Approximating Difference Evaluations with Local Knowledge

(Extended Abstract)

Mitchell Colby
Oregon State University
colbym@engr.orst.edu

William Curran
Oregon State University
curranw@engr.orst.edu

Carrie Rebhuhn
Oregon State University
rebhuhnc@engr.orst.edu

Kagan Tumer
Oregon State University
kagan.tumer@oregonstate.edu

ABSTRACT

Difference evaluation functions have resulted in excellent multiagent behavior in many domains, including air traffic control and distributed sensor network control. In addition to empirical evidence, there is theoretical evidence that suggests difference evaluation functions help shape private agent utilities/objectives in order to promote coordination on a system-wide level. However, calculating difference evaluation functions requires global knowledge about the system state and joint action as well as the mathematical form of the system objective function, which are often unavailable. In this work, we demonstrate that a local estimate of the system evaluation function may be used to locally compute difference evaluations, allowing for difference evaluations to be computed in multiagent systems where only local state and action information as well as a broadcast value of the system evaluation function are available. We demonstrate that approximating difference evaluation functions results in better performance and faster learning than when using global evaluation functions, and performs only slightly worse than when directly computing difference evaluations.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence —*Multiagent systems*

Keywords

Multiagent learning, difference evaluation functions

1. INTRODUCTION

Difference evaluation functions have been shown to significantly improve learning in multiagent systems, and have produced excellent results in many multiagent settings, including air traffic control and multiple mobile robot control [1, 2]. Difference evaluation functions are defined as [1]:

$$D_i(s,a) = G(s,a) - G(s_{-i} + c_{s,i}, a_{-i} + c_{a,i}) \quad (1)$$

Appears in: *Alessio Lomuscio, Paul Scerri, Ana Bazzan, and Michael Huhns (eds.), Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2014), May 5-9, 2014, Paris, France.*

where $G(s,a)$ is the system evaluation function, s is the system state, a is the joint action, s_{-i} is the system state excluding agent i, a_{-i} is the joint action excluding agent i, and $c_{s,i}$ and $c_{a,i}$ are *counterfactual* terms used to replace the state and action of agent i, respectively. Intuitively, the difference evaluation function determines the impact that agent i has on the system evaluation function. Difference evaluations have two key properties that lead to their effectiveness. First, they are aligned with the system evaluation function, meaning an agent which increases $D_i(s,a)$ also increases $G(s,a)$. Second, as the last term in $D_i(s,a)$ removes portions of $G(s,a)$ which aren't impacted by agent i, difference evaluations provide a favorable signal-to-noise ratio in the learning feedback signal, allowing for agents to more easily discern the effects of their actions.

Although difference evaluations provide excellent learned performance, they are often difficult to compute in practice. In order to compute the second term of $D_i(s,a)$, the global state and joint action must be known, as well as the mathematical form of $G(s,a)$. In practice, such knowledge is typically unavailable to learning agents, making direct computation of $D_i(s,a)$ impractical. In order to allow for the implementation of difference evaluations, they must be approximated when global knowledge is unavailable. Difference evaluations have been approximated in past work [3], but this approximation required expert domain knowledge, and thus did not address the key factors requiring the approximation of difference evaluations.

2. DOMAIN AND APPROACH

In order to approximate difference evaluation functions, we assume that each agent has access to its local state and action, as well as a broadcast value of $G(s,a)$. This information is typically available in any multiagent learning system, as some type of global evaluation function is used to provide feedback to the system. At each time step, each agent record its local state s_i and action a_i, as well as the broadcast value of the system evaluation function $G(s,a)$. Each agent maintains a local approximation $\hat{G}(s_i,a_i)$, and uses the $(s_i,a_i,G(s,a))$ tuple to update the approximation. The approximate difference evaluation function is defined as:

$$\hat{D}_i(s,a) = G(s,a) - \hat{G}(c_{s,i}, c_{a,i}) \quad (2)$$

This approximation only requires local state and action in-

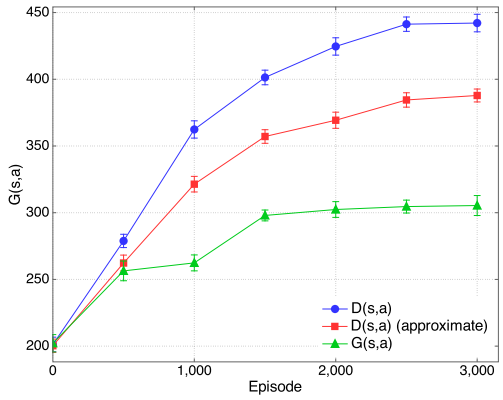

Figure 1: Rover domain results (10 agents). Approximating $D_i(s,a)$ results in 88% of the performance attained when analytically computing $D_i(s,a)$. Approximating the difference evaluation function results in significant performance gains when compared to using the system evaluation function $G(s,a)$.

