Who to Learn from: A Preference-based Method for Social Reinforcement Learning

Banafsheh Karimian, Erfan Mirzaei*, Amir Hossein Mesbah*, Reshad Hosseini, Seyed Pooya Shariatpanahi, Majid Nili Ahmadabadi

***Abstract*—Reinforcement Learning (RL) is mainly inspired by studies on animal and human learning. However, RL methods suffer higher regret in comparison to natural learners in real-world tasks. This is partly due to the lack of social learning in RL agents. We propose a social learning method for improving the performance of RL agents for the multi-armed bandit setting. The social agent observes other agents' decisions, while their rewards are private. The agent uses a preference-based method, similar to the policy gradient learning method, to find if there are any agents in the heterogeneous society worth learning from their policies to improve their performance. The heterogeneity is the result of diversity in learning algorithms, utility functions, and expertise. We compare our method with state-of-the-art studies and demonstrate that it results in higher performance in most scenarios. We also show that performance improvement increases with the problem complexity and is inversely correlated with the population of unrelated agents.**

***Index Terms*—Reinforcement Learning; Social Learning; Multi-armed Bandit**

I. Introduction

SOCIAL learning, in which interactions and observations are used to learn from other agents in society, helps humans and animals learn complex behaviors faster and more efficiently. Both humans and animals rely on social learning when individual learning is difficult for them [1]–[3]. It is noted that humans owe their evolution to the ability to learn from their society [4]–[11]. The complexity of human skills could not have been achieved if they had to rely only on individual learning in order to identify solutions for everyday decision-making problems. An isolated learner must invest sufficient energy and time to explore the available options and may thus encounter unexpected negative outcomes, making individual learning slow and costly [12].

Reinforcement Learning (RL) is one of the most popular machine learning methods [13] and has evolved as one of the learning tools alongside social learning in diverse societies. However, the fact that RL is used by socially situated creatures in nature is largely ignored. Therefore, RL as a standalone learning method, even though is highly effective and is improved through various methods such as intrinsic motivations [14]–[19], transfer learning [20] or sub-spacing [21], has some drawbacks. For instance, it suffers from exponentially increased regret in high dimensional state space, has difficulties in adapting online to new environments, and is slow in producing near-optimal results [1], [22]. Considering the fact that RL agents could address the mentioned complications through learning from social cues [23], here we aim to use social learning to import social aspect to individual RL in multi-armed bandit settings.

In this paper, we investigate the importance of using social cues in speeding the k-armed bandit problem learning. We first introduce a preference-based method for the agent to evaluate the level of expertise of other agents interacting with the environment and its relevance to those agents, in addition to their information's importance for speeding the agent's learning. We also show that the social agent can evaluate other agents properly even in a populated society and if an agent becomes more or less informative for the social agent, the social agent is able to detect and change its preference to that agent. After continuously evaluating the society, if the social agent finds another agent informative, it will learn from that agent by selecting it for an iteration in order to find the optimal solution faster. To prevent the agent from becoming totally reliant on the other agents, the agent calculates its preference for its own internal learner and decides if other agents are more informative to be selected or if its internal learner has mastered the skills and needs no effort in selecting others. We analyze the influence of problem complexity, society population size, and the quality of information the society has for the agent on the social learner's performance.

In order to state realistic applications of our method, we need to say that nowadays, various bandit settings are widely used in different aspects of marketing, such as optimizing ad placement, or dynamic pricing [24], [25]. The problem we study arises naturally due to the preserving privacy of users. For example, we can consider an international online web store that uses dynamic pricing to automatically determine the best price for its products. However, due to laws such as the General Data Protection Regulation (GDPR), or different economic situations of different countries, the web store can not use the whole data gathered from customers. It also can create different agents for each part that are able to observe the actions (pricing) of other agents but not the rewards (whether buying the products or not).

We compare our method to the literature [26] and [27] and show that the society in which our agent learns, not only is assumed to have a realistic number of agents (a populated society), but also has agents with a variety of goals and levels of expertise, and not just experts and teachers. In addition, we assume that the social agent, realistically, has no access to

(*equal contributor)

The authors are with School of Electrical and Computer Engineering, University of Tehran, Tehran, Iran

other information about other agents of the society, such as rewards received by other agents. Thus, our proposed method contributes to learning more efficiently in a realistic society by evaluating other agents based on their internal goals and using informative ones, if available, in order to improve their performance. Below, after discussing the literature, we state our assumptions and preliminaries. Following that, we propose our method and the experimental results. In the end, we discuss the conclusion and future work.

II. Related Works

There are several drawbacks to using Reinforcement learning, such as its slowness in producing near-optimal results, suffering from exponentially increased regret in tasks with high-dimensional state space, and having difficulties in adapting online to new environments. Different Social Learning methods can help Reinforcement Learning to address such problems at some level. For example, human guidance can be used in order to speed up learning. However, to achieve this, a human needs to manually specify whether a performed action is correct or incorrect or needs to rank available solutions based on their performance [28]–[30] that makes this method not scalable. Other Social Learning methods used in order to improve Reinforcement Learning and Machine Learning methods are Imitation learning, Vicarious learning, and Observational learning.

Imitation Learning [31] and Observational Learning (also known as Apprenticeship Learning [32]) are both the most used methods in robot learning that is assumed to be a complex problem [33]–[37]. In [38] and [39] the focus is to use Observational Learning in order to improve the agent's performance. In [39] the authors focus on using the observed behavior of a teacher combined with intrinsic motivation to accelerate learning. Similarly, in [38] a deep Reinforcement Learning method combined with memory is used by the agent in order to learn new tasks only through reward signals given by the environment and if it existed, the observed behavior of another agent (named teacher). Through both Observational Learning and Imitation Learning, the agent can speed up its learning by observing a teacher with the goal of teaching the agent, or an expert agent that is supposed to have relevant information for the agent's learning.

One limitation of Imitation and Observational Learning is that other agents associating with the agent are considered to be experts or have relevant information to what the agent needs to learn or some relevant demonstrations are manually obtained. These are not available or are time-consuming to obtain for many tasks especially when the task or environment is novel [22]. In addition, when using Imitation Learning, the agent is forced to follow the behavior of the expert and most of the time duplicate it exactly as it is, so further development is hard to achieve. In contrast to Imitation and Observational Learning, our proposed method needs no expert or teacher in the environment and is not limited to their knowledge and only uses their selected actions if they are considered to be informative by the social agent.

