkeson 1993], *Q*-iteration [Busoniu et al. 2010], Policy Gradient (PG) [Williams 1992], and the variation of PG [Baxter and Bartlett 2001; Kakade 2001]. The model-free methods ignore the model and just focus on figuring out the value functions directly from the interaction with the environment. To accomplish this, the methods depend on sampling and observation heavily; thus they don't need to know the inner working of the system. Some examples of these methods are Q-learning [Kröse 1995], SARSA [Rummery and Niranjan 1994], LSPI [Lagoudakis and Parr 2003], and Actor-Critic [Konda and Tsitsiklis 1999].

2.2.2 *Value function v.s. Policy search.* The algorithms, which first find the optimal value functions and then extract optimal policies, are value function methods, such as Dyna, Q-learning, SARSA, and DQN [Mnih et al. 2015]. The alternative approaches are policy search methods which solve an MDP problem by directly searching in the space of policies. An important class of policy search methods is that of Policy Gradient (PG) algorithms [Williams 1992; Baxter and Bartlett 2001; Kakade 2001; Deisenroth and Rasmussen 2011]. These methods target at modeling and optimizing the policy directly. The policy is usually modeled with a parameterized function with respect to $\pi_\theta(a|s)$. The value of the reward (objective) function depends on this policy and then various algorithms can be applied to optimize θ for the best reward. There are a series of algorithms, which use the PG to search in the policy space, and at the same time estimate a value function. The important class of these methods are Actor-Critic (AC) and its variation [Konda and Tsitsiklis 1999; Peters et al. 2005; Peters and Schaal 2008; Bhatnagar et al. 2007; Bhatnagar et al. 2009]. These are two-time-scale algorithms where the critic uses Temporal-Difference (TD) learning with a linear approximation architecture and the actor is updated in an approximate gradient direction based on information provided by the critic.

3. REINFORCEMENT LEARNING FOR SEARCH

Search aims to find and rank a set of objects (e.g. documents, records) based on a user query. In this section, we review RL applications in key topics of search.

3.1 Query understanding

Query understanding is the primary task for the search engine to understand users' information needs. It can be potentially useful for improving general search relevance, user experience, and helping users to accomplish tasks [Croft et al. 2010]. In [Nogueira and Cho 2017], RL has been leveraged to solve the query reformulation task: a query reformulation framework is proposed based on a neural network, which rewrites a query to maximize the number of relevant documents returned. In the proposed framework, a search engine is treated as a black box that an agent learns to use in order to retrieve more relevant items, which opens the possibility of training an agent to use a search engine for a task other than the one it was originally intended for. Additionally, the upper-bound performance of an RL-based model is estimated in a given environment. In [Nogueira et al. 2018], a multi-agent based method is introduced to efficiently learn diverse query reformulation. It is argued that it is easier to train multiple sub-agents than a single generalist one since each sub-agent only needs to learn a policy that performs well for a subset of examples. In the proposed framework, an agent consists of multiple specialized sub-agents and a meta-agent that learns to aggregate the answers from sub-agents to produce a final answer. Thus, the method makes learning faster with parallelism.

3.2 Ranking

Relevance Ranking is the core problem of information retrieval [Yin et al. 2016] and learning to rank (LTR) is the key technology in relevance ranking. In LTR, the approaches to directly optimize the ranking evaluation measures are representative and have been proved to be effective[Yue et al. 2007; Xu and Li 2007; Xu et al. 2008]. These methods usually only optimize the evaluation measure calculated at a predefined ranking position, e.g. NDCG at rank K in [Xu and Li 2007]. The information carried by the documents after the rank K are neglected. To solve such problem, in [Zeng et al. 2017], a LTR model, MDPRank, is proposed based on Markov decision process, which has the ability of leveraging the measures calculated at all of the ranking positions. The reward function is defined based upon the IR evaluation measures and the model parameters can be learned through maximizing the accumulated rewards to all of the decisions. Implicit relevance feedback refers to an interactive process between search engine and user, and has been proven to be very effective for improving retrieval accuracy [Lv and Zhai 2009]. Both Bandits and MDPs can model such an interactive process naturally [Vorobev et al. 2015; Katariya et al. 2016; Katariya et al. 2017]. In [Kveton et al. 2015], cascading bandits are introduced to identify the most attractive items, and the goal of the agent is to maximize its total reward with respect to the list of the most attractive items. Through maintaining state transition, MDP is able to model the user state in the interaction with search engine. In [Zeng et al. 2018], the interactive process is formulated as an MDP and the Recurrent Neural Network is applied to process the feedback.

