Enhanced Learning from Multiple Demonstrations with a Flexible Two-level Structured Approach

Paper #

Abstract

Learning from demonstration has emerged as a successful technique to speed up reinforcement learning. However, such success is limited by the quality of the demonstrations. This paper investigates how to mitigate the impact of noisy or poor demonstrations. Factors like inconsistent or conflicting information provided from different demonstrators or noise in the demonstrations may affect the policy generalization and hurt the learning performance. To address the above challenge, this paper drew inspiration from the multi-armed contextual bandit problem and Human Agent Transfer algorithm to develop a "Flexible Two-level Structured Approach." The proposed approach allows the RL agent to extract useful information from multiple demonstrations of different quality, mitigate the effects of noise or bad examples, while retaining the benefits provided by the good ones. Evaluated on Mario, Cart Pole, and an RC Car domain, experimental results show that this approach holds promise and can successfully leverage demonstrations of different quality.

1 Introduction

Reinforcement learning [Sutton and Barto, 1998] (RL) has had many successes in sequential decision tasks, where an agent learns to maximize a real-valued reward. However, learning tabula rasa can be slow, particularly in difficult domains. Learning from demonstration [Argall *et al.*, 2009] (LfD) is an alternative formulation where an agent typically learns to mimic a human demonstrator. Following this general idea, a rich set of LfD techniques have been developed. For example, one of those techniques is the Human Agent Transfer [Taylor *et al.*, 2011] (HAT) algorithm, which allows the agent to summarize the demonstrated policies with a rule-based learner, the learned rules are transferred to the agent [Taylor and Stone, 2007], and then used in a probabilistic policy reuse [Fernández and Veloso, 2006] (PPR) manner to improve the agent's performance. However, the majority of such techniques are still limited by the demonstrator's performance: an LfD agent's goal is typically to mimic the human, while an RL agent's goal is to maximize total reward.

In this paper, we are concerned with how to best combine LfD with RL so that we can reap the benefits of fast learning, while not being limited by the demonstrator's ability. In particular, we focus on cases of heterogeneous demonstrations. One could consider combining demonstrations from sources with different average performances, or one demonstrator with a high variance in performance. In both cases, we would like to maximize how much we can learn from the demonstrators while minimizing how much poor demonstrations hurt the learner.

In dealing with this problem, several straightforward solutions have been developed, including removing unnecessary or inefficient examples from the demonstration [Friedrich and Dillmann, 1995], requesting additional clarifications [Chernova and Veloso, 2007], etc. However, those ideas try to eliminate the effects of heterogeneous and noisy demonstrations by removing them instead of trying to extract useful information and learn from them. The challenge of how to effectively learn from heterogeneous demonstrations, making use of the beneficial ones while avoiding the influence of bad ones, still exists.

To address this challenge, we draw inspiration from the multi-armed contextual bandit problem and propose a flexible two-level structure based on the probabilistic policy reuse scheme of HAT: level 0 uses multiple classifiers to summarize demonstrations and level 1 takes advice from the low-level classifiers and combines their opinions with a decision algorithm (e.g., majority voting).

Our experimental results show that the two-level structure can improve the overall performance (total reward) and initial performance (jumpstart), while minimizing the effects of bad demonstrations (relative to the existing HAT algorithm).

2 Background and Related works

This section briefly introduces the techniques needed to understand this paper and related work in RL, LfD, and the contextual multi-armed bandit problem.

2.1 Reinforcement Learning

Reinforcement learning is an approach for an agent to learn from experience through the interaction with the environment [Sutton and Barto, 1998].

RL tasks are often formalized as Markov Decision Processes (MDPs), where A is the set of available actions, S is the set of states, the transition function $T : S \times A \rightarrow S$ defines how the state changes over time, and the reward function $R : S \times A \rightarrow \mathbb{R}$ is to be maximized. Action-value function $Q : S \times A \rightarrow \mathbb{R}$ provides the estimated value of state-action pairs. At each time step, the agent will select an action to execute and receive the states and reward determined by the environment. The selection of actions follows a policy π, which could be approximated with the estimated value of a certain state-action pair $Q(s, a)$. During the updating of $Q(s, a)$, the agent gradually improves the policy and maximizes the expected reward. This could be done with different algorithms like Sarsa, Q-learning [Watkins and Dayan, 1992] (which is used in this paper), etc.