In [40], the authors designed two social learning configurations, to check whether there is a correlation between depressive symptoms and reward learning in a social context. In the first one, the participant just can see the choice of the demonstrator (i.e. imitation), however, in the second one, the participant can see the choice and the outcome of the demonstrator (i.e. Vicarious Learning). Vicarious Learning is similar to Observational and Imitation Learning with the difference that the agent is assumed to have access to both actions and rewards of other agents which is not available in many realistic cases, especially for novel tasks. Humans and other species do not communicate their received internal rewards, such as observing others being praised for their actions, that we understand the consequences of others' actions. Thus, in contrast to Vicarious Learning, we assume that the social agent has no access to other agents' received rewards.

The Social Learning setup we consider in this paper is different from Vicarious, Observational, and Imitation Learning in the sense that different agents in society have various goals which make them either good sources of information for the learner or useless sources. It is also important to note that multi-agent Reinforcement Learning can be considered as a form of social learning. Some multi-agent Reinforcement Learning methods use a social intrinsic motivation as a means to improve performance [41], [42]. However, in multi-agent problems agents have the same goal to achieve and most of them need to share private information through communication such as rewards, observations, or even learning parameters [43]–[46].

Our work is mostly related to [26], [27], [12] and [23]. An online social bandit learning algorithm is introduced in [26] and [27] that use methods inspired from the Upper Confidence Bound (UCB) learning method in order to benefit the decisions of other agents. In [23], a model-based auxiliary loss is added to model-free deep RL to be able to train agents that are able to use cues from expert agents in order to learn new tasks. In [12], the authors compared 4 computational models of imitation in reinforcement learning that are assumed to be used by humans through conducting a social reinforcement learning task. The results showed that the Value Shaping (VS) method can represent imitation better than the other models and self-value guides imitation rate.

Compared to [26] and [23] that assumes other agents to have relevant information, we assume that other agents might be totally irrelevant or misleading for the social agent. In contrast to our method, the authors of [12] designed their hypotheses for a 2-armed bandit environment, with binary rewards, and just one demonstrator. In addition [12], [27] and [23] all consider a limited number of other agents, while we consider a populated society. In contrast to previous works, our setup includes a populated society with a variety of agents with different levels of expertise and goals, without the need for any private information except for selected actions, such as reward, from other agents.

III. Problem Statement and Assumptions

We study the stochastic bandit problem where $\mathcal{A}$ denotes the set of available actions for all agents and $|\mathcal{A}|$ is the number of actions. This setting is also known as k-armed bandits. Each action $a \epsilon \mathcal{A}$ corresponds to an unknown reward distribution with expected value $q_*(a)$. Apart from our social agent, we have N other agents in the problem. All the agents select an action $a_t \epsilon \mathcal{A}$ at each trial t and observe reward $r_t \sim \nu(q_*(a_t))$, where $\nu(q_*(a_t))$ is a probability distribution of reward signal for arm a_t. Only the social agent has the ability to observe the actions selected by other agents at each trial t, $a_{i,t}$, without any other extra information (e.g. reward). Each agent seeks to reach its own internal goal. The goal of the social agent is to maximize its expected reward:

$$\mathbb{E}[R_t] = \sum_{a=1}^{k} \pi_t(a) q_*(a). \tag{1}$$

We did not assume that all agents have the same goal or level of expertise and thus, the other agents do not have a purpose of teaching our agent. Pseudo-regret is considered as a measure for evaluating and comparing algorithms. The pseudo-regret over T trials becomes:

$$\mathcal{R}_T = T \times q_*(a_*) - \sum_{t=1}^{T} R_t, \tag{2}$$

here $q_*(a_*) = max_{a \in \mathcal{A}}\ q_*(a)$, and R_t is the reward signal which is received at trial t. In the following, the term "regret" will be used to refer to "pseudo-regret".

IV. Preliminaries

In this section, we explain the agent's individual learning method, which is used by our social agent. In the following, we provide an overview of the gradient preference-based method that is used in our method to select agents.

A. Agent individual learning method

The individual learning method of the social agent can be any learning method including UCB [24], Thompson sampling [47], etc. Our proposed method can operate independently of the choice of the individual learning method. The chosen individual learning method of our base work is decaying ϵ-greedy, which is one of the simplest Reinforcement Learning methods, as its internal learner method. This method balances exploration and exploitation by taking a random action with probability ϵ, and taking the best-known action thus far otherwise [13]. Considering the fact that exploration is needed at the beginning of the learning, the value of ϵ is better not to be constant and decay over time [48]. Thus, the internal learner of the social agent selects a random action with the probability of:

$$p < \frac{1}{1 + t/N} \tag{3}$$

where t is the number of trials the agent has been learning and N is the number of actions the environment has.

B. Gradient preference-based method

In order to solve an k-armed bandit problem, a numerical preference for each action (initially zero), showing that the action's relative preference over other actions can be calculated. Then, actions are selected with probabilities according to Gibbs or Boltzmann distribution.

$$\Pr(A_t = a) = \frac{e^{H(a)}}{\sum_{all\, action\, b}(e^{H(b)})} = \pi_t(A_t) \tag{4}$$

where $Pr(A_t = a)$ shows the probability of selecting action a at trial t and $H(a)$ shows the preference of action a. After receiving the reward R_t through performing action A_t, the preferences are updated based on the stochastic gradient ascent rule as follows:

$$\begin{aligned} H_{t+1}(A_t) &= H_t(A_t) + \alpha(R_t - \hat{R}_t)(1 - \pi_t(A_t)) \\ H_{t+1}(a) &= H_t(a) - \alpha(R_t - \hat{R}_t)\pi_t(a);\ \forall a \neq A_t, \end{aligned} \tag{5}$$

where $\hat{R}_t$ is the mean reward of all received rewards until trial t ($\hat{R}_t = \hat{R}_t + \frac{1}{n}(R_t - \hat{R}_t)$) and α is constant. This formula increases the preference of the selected action and decreases the preference of other actions if the selected actions result in a reward better than the mean of rewards received, and otherwise decreases the preference of the selected action and increase others [13].

