Execution Skill Estimation

Extended Abstract

Christopher Archibald
Mississippi State University
archibald@cse.msstate.edu

Delma Nieves-Rivera
Mississippi State University
din7@msstate.edu

ABSTRACT

In domains with continuous action spaces, one characteristic of an agent is their precision in executing intended actions. An agent's execution skill significantly impacts their success as it determines how much executed actions deviate from intended actions. We introduce the problem of estimating an agent's execution skill level given only observations of their executed actions. The main difficulty is that while executed actions are observed, the intended actions are not, thus the amount of action deviation due to imperfect execution skill is not obvious. We introduce a simple experimental domain in which this problem can be studied and present a method that focuses on observed rewards to estimate execution skill. This method is experimentally evaluated and shown be able to estimate an agent's execution skill under certain conditions.

KEYWORDS

Execution Uncertainty; Opponent Modeling; Parameter Estimation

ACM Reference Format:
Christopher Archibald and Delma Nieves-Rivera. 2018. Execution Skill Estimation. In *Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), Stockholm, Sweden, July 10–15, 2018,* IFAAMAS, 3 pages.

1 INTRODUCTION

Many decision-making settings require agents to select actions from a continuous action space. Examples of this can be found in robotics and in other real-world settings like golf, billiards [2, 4, 13], and curling [1, 17], the latter two of which have been investigated in recent years as challenging domains for computer agents.

In domains with continuous action spaces, it is typically unrealistic to assume that an agent has the ability to perfectly execute a planned action. In computer billiards and curling, as examples, noise is added to actions before they are executed in a deterministic simulator. Dealing with imperfectly executed actions is one of the challenges faced by computational agents in these domains.

This execution uncertainty has been identified as a unique aspect of settings with continuous action spaces, where it has been called an agent's execution skill [3]. Execution skill can be viewed as a property of an agent that potentially differs between agents. It stands in contrast to the notion of *strategic*, or *planning skill*, which refers to an agent's ability to select a quality action for execution, given understanding of their own execution skill level. Furthermore, it can have a large impact on the success of an agent [3] and thus, knowledge of this attribute is vital when an agent's performance must be predicted. This is especially important in game settings, like billiards or curling, where knowledge of an agent's execution skill level can impact the strategies selected by an opponent.

Execution uncertainty has been explored in several settings, including auctions [16], general games [5, 9], and security games [12, 18]. The area of opponent modeling, also has a rich history, especially in imperfect information games like poker [6–8, 11], but this work focuses on strategic characteristics and limitations of the opponents and there is no execution uncertainty. Opponent modeling has also been done in general multi-agent systems [10], real-time strategy games [14], and n-player games [15].

We introduce the problem of estimating an agent's execution skill level given only observations of their executed actions. This problem would be trivial *if* observations included the intended actions of an agent. In general, this information is not accessible to outside observers and so any determinations of an agent's execution skill level must be based solely on the final "noisy" executed action. We propose a method for estimating execution skill from observations and evaluate such in a simple experimental domain. Results demonstrate that it is possible to estimate an agent's execution skill level under certain conditions.

2 PROBLEM DEFINITION

We utilize traditional Markov Decision Processes (MDPs) to model domains with continuous action spaces where agents have an imperfect ability to execute intended actions, requiring only that the set of actions must be a compact subset of $\mathbb{R}^m$. An *agent* possesses strategic skill and execution skill. Strategic skill refers to the action selection method π which specifies the *planned* or *intended* action for a given state. The execution skill refers to a distribution over random perturbations. A sample from this distribution is added to each attempted action before it is executed in a state. We assume that an agent's execution skill distribution is independent of both the state and the planned action, as this noise is meant to model uncertainties and imperfections in the agent over which the agent has no control. We also assume that such distribution is fully known to the planning component of an agent, allowing it to be considered zero-mean without loss of generality. With these assumptions, the main property of interest regarding an agent's execution skill is its standard deviation, which will be referred to as σ.

Let an observation consist of a tuple (s, a, r, s') which specifies a state s, the action a that was *actually executed* in s, the subsequent reward r and the next state s' that resulted from executing a in s. The *execution skill estimation problem* is: given a set of observations of an agent acting in an environment, can the agent's execution skill parameter σ be determined?