2.2 Learning from Demonstration

LfD is a useful technique to deal with the poor initial performance of *tabula rasa* agents [Argall *et al.*, 2009]. Demonstrations are usually trajectories of state-action pairs performed by human (or, sometimes, other agents). With demonstrations recorded from same or similar tasks, the agent could transfer the learned features of neural networks [Yosinski *et al.*, 2014], learn heuristics for reward shaping [Brys *et al.*, 2015], construct skill trees with experience replay [Konidaris *et al.*, 2010], etc.

Among those related works, the Human Agent Transfer [Taylor *et al.*, 2011] (HAT) algorithm first summarizes the policies with a rule-based learner (e.g., a decision tree) then uses the rule transfer [Taylor and Stone, 2007] technique of transfer learning in a Probabilistic Policy Reuse [Fernández and Veloso, 2006] (PPR) manner, to speed up and improve the performance of an RL agent. This paper also adopts the PPR technique, while focusing on making use of multiple demonstrations regardless of their qualities, and more details could be found in 3.2.

There are also works that follow the Reinforcement Learning with Expert Demonstrations (RLED) framework, which try to extract optimal policies by directly applying iterations on the demonstration data [Kim *et al.*, 2013; Piot *et al.*, 2014; Chemali and Lazaric, 2015], identify relevant task features through demonstrations and apply RL with the abstract state space [Cobo *et al.*, 2011], etc. Different from those solutions, our approach focuses on guiding the learning process using policies generalized from demonstrations instead of explicitly using the demonstration examples as an additional source for updating value estimates.

2.3 Learning from Multiple Experts

Online learning with expert advice could be viewed as a contextual multi-armed bandit problem [Lu *et al.*, 2010], where the agent's goal is to find the best expert (that has the highest expected reward). At each time step, an expert will generate advice about the arm selection based on the current context, and the agent will build a strategy of pulling the arm, considering all the given advice, and dynamically update its belief of each expert [Zhou, 2015]. The most relevant work is the Exponential-weight Algorithm for Exploration and Exploitation using Expert advice (EXP4) algorithm [Auer *et al.*, 2002], and similar ideas are also adopted in this paper. Additionally, such modeling could be applied to applications like recommendation [Li *et al.*, 2010], selecting state machine policies for robotic systems [Matikainen *et al.*, 2013], etc.

3 Flexible Two-Level Structured Approach

In this section, we propose a Flexible Two-level Structured Approach (FTSA) to address the problem of combining multiple demonstrations. This approach integrates the multi-armed bandit approach with the probability policy reuse scheme of HAT and allows the agent to take advantage of multiple demonstrations regardless of their quality.

3.1 Problem Statements

To address the problem of learning from multiple demonstrations, we formalize this as a contextual multi-armed bandit problem: With N experts (or classifiers where each is trained on a demonstration), at each time step t, each expert will give out advice vector ξ based on the context vector x_t, and the agent will select K actions according to a certain strategy. The strategy will be adjusted according to the reward r of the selected action. The goal of the agent is to maximize its reward.

3.2 Methodology

To capture the multi-armed bandit selection, we propose a Flexible Two-level Structured Approach (FTSA) based on the framework of HAT and adopt the Probabilistic Policy Reuse (PPR) [Fernández and Veloso, 2006] scheme to determine whether the agent will accept the suggested actions. With a certain probability ϕ, the agent will follow the recommended actions, otherwise, continue following its value estimation or explore around. The probability ϕ usually less than 1 and decays exponentially over time. The two levels of the FTSA approach will be responsible for extracting policies from the demonstration with classifiers, then work as a decision component to select/combine multiple advice and make the suggestion to the agent. Figure 1 illustrates how this two-level structure works, and more details could be found in Algorithm 1.