Beyond relevance ranking, another important goal is to provide search results that cover a wide range of topics for a query, i.e., search result diversification [Santos et al. 2015; Xu et al. 2017]. Typical methods formulate the problem of constructing a diverse ranking as a process of greedy sequential document selection. To select an optimal document for a position, it is critical for a diverse ranking model to capture the utility of information users have perceived from the preceding documents. To explicitly model the utility perceived by the users, the construction of a diverse ranking is formalized as a process of sequential decision making and the process is modeled as a continuous state Markov decision process, referred to as MDP-DIV [Xia et al. 2017]. The ranking of M documents is formalized as a sequence of M decisions and each action corresponds to selecting one document from the candidate set. In the parameter training phase, the policy gradient algorithm of REINFORCE is adopted and the expected long-term discounted rewards in terms of the diversity evaluation measure is maximized. More works for diversity ranking see [Feng et al. 2018; Kapoor et al. 2018]

3.3 Whole-page optimization

To improve user experiences, modern search engines aggregate versatile results from different verticals – web-pages, news, images, video, shopping, knowledge cards, local maps, etc. Page presentation is broadly defined as the strategy to present a set of items on search result page (SERP), which is much more expressive than a ranked list. Finding proper presentation for a gallery of heterogeneous results is critical for modern search engines.

One approach to efficiently learning to optimize a large decision space is fractional factorial design. However, the method could cause combinatorial explosion problem with a large search space. In [Hill et al. 2017], bandit formulation is applied to explore the layout space efficiently and hill-climbing is used to select optimal content in real-time. The model avoids a combinatorial explosion in model complexity by only considering pairwise interactions between page components. This approach is a greedy alternating optimization strategy that can run online in real-time. In [Wang et al. 2016; Wang et al. 2018], a framework is proposed to learn the optimal page presentation to render heterogeneous results onto SERP. It leveraged the MDP setting and the agent is designed as the algorithm that determines the presentation of page content on a SERP for each incoming search query. To solve the critical efficiency problem, it proposed a policy-based learning method which can rapidly choose actions from the high-dimensional space.

3.4 Session search

The task oriented search includes a series of search iterations triggered by the query reformulations within a session. Markov chain in session search is observed: users judgment of search results in the prior iteration will influence users behaviors in the next search iteration. Session search is modeled as a dual-agent stochastic game based on Partially Observable Markov Decision Process (POMDP) in [Luo et al. 2014]. They mathematically model dynamics in session search as a cooperative game between the user and the search engine, while user and the search engine work together in order to jointly maximize the long-term cumulative rewards. Log-based document re-ranking is a special type of session search that re-ranks documents based on the historical search logs which includes the target user's personalized query log and other users' search activities. The re-ranking aims to offer a better order of the initial retrieved documents [Zhang et al. 2014]. Nowadays, deep reinforcement learning technology has been applied in the E-Commerce search engine [Hu et al. 2018; Feng et al. 2018]. For better utilizing the correlation between different ranking steps, RL is used to learn an optimal ranking policy which maximizes the expected accumulative rewards in a search session [Hu et al. 2018]. It formally defined the multi-step ranking problem in the search session as MDP, denoted as SSMDP, and proposed a novel policy gradient algorithm for learning an optimal ranking policy, which is able to deal with the problem of high reward variance and unbalanced reward distribution. In [Feng et al. 2018], multi-scenario ranking is formulated as a fully cooperative, partially observable, multi-agent sequential decision problem, denoted as MA-RDPG. MA-RDPG has a communication component for passing message, several private agents for making action for ranking, and a centralized critic for evaluating the overall performance of the co-working agents. Agents collaborate with each other by sharing a global action-value function and passing messages that encode historical information across scenarios.

4. REINFORCEMENT LEARNING FOR RECOMMENDATION

Recommender systems target to capture users' preferences according to their feedback (or behaviors, e.g. rating and review) and suggest items that match their preferences. In this section, we briefly review how RL is adapted in several key tasks in recommendations.

4.1 Exploitation/Exploration Dilemma

Traditional recommender systems suffer from the exploitation-exploration dilemma, where exploitation is to recommend items that are predicted to best match users' preferences, while exploration is to recommend items randomly to collect more users' feedback. The contextual bandit models an agent that attempts to balance the competing exploitation and exploration tasks in order to maximize the accumulated long-term reward over a considered period. The traditional strategies to balance exploitation and exploration in bandit setting are ϵ-greedy [Watkins 1989], EXP3 [Auer et al. 2002], and UCB1 [Auer et al. 2002]. In the news feeds scenario, the exploration/exploitation problem of personalized news recommendation is modeled as a contextual bandit problem [Li et al. 2010], and a learning algorithm LinUCB is proposed to select articles sequentially for specific users based on the users' and articles' contextual information, in order to maximize the total user clicks.