V. Proposed Method

In this paper, we introduce a new method to use the benefits of social learning in reinforcement learning. The social agent of the proposed method learns in a society of different agents and selects action based on two methods: 1) selecting action based on the agent's internal learning method, and 2) selecting action learned from other agents. By learning from other agents we mean to do the action that the other agent has performed frequently.

Knowing that other agents in the environment might have different goals and utility functions and different levels of expertise as well as biases, the social agent needs a method to evaluate them in order to find if other agents are suitable to learn from. Considering that the social agent can only observe actions that other agents perform, it is a challenge to define a measure to detect if any appropriate agent is in the society, select the most informative one to learn from, and integrate the observed information and its own for decision making. Accordingly, we can consider the problem of who to learn from as a multi-armed bandit problem. Inspired by the gradient preference-based learning method, the social agent assigns a preference to all agents including itself, and then uses these calculated preferences while selecting which agent to learn from according to Gibbs or Boltzmann distributions (see (4)). After selecting an agent to learn from, the social agent performs that agent's most frequently performed action that was observed by the social and updates preferences similar to (5) as follows:

$$\begin{aligned} H_{t+1}(Agent_i) &= H_t(Agent_i) \\ &\quad + \alpha(R_t - \hat{R}_t) \times (1 - \pi_{s_t}(Agent_i)); \\ &\quad \text{for all } \mathrm{Agent_i} \text{ with same selected action }, \\ H_{t+1}(Agent_j) &= H_t(Agent_j) \\ &\quad - \alpha(R_t - \hat{R}_t) \times \pi_{s_t}(Agent_j); \\ &\quad \text{for all } \mathrm{Agent_j} \text{ with different selected action} \end{aligned} \tag{6}$$

Here $\hat{R}_t$ is the mean of rewards received from performing actions selected through either selecting others or the agent's internal learner, R_t is the received reward from the performing action, α is constant, and $\pi_{s_t(Agent_i)}$ is the probability of selecting $Agent_i$ to learn from, based on (1). To direct our learning from others, we have chosen policy-gradient-based learning methods because of the generalizability of them to different RL tasks such as continuous and MDP settings. Additionally, these methods do not require excess hyperparameters to be determined during the learning, from which they use gradients to direct their learning. . The derivation of the formula, (6), is explained in appendix A.

Algorithm 1 shows the core procedure of the social agent's learning method. First, the agent selects an action to perform by deciding whether to follow its internal learner or one agent from the society and if the latter, which agent is among them (Line 3). Then, after performing the selected action, it updates the required parameters. At the end of the trial, the agent updates its preference, increasing the preference of the selected agent and all agents with similar last selected action including its internal learner, if the received reward R_i is greater than the mean reward, and decreasing the performance of other agents. Vice versa, if the received reward, R_i is smaller than the mean reward, the agent decreases the preference of all agents who had performed the selected action and increases the preference of others (Line 6).

Algorithm 1 Social Learning

```
Initialize preferences to zero
for each trial $t$ do
    action ← Action Selection {Algorithm 2}
    $R$ ← perform action and receive reward
    Update Preferences {Algorithm 3}
end for
```

Algorithm 2 shows the procedure the social agent uses to select an action. Self-efficacy or the agent's internal level of expertise has an important role in observational learning procedure [49]. Considering that, the social agent needs to evaluate its Self-efficacy in order to not totally depend on other agents, the agent also considers itself as an outer agent and considers the recommended action by its internal learning method as the action to be performed.

Algorithm 2 Action Selection

```
for $agent_i$ in society do
    {Including the social agent}
    $P_i \leftarrow \frac{e^{AgentPreference_i}}{\sum e^{AgentPreference_j}}$
end for
Agent ← Select agent with probability of $P_i$ {See (1)}
if Another Agent is selected then
    Return the most frequent action of Agent
else
    Return the action selected through the internal learner
end if
```

For the sake of sample efficiency and improving performance, all agents in the society who performed the social agent's selected action in the following trial would be considered as the selected agent (Algorithm 3, Line 4). Considering that every bit of information should be used in learning, if the selected action of the agent's internal learner is also the performed action, the preference of agent to its internal learner also is updated (Algorithm 3, Line 13). Thus, after performing the selected action and receiving a reward R_i from the environment, the social agent updates its preference towards itself and other agents in the society (Algorithm 3, Line 15 to 22).

Algorithm 3 Update Preferences

```
Input: $A$: the performed Action, $R$: received reward
Initialize agentList to an empty list
for $agent_i$ in society do
        {Including the social agent}
        if last Action Performed by $agent_i$ is A then
        add $agent_i$ to agentList
    end if
end for
for $agent_i$ in society do
        {Update preferences (4)}
        if $agent_i$ is in agentList then
        $\delta H \leftarrow \alpha(R_t - \hat{R}_t)(1 - \pi_t(Agent_i))$
        $H_{t+1}(Agent_i) \leftarrow H_t(Agent_i) + \delta H$
    else
        $\delta H \leftarrow \alpha(R_t - \hat{R}_t)\pi_t(Agent_j)$
        $H_{t+1}(Agent_j) \leftarrow H_t(Agent_j) - \delta H$
    end if
end for
```

VI. Experimental Results

In order to compare our proposed social learner's performance with that of the individual learner, we test multiple aspects and scenarios. We first test the influence of the size of the society population and the problem difficulty on the difference between social and individual learning. After that, we analyze the influence of society and evaluate our proposed methods' ability in finding which agents are better to learn from. In the end, for further evaluation, we compare our method to the existing similar methods.

Throughout all of our scenarios, the rewards are from the Gaussian distribution $N(\mu, \sigma^2)$ with $\sigma = 1$ and μ randomly selected from one decimal point number between -10 and 10, i.e. $\{(k-100)/10 | k \in [0, 200]\}$, unless otherwise specified. For each test case 30 reward sets $\{N(\mu_i, \sigma^2) | i \in [0, 10]\}$ are chosen randomly for each possible K action counts. Each chosen reward set is tested 10 iterations for 1000 trials. The reported result is the average result of these 10 randomly selected reward sets testing for 10 iterations (300 in total).