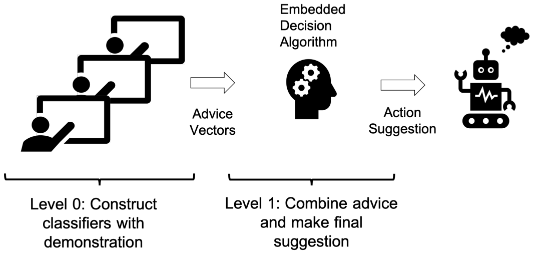


Figure 1: Two-level Structure of Bandit Selection

Level 0: Construct Classifiers with Demonstrations
Level 0 will take the provided demonstrations as inputs and build ”expert” classifiers with any supervised learning methods. Each classifier would be trained with only one demonstration, and it will summarize the policies according to the transactions of the demonstration. Those ”experts” will then generate advice according to the context while the agent performing its task and interacting with the environment.

Level 1: Combine Advice and Make Suggestion

Level-1 will take the advice from the level-0 classifiers, and the embedded decision component will combine the advice and make a final suggestion to the agent. There are widely choices of the decision component, and in this paper, we choose Majority Vote, a meta-classifier and Exponential Weighted Algorithm as examples to show how this approach works:

1. **Majority Vote** The majority vote is an efficient rule in making decisions among alternatives. And when applying this to the approach, the entire structure would become similar to the bagging ensemble method [Breiman, 1996]. Since the advice from each classifier is treated equally, this method works well when the quality of most demonstrations are good or similar to each other, as the advice from those few classifiers trained with bad demonstrations will be ignored.
2. **Meta-Classifier** Building a meta-classifier based on the advice of level 0 could also be a way of constructing the decision component of FTSA, and it would be similar to the stacking (stacked generalization) ensemble method. The level 1 classifier could use either the same base classification method as level 0 does or different methods.
3. **Exponential Weighted Algorithm (EWA)** In contrast to the majority vote, EWA will assign weights to each classifier and keep updating the weights based on the advice quality. In each round, the agent will estimate the reliability of each classifier based on the advice they give and choose which classifier to listen to and use the reward from the environment (or other loss functions) to update the weights accordingly. In this way, the FTSA would become similar to the EXP4 bandit algorithm [Auer *et al.*, 2002]. And the difference is that the exploration/exploitation balancing will not be handled by the decision component itself, but by the HAT framework. And compared to the previous two choices, using EWA would give an expected regret bounded by $O(\sqrt{TKlnN})$ (where K is the number of arms/actions, and N is the number of experts, as detailed in the Appendix).

4 Evaluation

This section discusses the empirical evaluation of the proposed methods.

4.1 Experimental Settings

Three domains are selected for the evaluation of the proposed approach:

Mario is a benchmark domain [Karakovskiy and Togelius, 2012] based on the Nintendo's game Super Mario Bros. The state is represented as a 27-tuple vector and the agent can choose from 12 different actions with the combinations of three sets: left, right, no action, jump, do not jump, run, do not run. The reward function is detailed elsewhere [Suay *et al.*, 2016].

Cartpole is a classical control problem that balancing a pole attached to a cart and prevent it from falling over. We use the simulator released by the RLPy [Geramifard *et al.*, 2013] for the experiments. The continuous state is represented by a 4-tuple vector of angle, angular rate, position, and velocity. Possible actions are applying a force of +10 or -10 Newtons to the cart. The agent receives a +1 reward for each step it has not fallen, and a 0 otherwise.

Algorithm 1: Flexible Two-level Structured Approach for LfD

```
Input: Demonstration data set $D = \{d_0, d_1, ..., d_n\}$,
       reuse probability $\Phi_0$, decay rate $\Phi_D$
$\Phi \leftarrow \Phi_0$
for demonstration $d_i \in D$ do
    Train classifier $c_i$ with $d_i$ ▷ Build level-0 classifiers
end
Initialization for level-1 algorithms
for each episode do
    Initialize state $s_0$
    for step t do
        Get state vector $s_t$, reward $r_t$
        $a \leftarrow \phi$
        if $rand() \leq \Phi$ then
            Construct context vector $x_t$ with $s_t$
            for each classifier $c_i$ do
                Get $\xi_t^i$ with $x_t$
            end
            Apply level-1 algorithms to select action $a$
                ▷ i.e. majority vote, value estimation,
              classification, etc.
        else
            if $rand() \leq \epsilon$ then
                $a \leftarrow$ random action
            else
                $a \leftarrow argmax_a Q$
        Execute action $a$
        Update Q-value (with SARSA, Q-Learning,
          etc.)
        $\Phi \leftarrow \Phi * \Phi_D$
    end
end
```

RC Car is a remote control car simulator by RLPy [Geramifard *et al.*, 2013], where a car learns to drive towards a pre-defined target. The state is the 2-tuple vector of the car's position inside the room. 9 actions are possible as the combinations of forward, coast, backward, turn left, go straight, turn right. The agent receives a +200 for reaching the goal and -1 for every timestep.