4.2 Temporal Dynamics

Most existing recommender systems such as collaborative filtering, content-based and learning-to-rank have been extensively studied with the stationary environment (reward) assumption, where user's preference is assumed to be static. However, this assumption is usually not true in reality since users preferences are dynamic, thus the reward distributions usually change over time. In bandit setting, it usually introduces a variable reward function to delineate the dynamic nature of the environment. For instance, the particle learning based dynamical context drift model is proposed to model the changing of reward mapping function in multi-armed bandit problem, where the drift of the reward mapping function is learned as a group of random walk particles, and good fitted particles are dynamically chose to describe the mapping function [Zeng et al. 2016]. A contextual bandit algorithm is presented to detect the changes of environment according to the reward estimation confidence, and updates the arm selection policy accordingly [Wu et al. 2018]. The change-detection based framework under the piecewise-stationary reward assumption for the multi-armed bandit problem is proposed in [Liu et al. 2018], where upper confidence bound (UCB) policies is used to detect change points actively and restart the UCB indices. Another solution for capturing user's dynamic preference is to introduce the MDP setting [Chen et al. 2018; Liu et al. 2018; Zhao et al. 2018]. Under the MDP setting, *state* is introduced to represent user's preference and *state transition* captures the dynamic nature of user's preference over time. In [Zhao et al. 2018], a user's dynamic preference (agent's state) is learned from his/her browsing history. Each time the recommender system suggests an items to a user, the user will browse this item and provide feedback (skip, click or purchase), which reveals user's satisficaiton of the recommended item. According to the feedback, the recommender system will update its state to represent user's new preferences [Zhao et al. 2018].

4.3 Long Term User Engagement

User engagement in recommendation is the assessment of user's desirable (even essential) responses to the items (products, services, or information) suggested by the recommender systems [Lalmas et al. 2014]. User engagement can be measured not only in terms of immediate response (e.g. clicks and rating of the recommended items), but more importantly in terms of long-term response (e.g. user repetitively purchases) [Schopfer and Keller]. In [Wu et al. 2017], the problem of long-term user engagement optimization is formulated as a sequential decision making problem. In each iteration, the agent needs to estimate the risk of losing a user based on the user's dynamic response to past recommendations. Then, a bandit based method [Wu et al. 2017] is introduced to balance the immediate user click and the expected future clicks when the user revisits the recommender system. In the news feeds scenario [Zheng et al. 2018], to incorporate more user feedback information, the long-term user response (i.e., how frequent user returns) is considered as a supplement to user's immediate click behaviors, and a Deep Q-Learning based framework is proposed to optimize the news recommendation strategies.

4.4 Page-Wise Recommendation

In practical recommender systems, each time users are typically recommended a page of items. In this setting, the recommender systems need to jointly (1) select a set of complementary and diverse items from a larger candidate item set and (2) form an item display (layout configuration) strategy to place the items in a 2-D web page that can lead to maximal reward. Given the massive number of items, the action space is extremely large if we treat each whole page recommendation as one action. To mitigate the issue of the large action space, a Deep Deterministic Policy Gradient algorithm is proposed [Dulac-Arnold et al. 2015] where the Actor generates a deterministic optimal action according to current state, and the Critic outputs the Q-value of this state-action pair. DDPG reduces the computational cost of conventional value-based reinforcement learning methods, thus it is a fitting choice for the whole page recommendation setting [Cai et al. 2018a; Cai et al. 2018b]. Several approaches are presented recently to enhance the efficiency [Choi et al. 2018; Chen et al. 2018]. In [Zhao et al. 2018; Zhao et al. 2018], CNN techniques are introduced to capture the item display patterns and users' feedback of each item in the page. To represent each item, item-embedding, category-embedding and feedback embedding are leveraged, which can help to generate complementary and diverse recommendations and capture user's interests within the pages. Bandit techniques are also leveraged for whole-page Recommendations [Wang et al. 2017; Lacerda 2017]. For instance, the whole page recommendation task is considered as a combinatorial semi-bandit problem, where the system recommends S actions from a candidate set of K actions, and displays the selected items in S (out of M) positions [Wang et al. 2017].

5. REINFORCEMENT LEARNING FOR ONLINE ADVERTISING

The goal of online advertising is to assign the right advertisements to the right users so as to maximize the revenue, click-through rate (CTR) or return on investment (ROI) of the advertising campaign. The two main marketing strategy in online advertising are guaranteed delivery (GD) and real-time bidding (RTB).

5.1 Guaranteed delivery

In guaranteed delivery, advertisements that share a single idea and theme are grouped into campaigns, and are charged on a pay-per-campaign basis for the pre-specified number of deliveries (click or impressions) [Salomatin et al. 2012]. Most popular GD (Guaranteed

Delivery) solutions are based on offline optimization algorithms, and then adjusted for online setup. However, deriving the optimal strategy to allocate impressions is challenging, especially when the environment is unstable in real-world application. In [Wu et al. 2018], a multi-agent reinforcement learning (MARL) approach is proposed to derive cooperative policies for the publisher to maximize its target in an unstable environment. They formulated the impression allocation problem as an auction problem where each contract can submit virtual bids for individual impressions. With this formulation, they derived the optimal impression allocation strategy by solving the optimal bidding functions for contracts.