Other agents in the society are either another learner (here it uses the Thompson sampling method and we refer it as the Thompson Learner) or the following manually designed agents:

- Random agent: always selects an action randomly.
- Worst agent: always selects the worst action.
- Percent agent (P_0, δP, P_{max}): the action with the most expected reward is selected with a probability of P_0 that is increased by δP each trial until reaching probability equal to P_{max}.

A. Testing the influence of society's population and problem difficulty

Considering that the social agent has the ability to update its preference to other agents with the same selected action as its own action, detecting a group of agents who act similarly and have similar goals, is not hard for our social agent. Thus one of the most complicated groups of agents for the social agent to deal with is a group of random agents. Keeping this fact in mind, in order to evaluate the agent's performance in societies with different population sizes, we assume one of the hardest societies for our social learner that has N+1 agents with N random agents and only one expert that acts rationally and is appropriate to learn from. In appendix B, we discuss this topic in more detail.

Fig. 1 shows the difference between the percent of selecting the optimal action of social and individual for action size k equal to 10 and 100 and society of N equal to 10 and 100. As it is shown, the difference between social and individual is considerable when we have a small society (N = 10) and a difficult problem (k = 100) for the individual learner to learn. In addition, we can observe that when it is harder to find good agents in society (a populated society with N = 100) and the problem itself is easy to learn individually (k = 10), using social learning is not profitable for our agent.

Fig. 2 compares the influence of problem difficulty and population size in the received reward of the social learner compared to the individual learner. We can observe that for an easy problem ($k = 10$), there is no significant difference between the reward of the social agent and that of the individual agent. So, we can conclude that social learning can be useful when a problem is hard to solve individually, and it can enhance convergence speed. On the other hand, when the society is populated ($N = 100$), the improvement of social learning compared to individual learning is decreased.

B. Testing the ability to detect better agents

Considering that among appropriate agents to select from in society some might be better to learn from (because of having faster learning or other reasons), the agent must have the ability to find better agents among appropriate agents in order to benefit more from social learning. We use two societies to test the agent's ability to detect the fittest agents. Both societies have four percent agents as well as 100 random agents and the problem each agent is trying to solve has $k = 10$ actions. Table I shows the properties of percent agents of societies 1 and 2.

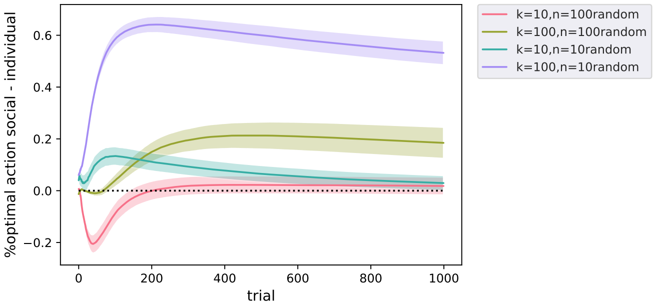

Fig. 1. Difference between percent of selecting the optimal action of social and individual for four cases of (k = 10, N = 10), (k = 100, N = 10), (k = 10, N = 100), and (k = 100, N = 100).

Fig. 3 shows the probability of the agent to select other agents) and itself (Algorithm 2, Line 2) per trial during its learning in societies 1 and 2 (probabilities are uniform at first and in the figure we only draw one random agent and other 99 are excluded). Agent 0 in society 1 has the fastest learning and Agent 1 is the second fastest and reaches 100% of selecting the best action at trial 200. As it is shown in Fig. 3 at the beginning the social agent was able to detect Agent 0 , which is the best and suddenly, about trial 200 increases its selection probability to Agent 1 and decreases the probability of selecting agent 0 that it now acts randomly for 20% of times. In addition, the selection probability or preference of the agent to Agent 3, as expected, was decreased to a probability near random agents after trial 90. Agent 2, which has the slowest learning, reaches 80% of selecting the best action at trial 700 and we can see that the agent was able to increase its preference towards Agent 2 around trial 700. In society 2, the best agent to select after trial 600 is Agent 2 because of having no P_{max} compared to the other three agents. As is shown in Fig. 3 the agent was able to detect this fact and increase its selection probability toward Agent 2 after trial 600. Agent 1 is the second best agent to select (higher P_{max} and fast learning) and we can see that the agent was able to detect that as well but after trial 600 the agent was able to detect that its learner and Agent 2 are now better than Agent 1 and it should change its selection probability according to their changes.

C. Comparing to other methods

In order to compare our method to [27], we test the performance of our social agent and OUCB agent with two parameters β_1 and β_2. Fig. 4 shows the regret of our agent and OUCB agent for a problem with $k = 10$ and a society with one learner. We can observe that our agent outperforms the OUCB in both configurations. In addition, in contrast to our method, it is stated in the paper that OUCB's performance

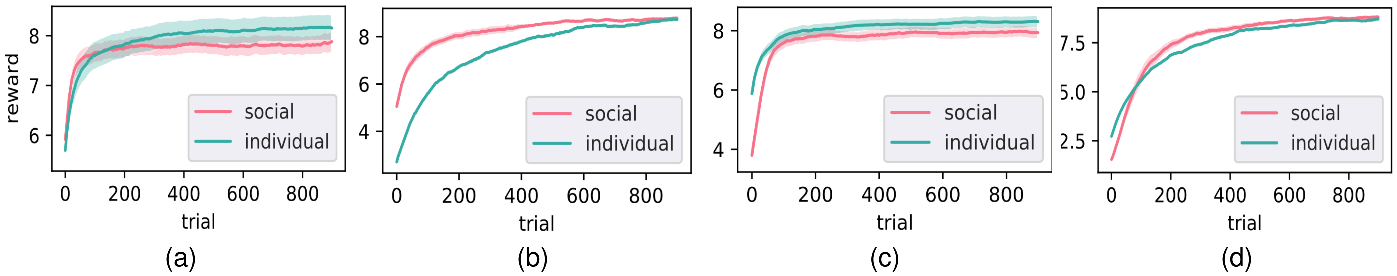

Fig. 2. Comparing reward of social and individual for four cases of (a) k = 10, N = 10, (b) k = 100, N = 10, (c) k = 10, N = 100, and (d) k = 100, N = 100.