To investigate whether our approach could learn reasonable performance with multiple demonstrations, we conduct experiments and provide the agent with: 1) good demonstrations, 2) bad demonstrations, and 3) a mixture of demonstrations with different quality. For Mario, we use 10 good demonstrations and 20 bad demonstrations. For Cartpole, we use {10, 25, 50} good demonstrations and {25, 50, 120} bad demonstrations. For the RC Car, we use {10, 20, 50} good demonstrations and $\{3, 8, 20\}$ bad demonstrations.[1]

We use the J48 decision tree to build base classifiers in level-0 and the following algorithms are embedded in level-1 as decision components:

1. Majority Vote;
2. Exponential Weighted Algorithm, where $\eta = 0.001$, iterate for 100 rounds; and
3. Meta-classifier, using J48 as level-1 classifier.

All the good demonstrations are recorded with a well-trained Q-learning agent and bad demonstrations are recorded with a simple agent in all three domains. The baseline is the Q-learning agent without any prior knowledge. We adopt the HAT agent with J48 as a benchmark method (which uses a merged data file of all demonstrations as input).

To avoid falling into the local optimum and encourage the exploration, we also apply the **Adaptive Discount Factor** strategy [François-Lavet *et al.*, 2015] in the experiments with the Cartpole and RC Car domains. With this strategy, the discount factor changes along the way according to the following formula:

$$\gamma_{k+1} = 1 - 0.98(1 - \gamma_k)$$

where k is the current epoch/episode.

For the evaluation metrics, we consider 1) *Jumpstart*, the improvements of initial performance vs. the benchmark agent and 2) *Total Reward*, the accumulated reward achieved by an agent (i.e., the area under the learning curve).

4.2 Results and Discussion

To present the results of the experiments, we plot the learning curves with the averaged score over 10 runs with a sliding window. The length and number of episodes, and the window size of the sliding window differs for each domain.

Mario

According to Figure 2a, when using 10 good demonstrations, the performance of the proposed approach is much better than the baseline, and could converge to a similar performance level of HAT; at the beginning stage, HAT has the highest jumpstart, and FTSA with majority vote has the lowest jumpstart, while the jumpstarts of using EWA and the meta-classifier lie between. However, even with a low jumpstart, the performance of FTSA with all different methods could catch up with the performance of the HAT within 5,000 episodes.

Figure 2b shows the performance of training an agent with the proposed approach using 10 good demonstrations and 20 bad demonstrations. With the blending of good and bad demonstrations, both HAT and FTSA with all the methods could outperform the baseline. The jumpstart of HAT is lower than the approach with EWA and majority vote but higher than FTSA with meta-classifier. The performance level of all the methods will converge to a similar level between 15,000 to 20,000 episodes.

From Figure 2c, we could observe that, with 20 bad demonstrations, FTSA with majority vote and EWA could still provide some benefit to the jumpstart, while HAT and FTSA with meta-classifier could not. With the bad jumpstarts, the performance of HAT and FTSA with the meta-classifier could converge to a higher level than the baseline after 10,000 episodes, even though they could not benefit from the demonstrations at the beginning stage.

Cartpole

Figure 3a shows the performance of providing 50 good demonstrations to the agents. Both HAT and the proposed approach perform much better than the baseline, they could "capture" the near-optimal policy from the demos at the beginning stage, and remain at such a steady performance level. Figure 3b shows the performance of providing 50 good demonstrations and 120 bad demonstrations. With this setting, HAT and FTSA with all the methods still have a similar level of jumpstart and could outperform the baseline. Figure 3c shows that with 120 bad demonstrations, HAT and FTSA with meta-classifier have negative jumpstarts, then gradually converge to a performance level similar to the baseline. But FTSA with the majority vote and EWA could still benefit from the bad demonstrations, have a good jumpstart, and maintain a higher performance level.