5.2 Real-time bidding

RTB allows an advertiser to submit a bid for each individual impression in a very short time frame. Ad selection task is typically modeled as multi-armed bandit (MAB) problem with the setting that samples from each arm are iid, feedback are immediate and rewards are stationary [Yang and Lu 2016; Nuara et al. 2018; Gasparini et al. 2018; Tang et al. 2013; Xu et al. 2013; Yuan et al. 2013; Schwartz et al. 2017]. The payoff functions of an MAB are allowed to evolve, but they are assumed to evolve slowly over time. On the other hand, display ads created while others are removed regularly in an advertising campaign circulation. The problem of multi-armed bandits with budget constraints and variable costs is studied in [Ding et al.]. In this case, pulling the arms of bandit will get random rewards with random costs, and the algorithm aims to maximize the long-term reward by pulling arms with a constrained budget. This setting can model Internet advertising in a more precise way than previous works where pulling an arm is costless or has a fixed cost.

Under the MAB setting, the bid decision is considered as a static optimization problem of either treating the value of each impression independently or setting a bid price to each segment of ad volume. However, the bidding for a given ad campaign would repeatedly happen during its life span before the budget running out. Thus, the MDP setting have also been studied [Cai et al. 2017; Tang 2017; Wang et al. 2018; Zhao et al. 2018; Rohde et al. 2018; Wu et al. 2018; Jin et al. 2018]. A model-based reinforcement learning framework is proposed to learn bid strategies in RTB advertising [Cai et al. 2017], where neural network is used to approximate the state value, which can better deal with the scalability problem of large auction volume and limited campaign budget. A model-free deep reinforcement learning method is proposed to solve the bidding problem with constrained budget [Wu et al. 2018]: the problem is modeled as a λ-control problem, and RewardNet is designed for generating rewards to solve reward design trap, instead of using the immediate reward. A multi-agent bidding model is presented, which takes the other advertisers' bidding in the system into consideration, and a clustering approach is introduced to solve the large number of advertisers challenge [Jin et al. 2018].

6. CONCLUSION AND FUTURE DIRECTIONS

In this article, we present an overview of information seeking from reinforcement learning perspective. We first introduce mathematical foundations of RL based information seeking approaches. Then we review state-of-the-art algorithms of three representative information seeking mechanisms – search, recommendation and advertising. Next, we here discuss some interesting research directions on reinforcement learning that can bring the information seeking research into a new frontier.

First, most of existing works train a policy within one scenario, while overlooking users' behaviors (preference) in other scenarios [Feng et al. 2018]. This will result in a suboptimal policy, which calls for collaborative RL frameworks that consider search, recommendation and advertising scenarios simultaneously. Second, the type of reward function varies among different computational tasks. More sophisticated reward functions should be designed to achieve more goals of information seeking, such as increasing the supervising degree of recommendations. Third, more types of user-agent interactions could be incorporated into RL frameworks, such as adding items into shopping cart, users' repeat purchase behavior, users' dwelling time in the system, and user's chatting with customer service representatives or agent of AI dialog system. Fourth, testing a new algorithm is expensive since it needs lots of engineering efforts to deploy the algorithm in the practical system, and it also may have negative impacts on user experience if the algorithm is not mature. Thus online environment simulator or offline evaluation method based on historical logs are necessary to pre-train and evaluate new algorithms before launching them online. Finally, there is an increasing demand of an open online reinforcement learning environment for information seeking, which can advance the RL and information seeking communities and achieve better consistency between offline and online performance.

Acknowledgements

Xiangyu Zhao and Jiliang Tang are supported by the National Science Foundation (NSF) under grant numbers IIS-1714741, IIS-1715940 and CNS-1815636, and a grant from Criteo Faculty Research Award.

REFERENCES

ÅSTRÖM, K. J. 1965. Optimal control of markov processes with incomplete state information. *Journal of Mathematical Analysis and Applications 10,* 1, 174–205.

AUER, P., CESA-BIANCHI, N., AND FISCHER, P. 2002. Finite-time analysis of the multiarmed bandit problem. *Machine Learning 47,* 2-3, 235–256.

AUER, P., CESA-BIANCHI, N., FREUND, Y., AND SCHAPIRE, R. E. 2002. The nonstochastic multiarmed bandit problem. *SIAM J. Comput. 32,* 1, 48–77.

BAXTER, J. AND BARTLETT, P. L. 2001. Infinite-horizon policy-gradient estimation. *J. Artif. Intell. Res. 15*, 319–350.

BELLMAN, R. 2013. *Dynamic programming*. Courier Corporation.

BHATNAGAR, S., SUTTON, R. S., GHAVAMZADEH, M., AND LEE, M. 2007. Incremental natural actor-critic algorithms. In *NIPS '07.*

BHATNAGAR, S., SUTTON, R. S., GHAVAMZADEH, M., AND LEE, M. 2009. Natural actor-critic algorithms. *Automatica 45,* 11, 2471–2482.