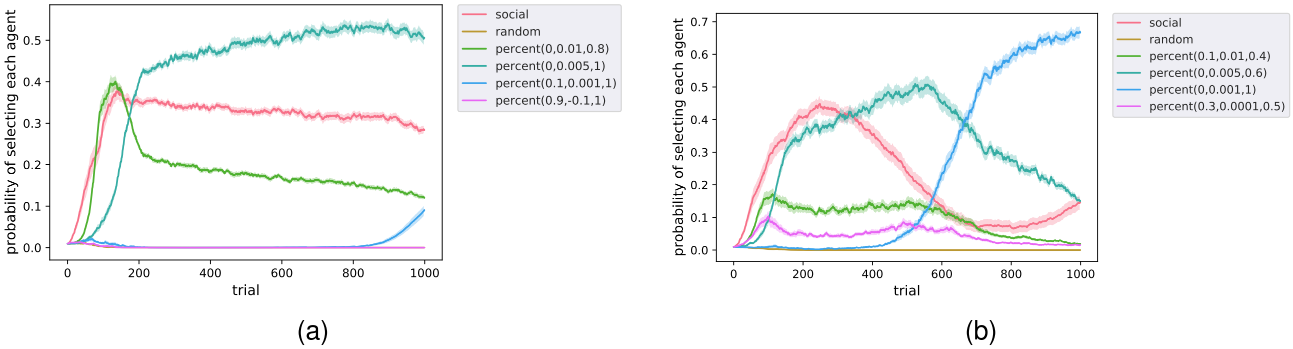

Fig. 3. (a) Selecting probability per trial of social agent for a society with four different percent agents and 100 random agents. (b) Selecting probability per trial of social agent for a society with four different percent agents and 100 random agents.

TABLE I
INDIVIDUAL AND SOCIAL LEARNER P-VALUE WHILE LEARNING A TASK WITH ACTIONS EQUAL TO 8, 32, AND 100

Society	Agent id	P_0 (%)	δP (%)	P_{max} (%)
1	0	0	1	80
	1	0	0.5	100
	2	10	0.1	100
	3	90	-1	100
2	0	10	1	40
	1	0	0.5	60
	2	0	0.1	100
	3	30	0.01	50

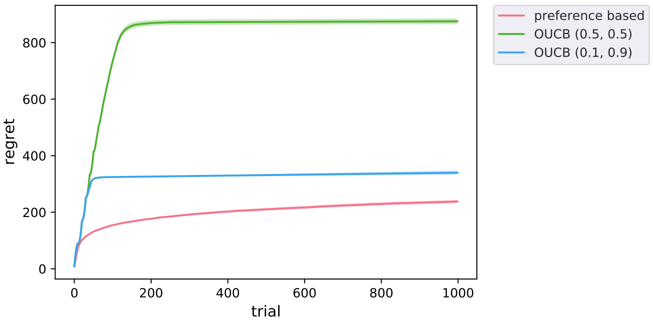

Fig. 4. Our social agent's regret compared to OUCB's for a problem with k = 10 one learner

highly depends on β_1 and β_2 and they need to be tuned.

Fig. 5 shows the regret of our agent and OUCB agent for a problem with $k = 10$ and a society of 100 random agents and one learner. Compared to our method, OUCB is better when we have many random agents and the problem is easy ($k = 10$). The reason for that is in the fact that the frequency of all actions in a large group of random agents (that is considered by OUCB) are equal and as a result random agents do not decrease the performance of OUCB. Thus, if we have an agent that selects the best action for just a few percent more than a random agent, the OUCB algorithm will find the optimal action more easily compared to our method. However, we also need to note that our method works better compared to OUCB when the problem is hard to solve. Fig. 6 shows our regret compared to OUCB's for a problem with $k = 100$, $n = 100$ random agents and one learner.

VII. CONCLUSION

In this paper, we investigate the importance of using social cues in speeding the k-armed bandit problem learning. We consider a realistic case that the agent only can see actions performed by other agents and has access to no other information. We propose a method similar to the gradient preference-based learning method to evaluate other agents in the society and find if there are agents worth learning from. Using a multi-armed bandit analogy, we examined the problem of evaluating other agents to learn from them. By learning from we mean using that agent's most frequent action to improve our performance.

We analyze the agent's preference and ability to evaluate other agents through testing the agent in a variety of societies. We assume cases where there are multiple experts

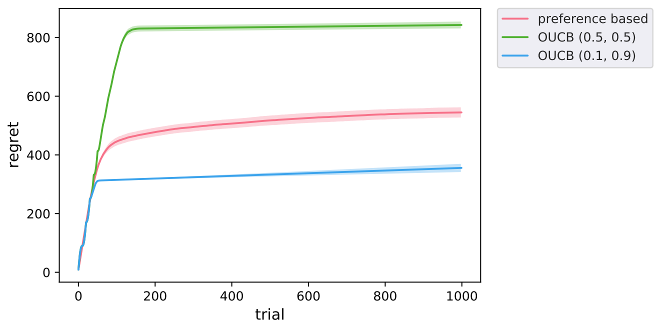

Fig. 5. Our social agent's regret compared to OUCB's for a problem with $k = 10$ and society with $N = 100$ random and one learner

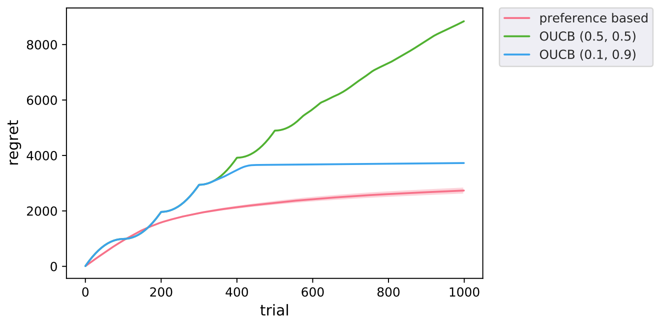

Fig. 6. Our social agent's regret compared to OUCB's for a problem with $k = 100$ and society with $N = 100$ random and one learner

with different levels of expertise in society. We show that the agent was able to find better agents among appropriate agents to learn from. We also show that the performance of the social learning agent was improved considerably compared to the individual learner when the problem is hard to learn individually. We further test our method by comparing adding social learning to other individual learners. As a result we conclude that social learning improves individual learning especially when we have a complex problem. For future work we can apply surprise/novelty signals to extend social learning in non-stationary environments. The social learning method can also be applied to MDP problems to learn complex behavior in long-horizon tasks. Furthermore, it can be used to build better learning agents with safer exploration phases. We did a sensitivity analysis between our method and OUCB in appendix C.