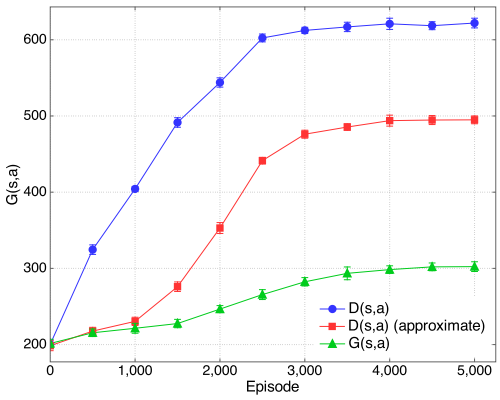

Figure 2: Rover domain results (100 agents). Approximating $D_i(s,a)$ results in 79% of the performance attained when analytically computing $D_i(s,a)$. Approximating the difference evaluation function results in significant performance gains when compared to using the system evaluation function $G(s,a)$.

formation, as well as the broadcast value of the system evaluation function. This information is typically available in any multiagent learning system.

This approximation approach is tested in a multiagent rover domain (See [2] for details on implementation), where a set of rovers move in a planar world in order to observe points of interest. Agents are trained using a cooperative coevolutionary algorithm, and fitness values are assigned with either $G(s,a)$, $D_i(s,a)$, or $\hat{D}_i(s,a)$.

3. RESULTS

The rover domain experiments were initialized as follows. For the first experiment, there are 10 agents and 10 points of interest in a 25 by 25 unit planar world. For coevolution, each agent maintains a population of 25 neural network policies. Each episode lasts 25 timesteps, and the coevolutionary algorithm is allowed to run for 3000 generations. 150 statistical runs were conducted, and reported error bars represent error in the mean. For the second experiment, there are 100 agents and 100 points of interest in a 50 by 50 unit planar world, and learning proceeds for 5000 generations. Other parameters are identical to the first experiment.

Results for the 10 agent rover domain are shown in Figure 1. Approximating $D_i(s,a)$ results in 23% better performance compared to $G(s,a)$, and achieves 88% of the performance when analytically computing $D(s,a)$. Although $\hat{D}_i(s,a)$ results in 12% lower performance than $D(s,a)$, it requires 90% less information to compute, demonstrating the approximation is effectively utilizing locally available information.

Results for the 100 agent rover domain are shown in Figure 2. $\hat{D}_i(s,a)$ results in 49% better performance than $G(s,a)$, and achieves 79% of the performance of analytically computing $D_i(s,a)$. It is of note that in this larger domain, although $\hat{D}_i(s,a)$ performs worse compared to $D_i(s,a)$ (79% vs. 88%), it outperforms $G(s,a)$ by a wider margin (49% vs 23%). Additionally, in this larger domain, $\hat{D}_i(s,a)$ requires even less information than $D_i(s,a)$ compared to the 10 agent domain (99% less vs. 90% less). This demonstrates that $\hat{D}_i(s,a)$ scales well with the number of agents in the system.

4. DISCUSSION

Although difference evaluation functions have produced excellent results in many multiagent settings, their requirement for global state and action information makes them difficult to compute in practice. The contribution of this work is to demonstrate that agents may approximate difference evaluations requiring only local knowledge. Our results demonstrate that the approximation uses far less information than $D_i(s,a)$ (90-99% less), but still achieves comparable performance (up to 88%). The information requirements for this approximation technique are equivalent to traditional multiagent learning techniques, allowing for the implementation of difference evaluations in any multiagent system where the system evaluation function is broadcast.

5. ACKNOWLEDGEMENTS

This work was partially supported by the NETL under grants DE-FE0012302 and DE-FE0011403.

6. REFERENCES

[1] A. K. Agogino and K. Tumer. Analyzing and Visualizing Multiagent Rewards in Dynamic and Stochastic Environments. *Journal of Autonomous Agents and Multi-Agent Systems*, 17(2):320–338, 2008.

[2] M. Colby and K. Tumer. Shaping Fitness Functions for Coevolving Cooperative Multiagent Systems. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Valencia, Spain, June 2012.

[3] S. Proper and K. Tumer. Modeling Difference Rewards for Multiagent Learning (Extended Abstract). In *Proceedings of the Eleventh International Joint Conference on Autonomous Agents and Multiagent Systems*, Valencia, Spain, June 2012.