BOWLING, M. H. AND VELOSO, M. M. 2002. Multiagent learning using a variable learning rate. *Artif. Intell. 136,* 2, 215–250.

BUSONIU, L., BABUSKA, R., DE SCHUTTER, B., AND ERNST, D. 2010. *Reinforcement learning and dynamic programming using function approximators*. CRC press.

BUSONIU, L., BABUSKA, R., AND SCHUTTER, B. D. 2008. A comprehensive survey of multiagent reinforcement learning. *IEEE Trans. Systems, Man, and Cybernetics, Part C 38,* 2, 156–172.

CAI, H., REN, K., ZHANG, W., MALIALIS, K., WANG, J., YU, Y., AND GUO, D. 2017. Real-time bidding by reinforcement learning in display advertising. In *WSDM '17.*

CAI, Q., FILOS-RATSIKAS, A., TANG, P., AND ZHANG, Y. 2018a. Reinforcement mechanism design for e-commerce. In *WWW '18*.

CAI, Q., FILOS-RATSIKAS, A., TANG, P., AND ZHANG, Y. 2018b. Reinforcement mechanism design for fraudulent behaviour in e-commerce. In *AAAI '18*.

CHANG, C., KAYED, M., GIRGIS, M. R., AND SHAALAN, K. F. 2006. A survey of web information extraction systems. *IEEE Trans. Knowl. Data Eng. 18,* 10, 1411–1428.

CHEN, H., DAI, X., CAI, H., ZHANG, W., WANG, X., TANG, R., ZHANG, Y., AND YU, Y. 2018. Large-scale interactive recommendation with tree-structured policy gradient. *CoRR abs/1811.05869*.

CHEN, S., YU, Y., DA, Q., TAN, J., HUANG, H., AND TANG, H. 2018. Stabilizing reinforcement learning in dynamic environment with application to online recommendation. In *SIGKDD '18*.

CHOI, S., HA, H., HWANG, U., KIM, C., HA, J., AND YOON, S. 2018. Reinforcement learning based recommender system using biclustering technique. *CoRR abs/1801.05532*.

CROFT, W. B., BENDERSKY, M., LI, H., AND XU, G. 2010. Query representation and understanding workshop. *SIGIR Forum 44,* 2, 48–53.

DEISENROTH, M. P. AND RASMUSSEN, C. E. 2011. PILCO: A model-based and data-efficient approach to policy search. In *ICML '11*.

DING, W., QIN, T., ZHANG, X., AND LIU, T. Multi-armed bandit with budget constraint and variable costs. In *AAAI '13*.

DULAC-ARNOLD, G., EVANS, R., VAN HASSELT, H., SUNEHAG, P., LILLICRAP, T., HUNT, J., MANN, T., WEBER, T., DEGRIS, T., AND COPPIN, B. 2015. Deep reinforcement learning in large discrete action spaces. *arXiv preprint arXiv:1512.07679*.

FENG, J., LI, H., HUANG, M., LIU, S., OU, W., WANG, Z., AND ZHU, X. 2018. Learning to collaborate: Multi-scenario ranking via multi-agent reinforcement learning. In *WWW '18*.

FENG, Y., XU, J., LAN, Y., GUO, J., ZENG, W., AND CHENG, X. 2018. From greedy selection to exploratory decision-making: Diverse ranking with policy-value networks. In *SIGIR '18*.

GARCIA-MOLINA, H., KOUTRIKA, G., AND PARAMESWARAN, A. G. 2011. Information seeking: convergence of search, recommendations, and advertising. *Commun. ACM 54,* 11, 121–130.

GASPARINI, M., NUARA, A., TROVÒ, F., GATTI, N., AND RESTELLI, M. 2018. Targeting optimization for internet advertising by learning from logged bandit feedback. In *IJCNN '18*.

HILL, D. N., NASSIF, H., LIU, Y., IYER, A., AND VISHWANATHAN, S. V. N. 2017. An efficient bandit algorithm for realtime multivariate optimization. In *SIGKDD '17*.

HU, Y., DA, Q., ZENG, A., YU, Y., AND XU, Y. 2018. Reinforcement learning to rank in e-commerce search engine: Formalization, analysis, and application. In *SIGKDD '18*.

JIN, J., SONG, C., LI, H., GAI, K., WANG, J., AND ZHANG, W. 2018. Real-time bidding with multi-agent reinforcement learning in display advertising. In *CIKM '18*.

KAELBLING, L. P., LITTMAN, M. L., AND CASSANDRA, A. R. 1998. Planning and acting in partially observable stochastic domains. *Artif. Intell. 101,* 1-2, 99–134.

KAELBLING, L. P., LITTMAN, M. L., AND MOORE, A. W. 1996. Reinforcement learning: A survey. *J. Artif. Intell. Res. 4*, 237–285.