Appendix A
Stochastic Gradient Ascent

Gradient policy algorithms parametrize policy functions and direct them toward gradient direction. In order to do this, we must first define the objective function that we want to maximize. It can be defined as follows :

$$\mathbb{E}[R_t] = \sum_{i=1}^{N} \pi_{s_t}(i) \sum_{a=1}^{k} \hat{\pi}_{i,t}(a) q_*(a), \tag{7}$$

where in this formula R_t is instant reward at trial t, N is number of agents in the environment(include ourself), k is number of actions, and $q_*(a)$ is the mean value of reward distribution of action a. Now we should define $\hat{\pi}_{i,t}$ as policy correspond to agent i^{th} till trial t as follows:

$$\hat{\pi}_{i,t} = \begin{cases} \pi_t : & i = ourself \\ \arg max\ \tilde{N}_{i,t}(a) : & otherwise \end{cases}, \tag{8}$$

where in the above equation π_t is our internal policy at trial t, and $\tilde{N}_{i,t}(a)$ is the number of repetition selection of action a by agent i untill trial t. Furthermore, we define π_{s_t}, selecting agents policy, in the following way:

$$\pi_{s_t}(x) = \frac{e^{H_{s_t}(x)}}{\sum_{y=1}^{N} e^{H_{s_t}(y)}}, \tag{9}$$

where in the equation $H_{s_t}(x)$ represent preference of social agent for selecting agent x (including ourself). Using exact gradient ascent, each agent selection preference,$H_{s_t}(x)$, is incremented proportionally to the increment's effect on performance :

$$H_{s_{t+1}}(x) = H_{s_t}(x) + \alpha \frac{\partial \mathbb{E}[R_t]}{\partial H_{s_t}(x)}. \tag{10}$$

The increment's effect is measured by the partial derivative of expected reward with respect to the selection agent preference. In our problem, exact gradient ascent cannot be implemented because we assume we do not know $q_*(a)$. Nevertheless, we will show that the algorithm's updates (6) and (10) are the same in expected value, making it an instance of stochastic gradient ascent.In order to do this, we start by looking at the exact performance gradient in more detail:

$$\begin{aligned} \frac{\partial \mathbb{E}[R_t]}{\partial H_{s_t}(j)} &= \frac{\partial}{\partial H_{s_t}(j)} [\sum_{i=1}^{N} \pi_{s_t}(i) \sum_{a=1}^{k} \hat{\pi}_{i,t}(a) q_*(a)] \\ &= \sum_{i=1}^{N} \frac{\partial \pi_{s_t}(i)}{\partial H_{s_t}(j)} \sum_{a=1}^{k} \hat{\pi}_{i,t}(a) q_*(a) \\ &= \sum_{i=1}^{N} \frac{\partial \pi_{s_t}(i)}{\partial H_{s_t}(j)} \sum_{a=1}^{k} \hat{\pi}_{i,t}(a) [q_*(a) - B_t]. \end{aligned}$$

In the above formula B_t, called the baseline, is any scalar that is independent of a and i. We can include a baseline here without changing the equality since the gradient sums to zero over all agents, $\sum_{i=1}^{N} \frac{\partial \pi_{s_t}(i)}{\partial H_{s_t}(j)} = 0$. Changing $H_{s_t}(x)$ will cause some agents' selection probabilities to increase and others to decrease, but the sum of the changes must be zero because selection probabilities sum to one. Next we multiply the equation by $\frac{\pi_{s_t}(i)}{\pi s_t(i)}$ and rearrange the equation:

$$\begin{aligned} \frac{\partial \mathbb{E}[R_t]}{\partial H_{s_t}(j)} &= \sum_{i=1}^{N} \pi s_t(i) \sum_{a=1}^{k} \hat{\pi}_{i,t}(a) [q_*(a) - B_t] \frac{\partial \pi_{s_t}(i)}{\partial H_{s_t}(j)} / \pi s_t(i) \\ &= \mathbb{E}_{\pi_{s_t}} [\frac{1}{\pi_{s_t}(i_t)} \frac{\partial \pi_{s_t}(i_t)}{\partial H_{s_t}(j)} (q_*(A_t) - B_t)] \\ &= \mathbb{E}_{\pi_{s_t}} [\frac{1}{\pi_{s_t}(i_t)} \frac{\partial \pi_{s_t}(i_t)}{\partial H_{s_t}(j)} (R_t - \hat{R}_t)], \end{aligned}$$

Here we have chosen the baseline $B_t = \hat{R}_t$, where $\hat{R}_t$ is the mean of all received rewards until trial t and substituted

R_t for $q_{*(A_t)}$ which is permitted because $\mathbb{E}[R_t|A_t] = q_*(A_t)$. Now, we should calculate $\partial \pi_{s_t}(i_t)/\partial H_{s_t}(j)$. Thus:

$$\frac{\partial \pi_{s_t}(x)}{\partial H_{s_t}(j)} = \frac{\frac{\partial H_{s_t}(x)}{\partial H_{s_t}(j)} \sum_{y=1}^{N} e^{H_{s_t}(y)} - e^{H_{s_t}(x)} \frac{\partial (\sum_{y=1}^{N} e^{H_{s_t}(y)}}{\partial H_{s_t}(j)}}{(\sum_{y=1}^{N} e^{H_{s_t}(y)})^2}$$

$$= \frac{\mathbb{I}_{x=j} \ e^{H_{s_t}(x)} \sum_{y=1}^{N} e^{H_{s_t}(y)} - e^{H_{s_t}(x)} e^{H_{s_t}(j)}}{(\sum_{y=1}^{N} e^{H_{s_t}(y)})^2}$$