RC Car

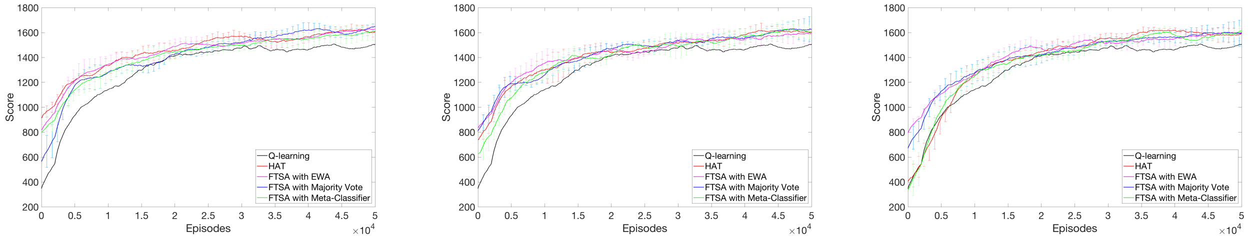
According to Figure 4a, when providing 50 good demonstrations, both HAT and the proposed approach could outperform the baseline. And at the beginning stage, FTSA with majority vote has the highest jumpstart, then HAT and FTSA with meta-classifier have a similar level of jumpstarts, and FTSA with EWA has the lowest jumpstart. And all those approaches could keep the steady performance and converge to a similar performance level as the baseline. From Figure 4b, we could observe that with 50 good demonstrations and 120 bad demonstrations, FTSA with majority vote and EWA could still have a high jumpstarts and maintain a steady performance level; HAT and FTSA with meta-classifier have lower jumpstarts, but could catch up later and converge to the similar performance level of baseline. When only providing 120 bad demonstrations, from 4c, both of the transfer approaches have positive jumpstarts at the beginning stage, but their final performance will be affected by the bad demonstrations.

According to the experimental results, this 2-level structure could efficiently capture the knowledge from multiple demonstrations. Embedded with different bandit algorithms, it could eliminate the effects of bad demonstrations better than the HAT approach and enable the agent to better use demonstrations of varying quality.

5 Conclusion and Future Works

In this paper, we proposed a flexible two-level structured approach to address the problem of learning from multiple demonstrations, regardless of the quality of the demonstrations. Formalizing this as a contextual multi-armed bandit problem, and embedding with different decision algorithms

[1] When choose the combination of good and bad demonstrations, we adjust the scale and present approximately the same number of examples from good and bad demonstrations. For example in the Mario domain, our 10 good demonstrations contain 23,259 state/action example pairs, and our 20 bad demonstrations contain 25,594 examples.

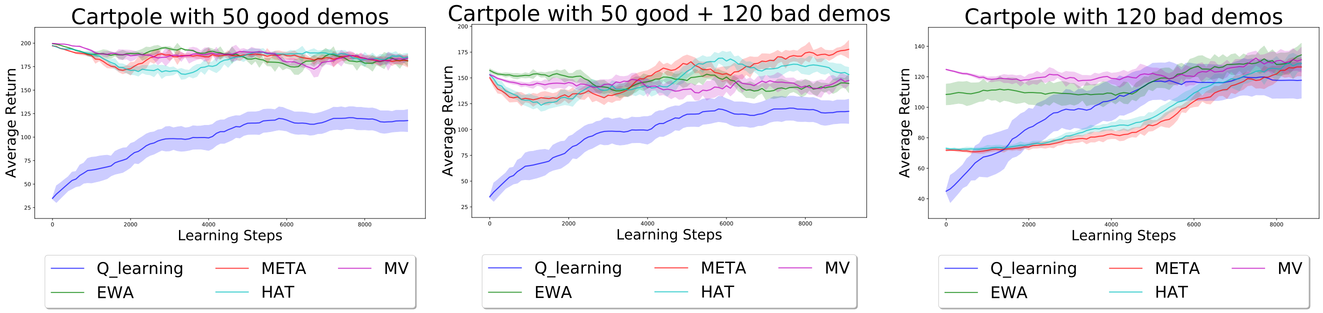


(a) With 10 good demonstrations

(b) With 10 good and 20 bad demonstrations

(c) With 20 bad demonstrations

Figure 2: Learning curves of providing different demonstrations to the agent in the Mario domain