KAKADE, S. 2001. A natural policy gradient. In *NIPS '01*.

KAPOOR, S., KESWANI, V., VISHNOI, N. K., AND CELIS, L. E. 2018. Balanced news using constrained bandit-based personalization. In *IJCAI '18*.

KATARIYA, S., KVETON, B., SZEPESVÁRI, C., VERNADE, C., AND WEN, Z. 2017. Bernoulli rank-1 bandits for click feedback. In *IJCAI '17*.

KATARIYA, S., KVETON, B., SZEPESVÁRI, C., AND WEN, Z. 2016. DCM bandits: Learning to rank with multiple clicks. In *ICML '16*.

KONDA, V. R. AND TSITSIKLIS, J. N. 1999. Actor-critic algorithms. In *NIPS '99*.

KRÖSE, B. J. A. 1995. Learning from delayed rewards. *Robotics and Autonomous Systems 15,* 4, 233–235.

KVETON, B., SZEPESVÁRI, C., WEN, Z., AND ASHKAN, A. 2015. Cascading bandits: Learning to rank in the cascade model. In *ICML '15*.

LACERDA, A. 2017. Multi-objective ranked bandits for recommender systems. *Neurocomputing 246*, 12–24.

LAGOUDAKIS, M. G. AND PARR, R. 2003. Least-squares policy iteration. *Journal of Machine Learning Research 4*, 1107–1149.

LALMAS, M., O'BRIEN, H., AND YOM-TOV, E. 2014. *Measuring User Engagement.* Synthesis Lectures on Information Concepts, Retrieval, and Services. Morgan & Claypool Publishers.

LI, L., CHU, W., LANGFORD, J., AND SCHAPIRE, R. E. 2010. A contextual-bandit approach to personalized news article recommendation. In *WWW '10*.

LIU, F., LEE, J., AND SHROFF, N. B. 2018. A change-detection based framework for piecewise-stationary multi-armed bandit problem. In *AAAI '18*.

LIU, F., TANG, R., LI, X., YE, Y., CHEN, H., GUO, H., AND ZHANG, Y. 2018. Deep reinforcement learning based recommendation with explicit user-item interactions modeling. *CoRR abs/1810.12027*.

LU, T., PÁL, D., AND PAL, M. 2010. Contextual multi-armed bandits. In *AISTATS '10*.

LUO, J., ZHANG, S., AND YANG, H. 2014. Win-win search: dual-agent stochastic game in session search. In *SIGIR '14*.

LV, Y. AND ZHAI, C. 2009. Adaptive relevance feedback in information retrieval. In *CIKM '09*.

MNIH, V., KAVUKCUOGLU, K., SILVER, D., RUSU, A. A., VENESS, J., BELLEMARE, M. G., GRAVES, A., RIEDMILLER, M. A., FIDJELAND, A., OSTROVSKI, G., PETERSEN, S., BEATTIE, C., SADIK, A., ANTONOGLOU, I., KING, H., KUMARAN, D., WIERSTRA, D., LEGG, S., AND HASSABIS, D. 2015. Human-level control through deep reinforcement learning. *Nature 518,* 7540, 529–533.

MOORE, A. W. AND ATKESON, C. G. 1993. Prioritized sweeping: Reinforcement learning with less data and less time. *Machine Learning 13*, 103–130.

NOGUEIRA, R., BULIAN, J., AND CIARAMITA, M. 2018. Learning to coordinate multiple reinforcement learning agents for diverse query reformulation. *CoRR abs/1809.10658*.

NOGUEIRA, R. AND CHO, K. 2017. Task-oriented query reformulation with reinforcement learning. In *EMNLP '17*.

NUARA, A., TROVÒ, F., GATTI, N., AND RESTELLI, M. 2018. A combinatorial-bandit algorithm for the online joint bid/budget optimization of pay-per-click advertising campaigns. In *AAAI '18*.

PETERS, J. AND SCHAAL, S. 2008. Natural actor-critic. *Neurocomputing 71,* 7-9, 1180–1190.

PETERS, J., VIJAYAKUMAR, S., AND SCHAAL, S. 2005. Natural actor-critic. In *ECML '05*.

ROHDE, D., BONNER, S., DUNLOP, T., VASILE, F., AND KARATZOGLOU, A. 2018. Recogym: A reinforcement learning environment for the problem of product recommendation in online advertising. *CoRR abs/1808.00720*.

RUMMERY, G. A. AND NIRANJAN, M. 1994. *On-line Q-learning using connectionist systems*. Vol. 37. University of Cambridge, Department of Engineering Cambridge, England.

SALOMATIN, K., LIU, T., AND YANG, Y. 2012. A unified optimization framework for auction and guaranteed delivery in online advertising. In *CIKM '12*.

SANTOS, R. L. T., MACDONALD, C., AND OUNIS, I. 2015. Search result diversification. *Foundations and Trends in Information Retrieval 9,* 1, 1–90.

