Multiagent Reactive Plan Application Learning in Dynamic Environments

1st Int. Conf. On Autonomous Agents and Multiagent Systems, 2002

Hüseyin Sevay
hsevay@ittc.ku.edu

Costas Tsatsoulis
tsatsoul@ittc.ku.edu

Department of Electrical Engineering and Computer Science
Information & Telecommunication Technology Center
2335 Irving Hill Road
University of Kansas
Lawrence, KS 66045-7612, USA

ABSTRACT

In addition to bottom-up learning approaches, which facilitate emergent policy learning, it also is desirable to have top-down control over learning so that a team of agents can also learn to apply general policies to diverse dynamic situations. We present a multiagent case-based learning methodology to achieve this top-down control. In this methodology, high-level symbolic plans describe policies a team of agents needs to learn to apply to different situations. For each plan whose preconditions match their current team state, agents learn to operationalize that plan. In each training scenario, each agent learns a sequence of actions that implements each step in the given plan such that the entire plan is operationalized under current external conditions. This application knowledge is acquired via searching through a small set of available high-level actions and testing the success of each sequence of actions in the situated environment. Similarity between a new situation and existing cases is measured by considering only the state internal to the team, and an agent stores the successful sequence of actions in the current plan step indexed under the current external state. By repeating this process for each plan step using many diverse training scenarios, a team of agents learns how to operationalize an entire plan in a wide variety of external situations, hence achieving generality. We demonstrate our approach using the RoboCup soccer simulator.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning—*case-based reactive plan learning in continuous domains*; I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*intelligent agents, multiagent systems, RoboCup soccer*

General Terms

Theory, Algorithms

1. INTRODUCTION

Due to their continuous nature, realistic dynamic environments pose steep challenges for multiagent learning, where agents need to cooperate with each other to achieve common goals by acting autonomously based on their local view of the world. With continuous state and action spaces, the search problem becomes intractable. Therefore continuous search spaces must be either discretized or generalized using function approximation to make them tractable [1, 2, 3].

Reinforcement Learning approaches are bottom-up, because they build emergent solutions by combining a series of actions without any high-level guidance of the policy search process. However, in some learning problems, it may be desirable to have agents learn certain general strategies in real-world games such as soccer and basketball, thus helping constrain the search space from top-down.

To address these issues, we developed a multiagent learning method that combines case-based learning with search. In our method, agents learn to operationalize the individual steps of general symbolic plans that they retrieve from an existing plan library. Plans represent high-level strategies that we wish agents to learn to apply in situated scenarios. Therefore plans initially contain only *preconditions* and *postconditions* for each step but no *application knowledge*. Hence the task of the learning component is to acquire the missing application knowledge during the training phase.

At the high-level, each agent acquires and retrieves application knowledge using case-based learning. An agent stores how a particular plan step can be successfully implemented in a specific situation in a *case*. Then, for each step, an agent retains a set of distinct cases to represent a variety of situations that can be encountered. When the application of a previously acquired case fails during training, an agent learns a new case that handles that failure situation.

At the low-level, each agent searches through a small set of available high-level actions and executes them on a trial-and-error basis to discover a short sequence of actions that successfully implements each plan step in a collaborative fashion with other agents in the same team such that the overall plan can also be successfully applied to the current situation.

By repeated training under different randomly-generated scenarios, agents augment and refine their individual case-bases with condition vectors that indicate potential future success and potential future failure coupled with a list of specific actions that have been shown to succeed previously under similar circumstances.

2. LEARNING METHODOLOGY

To deal with the high-dimensionality of a dynamic domain, where multiple agents can affect the state of the world simultaneously, we distinguish between those conditions that a multiagent system can exercise some control over (*internal* conditions) and those conditions that the multiagent system cannot directly affect or easily deduce (*external* conditions)

The reason for this distinction is to enable a multiagent system to identify a plan to apply in the current situation without reasoning about the details of the external situation dynamics and subsequently employ learning to operationalize the application of each step in a plan, once a candidate plan is selected.

A multiagent system first considers its own internal state to determine a ballpark plan to apply to the current situation. Then it considers the external conditions to choose which particular case to apply to the current problem. Therefore, we hypothesize that, if the multiagent system can learn how to handle the specializations of the external state, it can behave successfully in the dynamic environment in which it is situated by applying a sequence of high-level multiagent plans or policies that it has learned to operationalize.