(a) With 50 good demonstrations

(b) With 50 good and 120 bad demonstrations

(c) With 120 bad demonstrations

Figure 3: Learning curves of providing different demonstrations to the agent in the Cartpole domain

within the Human-Agent Transfer framework, this approach could summarize and combine the policies from multiple demonstrations. With the empirical experiments in three domains, this approach could outperform or has similar performance level when comparing to the HAT. In scenarios where providing bad demonstrations, this approach shows its robustness that it could successfully "capture" the useful information. Therefore we could conclude that this two-level structure enables the agent to benefit from multiple demonstrations efficiently while eliminating the effects of bad demonstrations.

Future work will consider multiple directions. First, we are going to investigate the possibility of combining the confidence-based decision scheme into this approach (e.g., CHAT [Wang and Taylor, 2017]) to access the reliability of action advice given by the demonstrations. Second, instead of using the PPR scheme of HAT, we want to explore the possibility of introducing "advising reward" [Omidshafiei *et al.*, 2018] to enable the agent to develop a decision policy from sketch. Third, we want to further validate the effectiveness of this approach in additional complex application scenarios, such as online settings, multi-agent systems, etc. Fourth, considering the noise and irregularity of the collected demonstration data, an in-depth analysis of the demonstrations can potentially be useful to further improve the effectiveness and explainability of the proposed framework.

A. Proof of the Bounded Expected Regret

Here we provide the proof that the upper bounded expected regret of using EWA as the 2-level decision algorithm is $O(\sqrt{TKlnN})$.

The highest expected reward G_{max} is

$$G_{max} = \max_i \sum_{t=1}^{T} \xi_t^i \cdot r_t$$

where ξ_t^i is expert i's advice vector at time t and r_t is the reward vector.

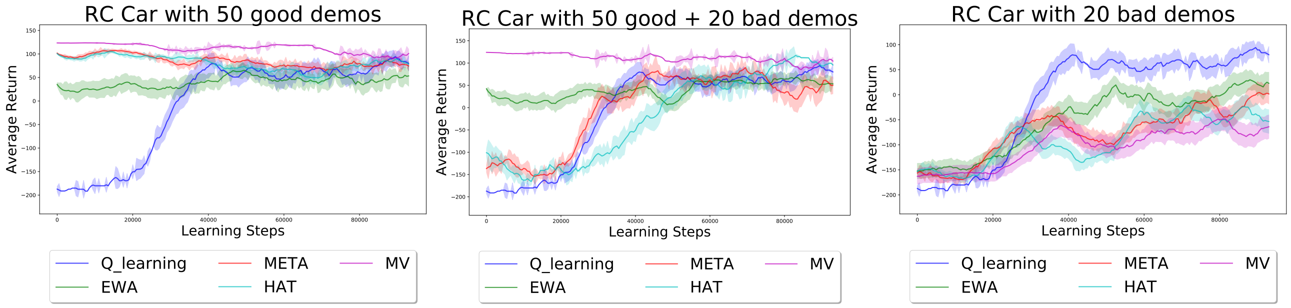
The regret can be written as

$$\begin{aligned} R_T &= G_{max} - E[G] \\ &= \max_i \sum_{t=1}^{T} \xi_t^i \cdot r_t - E\sum_{t=1}^{T} r_t(a_t) \end{aligned}$$

With the update rule

$$w_i(t+1) = w_i(t) \cdot exp(\eta \cdot \hat{y_i}(t))$$

Let $W_t = w_1(t) + ... + w_K(t)$. For all sequences $i_1, i_2, ..., i_T$ of actions, and let $q_i(t) = \frac{w_i(t)}{W_t}$, then



(a) With 50 good demonstrations (b) With 50 good and 20 bad demonstrations (c) With 20 bad demonstrations

Figure 4: Learning curves of providing different demonstrations to the agent in the RC Car domain

$$\begin{aligned}\frac{W_{t+1}}{W_t} &= \sum_{i=1}^{N} \frac{W_i(t+1)}{W_t}\\ &= \sum_{i=1}^{N} q_i(t)(\eta \cdot \hat{y}_i(t))\\ &\le \sum_{i=1}^{N} q_i(t)[1 + \eta\hat{y}_i(t) + (e-2)\eta^2\hat{y}_i(t)^2]\\ &\le exp[\eta \sum_{i=1}^{N} q_i(t)\hat{y}_i(t) + (e-2)\eta^2\hat{y}_i(t)^2]\end{aligned}$$