SCHOPFER, S. AND KELLER, T. Long term recommender benchmarking for mobile shopping list applications using markov chains. In *RecSys '14*.

SCHWARTZ, E. M., BRADLOW, E. T., AND FADER, P. S. 2017. Customer acquisition via display advertising using multi-armed bandit experiments. *Marketing Science 36,* 4, 500–522.

SHANI, G., HECKERMAN, D., AND BRAFMAN, R. I. 2005. An mdp-based recommender system. *Journal of Machine Learning Research 6*, 1265–1295.

SHOHAM, Y., POWERS, R., AND GRENAGER, T. 2003. Multi-agent reinforcement learning: a critical survey. Tech. rep., Technical report, Stanford University.

SMALLWOOD, R. D. AND SONDIK, E. J. 1973. The optimal control of partially observable markov processes over a finite horizon. *Operations Research 21,* 5, 1071–1088.

SONDIK, E. J. 1978. The optimal control of partially observable markov processes over the infinite horizon: Discounted costs. *Operations Research 26,* 2, 282–304.

SUTTON, R. S. 1991. Dyna, an integrated architecture for learning, planning, and reacting. *SIGART Bulletin 2,* 4, 160–163.

SUTTON, R. S. AND BARTO, A. G. 1998. *Introduction to reinforcement learning*. Vol. 135. MIT press Cambridge.

TANG, L., ROSALES, R., SINGH, A., AND AGARWAL, D. 2013. Automatic ad format selection via contextual bandits. In *CIKM '13*.

TANG, P. 2017. Reinforcement mechanism design. In *IJCAI '17*.

VARAIYA, P. AND WALRAND, J. C. 1983. Multi-armed bandit problems and resource sharing systems. In *Computer Performance and Reliability, Proceedings of the International Workshop, Pisa, Italy, September 26-30, 1983*. 181–196.

VOROBEV, A., LEFORTIER, D., GUSEV, G., AND SERDYUKOV, P. 2015. Gathering additional feedback on search results by multi-armed bandits with respect to production ranking. In *WWW '15*.

WANG, W., JIN, J., HAO, J., CHEN, C., YU, C., ZHANG, W., WANG, J., WANG, Y., LI, H., XU, J., AND GAI, K. 2018. Learning to advertise with adaptive exposure via constrained two-level reinforcement learning. *CoRR abs/1809.03149*.

WANG, Y., OUYANG, H., WANG, C., CHEN, J., ASAMOV, T., AND CHANG, Y. 2017. Efficient ordered combinatorial semi-bandits for whole-page recommendation. In *AAAI '17*.

WANG, Y., YIN, D., JIE, L., WANG, P., YAMADA, M., CHANG, Y., AND MEI, Q. 2016. Beyond ranking: Optimizing whole-page presentation. In *WSDM '16*.

WANG, Y., YIN, D., JIE, L., WANG, P., YAMADA, M., CHANG, Y., AND MEI, Q. 2018. Optimizing whole-page presentation for web search. *TWEB 12,* 3, 19:1–19:25.

WATKINS, C. J. C. H. 1989. Learning from delayed rewards. Ph.D. thesis, King's College, Cambridge.

WILLIAMS, R. J. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning 8*, 229–256.

WU, D., CHEN, C., YANG, X., CHEN, X., TAN, Q., XU, J., AND GAI, K. 2018. A multi-agent reinforcement learning method for impression allocation in online display advertising. *CoRR abs/1809.03152*.

WU, D., CHEN, X., YANG, X., WANG, H., TAN, Q., ZHANG, X., XU, J., AND GAI, K. 2018. Budget constrained bidding by model-free reinforcement learning in display advertising. In *CIKM '18*.

WU, Q., IYER, N., AND WANG, H. 2018. Learning contextual bandits in a non-stationary environment. In *SIGIR '18*.

WU, Q., WANG, H., HONG, L., AND SHI, Y. 2017. Returning is believing: Optimizing long-term user engagement in recommender systems. In *CIKM '17*.

XIA, L., XU, J., LAN, Y., GUO, J., ZENG, W., AND CHENG, X. 2017. Adapting markov decision process for search result diversification. In *SIGIR '17*.

XU, J. AND LI, H. 2007. Adarank: a boosting algorithm for information retrieval. In *SIGIR '07*.

XU, J., LIU, T., LU, M., LI, H., AND MA, W. 2008. Directly optimizing evaluation measures in learning to rank. In *SIGIR '08*.

XU, J., XIA, L., LAN, Y., GUO, J., AND CHENG, X. 2017. Directly optimize diversity evaluation measures: A new approach to search result diversification. *ACM TIST 8,* 3, 41:1–41:26.

XU, M., QIN, T., AND LIU, T. 2013. Estimation bias in multi-armed bandit algorithms for search advertising. In *NIPS '13*.