The main difficulty is due to only observing the executed action, not the intended one. The interaction between the strategic skill

Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), M. Dastani, G. Sukthankar, E. André, S. Koenig (eds.), July 10–15, 2018, Stockholm, Sweden.

and execution skill makes it challenging in general to estimate each component independently. As an example, consider the case where the agent has minimal strategic skill and utilizes a uniform random strategy π. In such a setting it is impossible to determine, simply by observing executed actions, the amount of noise that results from the agent's imperfect ability to execute actions. We focus on a constrained version of the *execution skill estimation problem* with only perfectly rational planning agents, i.e. agents for whom $\pi(s) = a^*$ results in an optimal action in each state. If π for an agent is rational in this way, can σ be accurately estimated?

If execution skill cannot be accurately estimated for an agent in this circumstance, then little hope lies in being able to complete the task when an agent has less planning acumen. The hope is that a successful method for the rational case will give insight and direction for future attempts in determining σ for agents whose π is of bounded or limited rationality.

3 EXPERIMENTAL DOMAIN

In order to experimentally investigate execution skill estimation methods, we introduce the simple *one-dimensional darts* or 1D-darts, domain. Random 1D-darts problem instances were created and used for experimental validation. Each 1D-darts problem instance consists of the same single state and a continuous action space $A = [-10, 10]$. The reward function differs for each instance and is defined by a sorted list of real values in which the reward function alternates between 0 and 1. For example, given the list $(-8, -1, 3, 7)$, the reward will be 0 on intervals $\{[-10, -8), [-1, 3), [7, 10)\}$ and 1 on $\{[-8, -1), [3, 7)\}$. Every reward-defining list has an even number of points, so the action space is bounded at each end by a reward interval with a reward of 0, allowing the action space to be wrapped. Given a 1D-darts reward function an agent must select an action which will have noise added to it from the agent's execution noise distribution. The agent then receives the reward corresponding to the interval in which the executed action lies. Each intended action has an associated expected reward with respect to the agent's execution noise distribution. The assumption of perfectly rational strategic skill means the selected action will always maximizes this expected reward.

4 ESTIMATION METHODS

How can we estimate an agent's execution skill level, given observations that exclude the *intended* action? If the intended action were included with observation t, the true noise value ϵ_t could be determined. σ could then be estimated directly as the sample standard deviation of the observed ϵ values. This method, the *True Noise Method* (*TN*), cannot be used in practice as it relies on unavailable information, but is introduced as a baseline for comparison.

We now introduce the *Observed Reward Method* (*OR*), which focuses on the reward received by the agent as part of each observation. This is a single sample of the mean reward the agent would receive from that state. Given knowledge of the structure of the state and reward function, we can determine, for different possible execution skill levels σ^i, the maximum expected reward (MER^i) that an agent with that execution skill level could receive in the observed state. As observations are processed at each time step t, the mean maximum expected reward ($MMER^i$) for each hypothesis

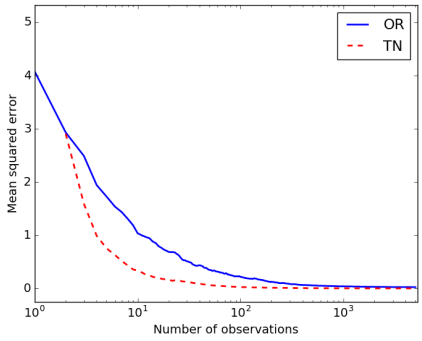

Figure 1: Method comparison (N = 320)

σ^i can be computed as the sample mean of MER^i for that σ^i across all observed states. As actual rewards obtained by the agent are observed, the sample mean of the observed rewards can be similarly computed. This estimate is called the mean observed reward MOR_t. At time step t, the OR method predicts σ using MOR_t and all of the $MMER_t^i$ values. This is done using linear interpolation on the $MMER_t^i$ values to obtain a prediction $\tilde{\sigma}_{OR}$ of the execution skill level that would result in observed rewards MOR_t.

Under our current assumption that the acting agent is perfectly rational, then this is equivalent to estimating the mean maximum expected reward from a set of samples drawn from the true distribution. This estimate will converge to the true mean maximum expected reward by the law of large numbers as the number of observations approaches infinity. Thus, the prediction of this method will also converge to the correct value, limited only by the resolution of our set of hypothesis execution skill levels.

5 EXPERIMENTS AND DISCUSSION

Experiments were carried out in the 1D-darts domain, using zero-mean Gaussian noise for the execution noise distribution. The standard deviation of this is randomly generated between 0.25 and 4.5. The rational intended action for a state was computed as the argmax of the convolution of the execution noise distribution with the reward function, using a resolution of 0.001.

The OR method used 10 hypothesis σ_i values and computed the expected reward of the rational action for each in each state. The TN method's estimate was also computed as described. Each method produced an estimate after each new observation, in an online manner, and the squared error of these estimates was computed. This squared error was averaged at each time step over all the experiments to give the mean squared error (MSE) for each method.