Since $1 + x \le e^x \le 1 + x + (e-2)x^2$ for $x \le 1$.
Taking logarithms and summing over t

$$\begin{aligned}\ln\frac{W_{T+1}}{W_1} \le (\eta) \cdot \sum_{t=1}^{T}\sum_{i=1}^{N} q_i(t) \cdot \hat{y}_i(t)\\ +(e-2) \cdot \eta^2 \cdot \sum_{t=1}^{T}\sum_{i=1}^{N} q_i(t) \cdot \hat{y}_i(t)^2\end{aligned}$$

And for any expert k,

$$\ln\frac{W_{T+1}}{W_1} \ge \ln\frac{w_k(T+1)}{W_1} = \eta \cdot \sum_{t=1}^{T} \hat{y}_i(t) - \ln(N)$$

then

$$\begin{aligned}\sum_{t=1}^{T}\sum_{i=1}^{N} q_i(t)\hat{y}_i(t) \ge \sum_{t=1}^{T} \hat{y}_k(t) - \frac{\ln N}{\eta}\\ - (e-2)\sum_{t=1}^{T}\sum_{i=1}^{N} q_i(t)\hat{y}_i(t)^2\end{aligned}$$

Because $p_j(t) = \sum_{i=1}^{N} \frac{w_i \cdot \xi_j^i(t)}{W_t}$, we could have $\sum_{i=1}^{N} q_i(t)\hat{y}_i(t)^2 \le r_t(a_t)$ and $\sum_{i=1}^{N} q_i(t)\hat{y}_i(t) \le \hat{r}_t(a_t)$.

The accumulated reward

$$\begin{aligned}G &= \sum_{t=1}^{T} \hat{r}_t(a_t)\\ &\ge \sum_{t=1}^{T} \hat{y_k}(t) - \frac{\ln N}{\eta} - (e-2)\sum_{t=1}^{T}\sum_{j=1}^{K} \hat{r}_t(j)\end{aligned}$$

Then, taking the expectation

$$E[G] \ge G_{max} - \frac{\ln N}{\eta} - (e-2) \cdot \eta \cdot G_{max} \cdot K$$

and

$$G_{max} - E[G] \le \frac{\ln N}{\eta} + (e-2) \cdot \eta \cdot K \cdot G_{max}$$

And to minimize this, we take the derivative with respect to η and set it to 0, yielding

$$\eta^* = \sqrt{\frac{\ln N}{(e-2) \cdot K \cdot G_{max}}} > 0$$

and

$$G_{max} - E[G] \le 2\sqrt{(e-2)\ln N \cdot K \cdot G_{max}}$$

Since that $G_{max} \le T$, then $R_T = O(\sqrt{TK\ln N})$.

References

[Argall *et al.*, 2009] Brenna D Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. A survey of robot learning from demonstration. *Robotics and autonomous systems*, 57(5):469–483, 2009.

[Auer *et al.*, 2002] Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. The nonstochastic multiarmed bandit problem. *SIAM journal on computing*, 32(1):48–77, 2002.

[Breiman, 1996] Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

[Brys *et al.*, 2015] Tim Brys, Anna Harutyunyan, Halit Bener Suay, Sonia Chernova, Matthew E Taylor, and Ann Nowé. Reinforcement learning from demonstration through shaping. In *IJCAI*, pages 3352–3358, 2015.

[Chemali and Lazaric, 2015] Jessica Chemali and Alessandro Lazaric. Direct policy iteration with demonstrations. In *IJCAI*, pages 3380–3386, 2015.

[Chernova and Veloso, 2007] Sonia Chernova and Manuela Veloso. Confidence-based policy learning from demonstration using gaussian mixture models. In *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, page 233. ACM, 2007.

[Cobo *et al.*, 2011] Luis C Cobo, Peng Zang, Charles L Isbell Jr, and Andrea L Thomaz. Automatic state abstraction from demonstration. In *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, volume 22, page 1243, 2011.