YANG, H. AND LU, Q. 2016. Dynamic contextual multi arm bandits in display advertisement. In *ICDM '16*.

YIN, D., HU, Y., TANG, J., JR., T. D., ZHOU, M., OUYANG, H., CHEN, J., KANG, C., DENG, H., NOBATA, C., LANGLOIS, J., AND CHANG, Y. 2016. Ranking relevance in yahoo search. In *SIGKDD '16*.

YUAN, S., WANG, J., AND VAN DER MEER, M. 2013. Adaptive keywords extraction with contextual bandits for advertising on parked domains. *CoRR abs/1307.3573*.

YUE, Y., FINLEY, T., RADLINSKI, F., AND JOACHIMS, T. 2007. A support vector method for optimizing average precision. In *SIGIR '07*.

ZENG, C., WANG, Q., MOKHTARI, S., AND LI, T. 2016. Online context-aware recommendation with time varying multi-armed bandit. In *SIGKDD '16*.

ZENG, W., XU, J., LAN, Y., GUO, J., AND CHENG, X. 2017. Reinforcement learning to rank with markov decision process. In *SIGIR '17*.

ZENG, W., XU, J., LAN, Y., GUO, J., AND CHENG, X. 2018. Multi page search with reinforcement learning to rank. In *ICTIR '18*.

ZHANG, S., LUO, J., AND YANG, H. 2014. A POMDP model for content-free document re-ranking. In *SIGIR '14*.

ZHAO, J., QIU, G., GUAN, Z., ZHAO, W., AND HE, X. 2018. Deep reinforcement learning for sponsored search real-time bidding. In *SIGKDD '18*.

ZHAO, X., XIA, L., ZHANG, L., DING, Z., YIN, D., AND TANG, J. 2018. Deep reinforcement learning for page-wise recommendations. In *RecSys '18*.

ZHAO, X., ZHANG, L., DING, Z., XIA, L., TANG, J., AND YIN, D. 2018. Recommendations with negative feedback via pairwise deep reinforcement learning. In *SIGKDD '18*.

ZHAO, X., ZHANG, L., DING, Z., YIN, D., ZHAO, Y., AND TANG, J. 2018. Deep reinforcement learning for list-wise recommendations. *CoRR abs/1801.00209*.

ZHENG, G., ZHANG, F., ZHENG, Z., XIANG, Y., YUAN, N. J., XIE, X., AND LI, Z. 2018. DRN: A deep reinforcement learning framework for news recommendation. In *WWW '18*.

Xiangyu Zhao is a Ph.D. student of computer science and engineering at Michigan State University (MSU). His supervisor is Dr. Jiliang Tang. Before joining MSU, he completed his MS(2017) at USTC and BS(2014) at UESTC. He is the student member of IEEE, SIGIR, and SIAM. His current research interests include data mining and machine learning, especially (1) Reinforcement Learning for E-commerce; (2) Urban Computing and Spatio-Temporal Data Analysis. After joining MSU, he has published his work in top journals (e.g. SIGKDD Explorations) and conferences (e.g., KDD, ICDM, CIKM, RecSys). He was the recipients of the RecSys18, KDD18, SDM18, and CIKM17 Student Travel Award.

Long Xia is a research scientist in Data Science Lab at JD.com. He is now mainly responsible for applying advanced technology to the E-commerce recommender system in JD.com. Before that, he received his Ph.D. in Computer Science from Institute of Computing Technology, Chinese Academy of Sciences. His research interests include data mining, applied machine learning, information retrieval, and recommender system. He has published his research in top journals and conferences, e.g. TIST, SIGIR, KDD, RecSys.

Jiliang Tang is an assistant professor in the computer science and engineering department at Michigan State University. Before that, he was a research scientist in Yahoo Research and got his PhD from Arizona State University in 2015. He has broad interests in social computing, data mining and machine learning. He was the recipients of the Best Paper Award in ASONAM 2018, the Best Student Paper Award in WSDM2018, the Best Paper Award in KDD2016, the runner up of the Best KDD Dissertation Award in 2015, Dean's Dissertation Award and the best paper shortlist of WSDM2013. He is now associate editors of ACM TKDD, ICWSM and Neurocomputing. He has published his research in highly ranked journals and top conference proceedings, which received thousands of citations and extensive media coverage.

Dawei Yin is Senior Director at JD.com, leading the science efforts of recommendation, search, metrics and knowledge graph. Prior to joining JD.com, he was Senior Research Manager at Yahoo Labs, leading relevance science team and in charge of Core Search Relevance of Yahoo Search. He obtained Ph.D. (2013), M.S. (2010) from Lehigh University and B.S. (2006) from Shandong University. His research interests include data mining, applied machine learning, information retrieval and recommender system. He published more than 60 research papers in premium conferences and journals, and won WSDM2016 best paper award, KDD2016 best paper award, and WSDM2018 best student paper award.