$$= \frac{\mathbb{I}_{x=j} \ e^{H_{s_t}(x)}}{\sum_{y=1}^{N} e^{H_{s_t}(y)}} - \frac{e^{H_{s_t}(x)} \ e^{H_{s_t}(j)}}{(\sum_{y=1}^{N} e^{H_{s_t}(y)})^2}$$

$$= \mathbb{I}_{x=j} \ \pi_{s_t}(x) - \pi_{s_t}(x)\pi_{s_t}(j)$$

$$= \pi_{s_t}(x) \ [\mathbb{I}_{x=j} - \pi_{s_t}(j)],$$

where in the above equations $\mathbb{I}_{x=j}$ is defined to be 1 if $x = j$, else 0. We intended to write the performance gradient as something that we can sample on each step, as we have just done, and then update each step proportionately to the sample. As a result of substituting a sample from above for the performance gradient in (10), we get:

$$H_{s_{t+1}}(x) = H_{s_t}(x) + \alpha(R_t - \hat{R}_t)(\mathbb{I}_{x=i_t} - \pi_{s_t}(x)), \ \forall x. \quad (11)$$

This would be equivalent to our original algorithm (6). We just demonstrated that the expected update of gradient bandit algorithms equals the gradient of expected rewards, thus the algorithm is an example of stochastic gradient ascent. The algorithm, therefore, has robust convergence properties.

Appendix B
The worst agent for social learning

The objective of this part is to determine which agents are the worst agents that can exist in society for our social agent. In every trial, we updated the preferences of the social agent about all other agents based on whether other agents selected the same action as the social agent. Thus, the behavior policy of other agents in society plays an important role in the updating process. Consequently, an agent with a uniform policy would be the worst agent in society, since it would have the maximum entropy. In other words, our agent has the greatest uncertainty about the random agents, which have uniform policies.

Appendix C
Sensitivity Analysis

In order to do a sensitivity analysis, we plotted the Fig. 7. In Fig. 7 the difference between the percent of selecting the best action at the last trial of our method and OUCB is plotted for different sets of beta1 and beta2. We plotted them in two 2D plots for simplicity of interpretation. As we can see for a society with one other learner and 100 random agents, our method performs better for all cases except one in which both methods perform equally.

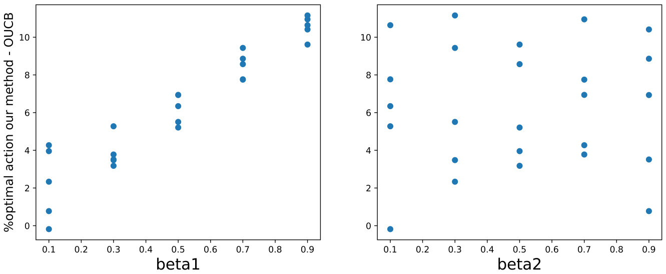

Fig. 7. A sensitivity analysis that compares our social learning method to OUCB with a broader parameter space.

References

[1] A. Bandura and R. H. Walters, *Social learning theory*. Englewood cliffs Prentice Hall, 1977, vol. 1.

[2] J. Henrich and R. McElreath, "The evolution of cultural evolution," *Evolutionary Anthropology: Issues, News, and Reviews: Issues, News, and Reviews*, vol. 12, no. 3, pp. 123–135, 2003.

[3] K. N. Laland, "Social learning strategies," *Animal Learning & Behavior*, vol. 32, no. 1, pp. 4–14, 2004.

[4] N. Humphrey, "The social function of intellect," 1976.

[5] E. Herrmann, J. Call, M. V. Hernández-Lloreda, B. Hare, and M. Tomasello, "Humans have evolved specialized skills of social cognition: The cultural intelligence hypothesis," *science*, vol. 317, no. 5843, pp. 1360–1366, 2007.

[6] R. Boyd, P. J. Richerson, and J. Henrich, "The cultural niche: Why social learning is essential for human adaptation," *Proceedings of the National Academy of Sciences*, vol. 108, no. Supplement 2, pp. 10 918–10 925, 2011.

[7] C. P. Van Schaik and J. M. Burkart, "Social learning and evolution: the cultural intelligence hypothesis," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 366, no. 1567, pp. 1008–1016, 2011.

[8] Y. N. Harari, *Sapiens: A Brief History of Humankind*, 1st ed. McClelland Stewart, 2014.

[9] J. Henrich, *The Secret of Our Success: How Culture Is Driving Human Evolution, Domesticating Our Species, and Making Us Smarter*, 1st ed. Princeton University Press, 2015.

[10] K. N. Laland, *Darwin's Unfinished Symphony: How Culture Made the Human Mind*, 1st ed. Princeton University Press, 2017.

[11] M. Kleiman-Weiner, "Computational foundations of human social intelligence," Ph.D. dissertation, Massachusetts Institute of Technology, 2018.

[12] A. Najar, E. Bonnet, B. Bahrami, and S. Palminteri, "The actions of others act as a pseudo-reward to drive imitation in the context of social reinforcement learning," *PLoS biology*, vol. 18, no. 12, p. e3001028, 2020.

[13] R. S. Sutton and A. G. Barto, *Reinforcement learning:An introduction*, 2nd ed. the MIT Press, 2018.

[14] D. B. Lenat, "Am, an artificial intelligence approach to discovery in mathematics as heuristic search," 1976.

[15] J. Schmidhuber, "Adaptive confidence and adaptive curiosity," Institut fur Informatik, Technische Universitat Munchen, Arcisstr. 21, 800 Munchen 2, Tech. Rep., 1991.

[16] S. Singh, A. G. Barto, and N. Chentanez, "Intrinsically motivated reinforcement learning," MASSACHUSETTS UNIV AMHERST DEPT OF COMPUTER SCIENCE, Tech. Rep., 2005.

[17] P. Capdepuy, D. Polani, and C. L. Nehaniv, "Maximization of potential information flow as a universal utility for collective behaviour," in *2007 IEEE Symposium on Artificial Life*. Ieee, 2007, pp. 207–213.

[18] S. Still and D. Precup, "An information-theoretic approach to curiosity-driven reinforcement learning," *Theory in Biosciences*, vol. 131, no. 3, pp. 139–148, 2012.