Figure 1 shows performance curves over 5000 observations, averaged for 320 experiments. As the number of observations increases, the OR method slowly converges, due to the increased variance from only seeing a single sample of each MER value. The TN method is more accurate with fewer observations, but the OR method eventually converges to the same correct value. Thus, the OR method is a feasible solution to the execution skill estimation problem.

In the future, we plan to explore other methods for this problem without rationality assumptions and in new domains.

REFERENCES

[1] Zaheen Farraz Ahmad, Robert C Holte, and Michael Bowling. 2016. Action Selection for Hammer Shots in Curling.. In *IJCAI*. 561–567.

[2] Christopher Archibald, Alon Altman, Michael Greenspan, and Yoav Shoham. 2010. Computational Pool: A new challenge for game theory pragmatics. *AI Magazine* 31, 4 (2010), 33–41.

[3] Christopher Archibald, Alon Altman, and Yoav Shoham. 2010. Success, strategy and skill: an experimental study. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 1089–1096.

[4] Christopher Archibald, Alon Altman, and Yoav Shoham. 2016. A Distributed Agent for Computational Pool. *IEEE Transactions on Computational Intelligence and AI in Games* 8, 2 (June 2016), 190–202. https://doi.org/10.1109/TCIAIG.2016.2549748

[5] Christopher Archibald and Yoav Shoham. 2011. Hustling in repeated zero-sum games with imperfect execution. In *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, Vol. 22. 31–36.

[6] Nolan Bard, Michael Johanson, Neil Burch, and Michael Bowling. 2013. Online Implicit Agent Modelling. In *Proceedings of the Twelfth International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*. 255–262.

[7] Nolan Bard, Deon Nicholas, Csaba Szepesvari, and Michael Bowling. 2015. Decision-theoretic Clustering of Strategies. In *Proceedings of the Fourteenth International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*. To Appear.

[8] Darse Billings, Denis Papp, Jonathan Schaeffer, and Duane Szafron. 1998. Opponent modeling in poker. In *AAAI/IAAI*. 493–499.

[9] Michael Bowling and Manuela Veloso. 2004. Existence of Multiagent Equilibria with Limited Agents. *Journal of Artificial Intelligence Research* 22 (2004), 353–384. A previous version appeared as a CMU Technical Report, CMU-CS-02-104.

[10] David Carmel and Shaul Markovitch. 1995. Opponent modeling in multi-agent systems. In *International Joint Conference on Artificial Intelligence*. Springer, 40–52.

[11] Trevor Davis, Neil Burch, and Michael Bowling. 2014. Using Response Functions to Measure Strategy Strength. In *Proceedings of the Twenty-Eighth Conference on Artificial Intelligence (AAAI)*. 630–636.

[12] Albert Xin Jiang, Zhengyu Yin, Chao Zhang, Milind Tambe, and Sarit Kraus. 2013. Game-theoretic randomization for security patrolling with dynamic execution uncertainty. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*. International Foundation for Autonomous Agents and Multiagent Systems, 207–214.

[13] J. F. Landry, J. P. Dussault, and E. Beaudry. 2015. A Straight Approach to Planning for 14.1 Billiards. *IEEE Transactions on Computational Intelligence and AI in Games* PP, 99 (2015), 1–1. https://doi.org/10.1109/TCIAIG.2015.2462335

[14] Frederik Schadd, Sander Bakkes, and Pieter Spronck. 2007. Opponent Modeling in Real-Time Strategy Games.. In *GAMEON*. 61–70.

[15] Nathan Sturtevant, Martin Zinkevich, and Michael Bowling. 2006. ProbMaxn: Opponent Modeling in N-Player Games. In *Proceedings of the Twenty-First National Conference on Artificial Intelligence (AAAI)*. 1057–1063.

[16] Gert Van Valkenhoef, Sarvapali D Ramchurn, Perukrishnen Vytelingum, Nicholas R Jennings, and Rineke Verbrugge. 2010. Continuous double auctions with execution uncertainty. In *Agent-Mediated Electronic Commerce. Designing Trading Strategies and Mechanisms for Electronic Markets*. Springer, 226–241.

[17] Timothy Yee, Viliam Lisỳ, and Michael H Bowling. 2016. Monte Carlo Tree Search in Continuous Action Spaces with Execution Uncertainty.. In *IJCAI*. 690–697.

[18] Zhengyu Yin, Manish Jain, Milind Tambe, and Fernando Ordónez. 2011. Risk-Averse Strategies for Security Games with Execution and Observational Uncertainty.. In *AAAI*.