[Fernández and Veloso, 2006] Fernando Fernández and Manuela Veloso. Probabilistic policy reuse in a reinforcement learning agent. In *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, pages 720–727. ACM, 2006.

[François-Lavet *et al.*, 2015] Vincent François-Lavet, Raphael Fonteneau, and Damien Ernst. How to discount deep reinforcement learning: Towards new dynamic strategies. *arXiv preprint arXiv:1512.02011*, 2015.

[Friedrich and Dillmann, 1995] H Friedrich and R Dillmann. Obtaining good performance from a bad teacher. In *Workshop: Programming by Demonstration vs Learning from Examples; International Conference on Machine Learning*, 1995.

[Geramifard *et al.*, 2013] Alborz Geramifard, Robert H Klein, Christoph Dann, William Dabney, and Jonathan P How. RLPy: The Reinforcement Learning Library for Education and Research. http://acl.mit.edu/RLPy, 2013.

[Karakovskiy and Togelius, 2012] Sergey Karakovskiy and Julian Togelius. The mario ai benchmark and competitions. *IEEE Transactions on Computational Intelligence and AI in Games*, 4(1):55–67, 2012.

[Kim *et al.*, 2013] Beomjoon Kim, Amir-massoud Farahmand, Joelle Pineau, and Doina Precup. Learning from limited demonstrations. In *Advances in Neural Information Processing Systems*, pages 2859–2867, 2013.

[Konidaris *et al.*, 2010] George Konidaris, Scott Kuindersma, Roderic Grupen, and Andrew G Barto. Constructing skill trees for reinforcement learning agents from demonstration trajectories. In *Advances in neural information processing systems*, pages 1162–1170, 2010.

[Li *et al.*, 2010] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 661–670. ACM, 2010.

[Lu *et al.*, 2010] Tyler Lu, Dávid Pál, and Martin Pál. Contextual multi-armed bandits. In *Proceedings of the Thirteenth international conference on Artificial Intelligence and Statistics*, pages 485–492, 2010.

[Matikainen *et al.*, 2013] Pyry Matikainen, P Michael Furlong, Rahul Sukthankar, and Martial Hebert. Multi-armed recommendation bandits for selecting state machine policies for robotic systems. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 4545–4551. IEEE, 2013.

[Omidshafiei *et al.*, 2018] Shayegan Omidshafiei, Dong-Ki Kim, Miao Liu, Gerald Tesauro, Matthew Riemer, Christopher Amato, Murray Campbell, and Jonathan P How. Learning to teach in cooperative multiagent reinforcement learning. *arXiv preprint arXiv:1805.07830*, 2018.

[Piot *et al.*, 2014] Bilal Piot, Matthieu Geist, and Olivier Pietquin. Boosted bellman residual minimization handling expert demonstrations. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 549–564. Springer, 2014.

[Suay *et al.*, 2016] Halit Bener Suay, Tim Brys, Matthew E Taylor, and Sonia Chernova. Learning from demonstration for shaping through inverse reinforcement learning. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*, pages 429–437. International Foundation for Autonomous Agents and Multiagent Systems, 2016.

[Sutton and Barto, 1998] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

[Taylor and Stone, 2007] Matthew E. Taylor and Peter Stone. Cross-Domain Transfer for Reinforcement Learning. In *Proceedings of the Twenty-Fourth International Conference on Machine Learning (ICML)*, June 2007.

[Taylor *et al.*, 2011] Matthew E Taylor, Halit Bener Suay, and Sonia Chernova. Integrating reinforcement learning with human demonstrations of varying ability. In *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pages 617–624. International Foundation for Autonomous Agents and Multiagent Systems, 2011.

[Wang and Taylor, 2017] Zhaodong Wang and Matthew E Taylor. Improving reinforcement learning with confidence-based demonstrations. In *Proceedings of the 26th International Conference on Artificial Intelligence (IJCAI)*, 2017.

[Watkins and Dayan, 1992] Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8(3-4):279–292, 1992.

[Yosinski *et al.*, 2014] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Advances in neural information processing systems*, pages 3320–3328, 2014.

[Zhou, 2015] Li Zhou. A survey on contextual multi-armed bandits. *arXiv preprint arXiv:1508.03326*, 2015.