Each plan in our system may consist of several steps. Each step is described at a high-level and not at the level of the primitive actions possible in the target domain. Therefore plans are to guide behavior rather than to dictate all runtime details, which can potentially have infinite variations in a continuous domain like soccer.

Before any learning takes place, plans contain *preconditions* and *postconditions* but no *application knowledge*. Since agents can encounter a wide variety of situations, the handling of runtime variations, that is, the learning of what actions to take in each situation, is left to the low-level learning component of each agent.

During training, each agent learns a sequence of actions and the external conditions under which a plan step was successful so it can predict the success of that step in *similar* cases. In addition, it learns the external conditions under which a plan step failed and how to handle that failure so that the team can successfully apply the entire plan.

When a case is initially learned, it is tagged as *successful*, since it has been successfully used. When that same case is retrieved during later training episodes and used successfully, that case will be reinforced as useful. However, when a previously successful case fails, the agent will learn a new case to handle the failure, and this new case will be tagged as *failed*. If a previously failed case is retrieved but used successfully, then a copy of that case will be made, and this copy will be tagged as *successful*. If a previously failed case fails in a similar situation, a new case is learned, and the new case is again tagged as *failed*.

Thus the successful or failed application of existing cases enables agents to increase the *resolution* of their casebases. Therefore, instead of having one case to handle a group of potentially similar situations for a plan step, an agent can choose from multiple cases. Therefore, an agent will be able to explore a potentially larger external state space using the features inherent in the stored indices of each case, even though the similarity metric would treat all these cases as similar. Moreover, if learning converges, the consistently successful cases will be apparent in each case because of the reinforcement they will have received.

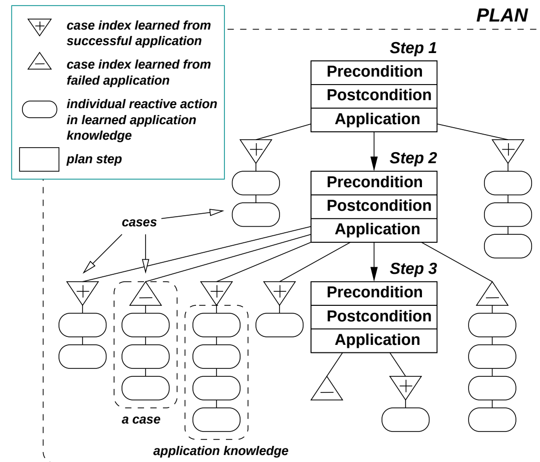

Figure 1: A plan with learned cases

Figure 1 shows an example 3-step plan. In addition to the *preconditions* and *postconditions*, each plan has cases it learned containing the *application knowledge* that implements each plan step in a variety of situations. The cases that have been applied successfully are indicated with a (+) sign, and the cases that have been learned as a result of a failure are indicated with a (−) sign.

Each case has two important parts: case retrieval index, and *application knowledge*. The retrieval index (tagged with (+) or (−)) is the *external condition* vector of the situation that has been stored during learning. The application knowledge is a series of individual reactive actions that, when applied in sequence, implements a plan step.

However, in some cases, a desirable action sequence may not be found at all as indicated in *step 3*. Such a case would at least indicate that the learning parameters may have to be adjusted and training repeated.

In our approach, agents start with a small set of hand-designed multiagent plans without any application knowledge. Agents choose a plan to apply by finding a match to current *internal conditions* in the precondition of the very first plan step in one of the general plans in the plan library. The choice of which particular application knowledge to use to implement each plan step then depends only on the *external conditions* present at each plan step application. Then, by applying a series of such plans, a group of agents can exhibit overall team-level behavior.

Our approach has been implemented in RoboSoccer and is being tested.

3. REFERENCES

[1] J. Forbes and D. Andre. Practical reinforcement learning in continuous domains. Tech. report, Comp. Sci. Division (EECS), University of California, Berkeley, CA 94720, Aug. 2000. UCB/CSD-00-1109.

[2] M. J. Matarić. Learning in multi–robot systems. *Lecture Notes in Artificial Intelligence (Adaptation and Learning in Multi–Agent Systems)*, 1042:152–163, 1996.

[3] P. Stone and R. S. Sutton. Scaling reinforcement learning toward robocup soccer. In *ICML-2001*, pages 537–544, Williams College, MA, USA, 2001.