[19] S. Mohamed and D. J. Rezende, "Variational information maximisation for intrinsically motivated reinforcement learning," *arXiv preprint arXiv:1509.08731*, 2015.

[20] Z. Zhu, K. Lin, and J. Zhou, "Transfer learning in deep reinforcement learning: A survey," *arXiv preprint arXiv:2009.07888*, 2020.

[21] M. Ghorbani, R. Hosseini, S. P. Shariatpanahi, and M. N. Ahmadabadi, "Reinforcement learning with subspaces using free energy paradigm," *arXiv preprint arXiv:2012.07091*, 2020.

[22] N. Jaques, "Social and affective machine learning," Ph.D. dissertation, Massachusetts Institute of Technology, 2019.

[23] K. K. Ndousse, D. Eck, S. Levine, and N. Jaques, "Emergent social learning via multi-agent reinforcement learning," in *International Conference on Machine Learning*. PMLR, 2021, pp. 7991–8004.

[24] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Machine learning*, vol. 47, no. 2, pp. 235–256, 2002.

[25] K. Misra, E. M. Schwartz, and J. Abernethy, "Dynamic online pricing with incomplete information using multiarmed bandit experiments," *Marketing Science*, vol. 38, no. 2, pp. 226–252, 2019.

[26] A. Lupu, A. Durand, and D. Precup, "Leveraging observations in bandits: Between risks and benefits," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 6112–6119.

[27] J. Zong, T. Liu, Z. Zhu, X. Luo, and H. L. Qian, "Social bandit learning: Strangers can help," *2020 International Conference on Wireless Communications and Signal Processing (WCSP)*, pp. 239–244, 2020.

[28] W. B. Knox and P. Stone, "Interactively shaping agents via human reinforcement: The tamer framework," in *Proceedings of the fifth international conference on Knowledge capture*, 2009, pp. 9–16.

[29] P. Christiano, J. Leike, T. B. Brown, M. Martic, S. Legg, and D. Amodei, "Deep reinforcement learning from human preferences," *arXiv preprint arXiv:1706.03741*, 2017.

[30] D. Sadigh, A. D. Dragan, S. S. Sastry, and S. A. Seshia, "Active preference-based learning of reward functions," in *Robotics: Science and Systems*, 2017.

[31] Y. Duan, M. Andrychowicz, B. C. Stadie, J. Ho, J. Schneider, I. Sutskever, P. Abbeel, and W. Zaremba, "One-shot imitation learning," in *NIPS*, 2017.

[32] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," in *Proceedings of the twenty-first international conference on Machine learning*, 2004, p. 1.

[33] J. Hua, L. Zeng, G. Li, and Z. Ju, "Learning for a robot: Deep reinforcement learning, imitation learning, transfer learning," *Sensors*, vol. 21, no. 4, p. 1278, 2021.

[34] A. Shon, K. Grochow, A. Hertzmann, and R. P. Rao, "Learning shared latent structure for image synthesis and robotic imitation," in *Advances in neural information processing systems*, 2006, pp. 1233–1240.

[35] S. Schaal, "Is imitation learning the route to humanoid robots?" *Trends in cognitive sciences*, vol. 3, no. 6, pp. 233–242, 1999.

[36] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and autonomous systems*, vol. 57, no. 5, pp. 469–483, 2009.

[37] A. Billard, S. Calinon, and R. S. S. Dillmann, "Robot programming by demonstration," in *Springer Handbook of Robotics*. Berlin, Heidelberg: Springer, 2008, ch. 59.

[38] D. Borsa, B. Piot, R. Munos, and O. Pietquin, "Observational learning by reinforcement learning," *arXiv preprint arXiv:1706.06617*, 2017.

[39] R. Rayyes, H. Donat, J. Steil, and M. Spranger, "Interest-driven exploration with observational learning for developmental robots," *IEEE Transactions on Cognitive and Developmental Systems*, pp. 1–1, 2021.

[40] L. Safra, C. Chevallier, and S. Palminteri, "Depressive symptoms are associated with blunted reward learning in social contexts," *PLoS computational biology*, vol. 15, no. 7, p. e1007224, 2019.

[41] C. Breazeal (Ferrell), "A motivational system for regulating human-robot interaction," in *Proceedings of the Fifteenth National/Tenth Conference on Artificial Intelligence/Innovative Applications of Artificial Intelligence*, ser. AAAI '98/IAAI '98. USA: American Association for Artificial Intelligence, 1998, p. 54–62.

[42] E. Hughes, J. Z. Leibo, M. G. Phillips, K. Tuyls, E. A. Duéñez-Guzmán, A. G. Castañeda, I. Dunning, T. Zhu, K. R. McKee, R. Koster *et al.*, "Inequity aversion improves cooperation in intertemporal social dilemmas," *arXiv preprint arXiv:1803.08884*, 2018.

[43] J. N. Foerster, Y. M. Assael, N. De Freitas, and S. Whiteson, "Learning to communicate with deep multi-agent reinforcement learning," *arXiv preprint arXiv:1605.06676*, 2016.

[44] J. Foerster, G. Farquhar, T. Afouras, N. Nardelli, and S. Whiteson, "Counterfactual multi-agent policy gradients," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[45] A. Lazaridou, K. M. Hermann, K. Tuyls, and S. Clark, "Emergence of linguistic communication from referential games with symbolic and pixel input," *ArXiv*, vol. abs/1804.03984, 2018.

[46] K. Cao, A. Lazaridou, M. Lanctot, J. Z. Leibo, K. Tuyls, and S. Clark, "Emergent communication through negotiation," *arXiv preprint arXiv:1804.03980*, 2018.

[47] D. Russo, B. Van Roy, A. Kazerouni, I. Osband, and Z. Wen, "A tutorial on thompson sampling," *arXiv preprint arXiv:1707.02038*, 2017.

[48] A. Maroti, "Rbed: Reward based epsilon decay," *arXiv preprint arXiv:1910.13701*, 2019.

[49] A. Bandura, *Social Foundations of Thought and Action: A Social Cognitive Theory*, 2nd ed. Prentice-Hall, 1986.