Appears in the Proceedings of the Third International Conference on Knowledge Capture (K-CAP 2005), Banff, Canada.

Knowledge Transfer Via Advice Taking

Lisa Torrey, Trevor Walker, Jude Shavlik
University of Wisconsin - Madison
1210 West Dayton Street
Madison, WI 53706 USA
{ltorrey,twalker,shavlik}@cs.wisc.edu

Richard Maclin
University of Minnesota – Duluth
1114 Kirby Drive
Duluth, MN 55811 USA
rmaclin@d.umn.edu

ABSTRACT

We present a framework for knowledge transfer from one reinforcement learning task to a related task through advice-taking mechanisms. We discuss the importance of transfer in complex domains such as RoboCup soccer, and show how to use automatically generated advice to perform transfer.

Categories and Subject Descriptors

I.2 [**Artificial Intelligence**]: Learning

General Terms

Algorithms, Experimentation

Keywords

Advice, Knowledge Transfer, Reinforcement Learning

INTRODUCTION

When a new reinforcement learning task arises, it is likely that human experts will be able to provide information about how the new task relates to other tasks in the domain. By identifying parallels between the features and actions of the new task and those of a known task, a human expert could help the learning algorithm transfer knowledge from the known task to the new one. In addition, this expert may also have direct tips on how to accomplish the new task. We call these forms of human assistance *advice*; they may not be complete or fully accurate, but they are likely to provide valuable information that can speed up learning.

For example, suppose a machine learner has learned to play the RoboCup soccer subtask *KeepAway* [3] , where a team of N robotic players try to keep the ball away from an opposing team of $N - 1$ players. Next suppose that the learner must learn to play the related RoboCup soccer subtask *BreakAway* [2], where a team of N robotic players try to score a goal against an opposing team of $N - 2$ defenders and a goalie. A human advisor could identify parallels such as:

- Features representing distances between players and angles among players are analogous.
- Actions for passing to teammates are analogous.
- The new action "shoot at goal" is analogous to the old action "pass to teammate" if you pretend that a teammate is standing in the goal.

The human advisor could also give direct tips about BreakAway, such as:

IF *distance to goal* < 10 AND *shot angle* > 30
THEN PREFER *shoot* TO *pass*

Both of these forms of advice are easy for a human to provide, and the learner could use them to learn to play BreakAway more quickly.

ADVICE-TAKING AND TRANSFER

To transfer knowledge between reinforcement learning tasks, we use advice in several ways as summarized in Figure 1.

We developed an algorithm to accept direct advice about action preferences in previous work [2]. This advice looks like the IF-THEN rule above, and corresponds to the optional advice for Task B in Figure 1. Such advice could also be provided for Task A before transfer. Note that it does not require the advisor to know anything about the internal details of the learning algorithm, such as *Q*-values of actions; the advice is expressed in terms of features and actions only. We have shown that advice like this can substantially improve performance in BreakAway [2].

In recent work [4], we developed an algorithm to accept advice about parallels between tasks. This advice communicates information like the feature and action parallels above, and corresponds to the circle in Figure 1. We apply it to a learned model for Task A and automatically extract *transfer advice* for Task B. This transfer advice has the same form as the direct advice above. Note that this means we do not have to worry if the ranges of the *Q*-values in the two tasks are different; the advice only specifies action preferences, rather than action *Q*-values.

The human advisor identifies parallels by providing a *mapping* that translates features and actions in the old task to

K-CAP'05, October 2–5, 2005, Banff, Alberta, Canada.

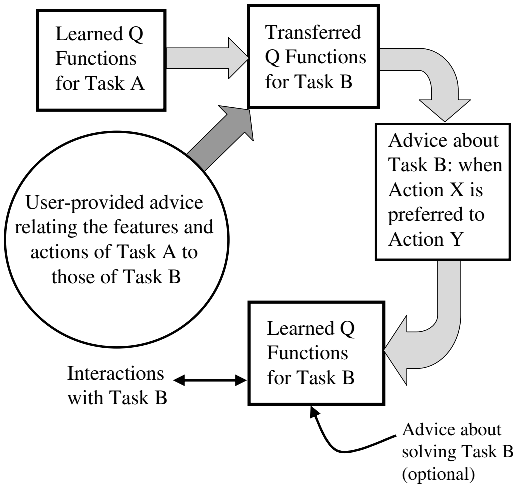
Learned Q Functions for Task A
Transferred Q Functions for Task B
Advice about Task B: when Action X is preferred to Action Y
User-provided advice relating the features and actions of Task A to those of Task B
Learned Q Functions for Task B
Interactions with Task B
Advice about solving Task B (optional)

Figure 1: Transferring knowledge using advice.

features and actions in the new task. Using this mapping, we can evaluate a situation from the perspective of the old task. If one old action would have been better than all the others in this situation, we create transfer advice that recommends taking the corresponding action in the new task. Table 1 outlines this automated process. The function Q'_a is a linear function of features expressing the value of action a in KeepAway, but with the KeepAway features replaced by corresponding BreakAway ones. The action a' is the BreakAway action that corresponds to a.

Here is an example of part of a transfer advice rule extracted for BreakAway. The KeepAway actions are "PassNearest" for passing to the nearest real teammate, "PassGoal" for passing to an imaginary teammate located in the goal, and "HoldBall". The analogous BreakAway actions are "PassNearest", "Shoot", and "MoveAhead".

```
IF      $Q'_{PassGoal} - Q'_{PassNearest} \geq \Delta$   AND
        $Q'_{PassGoal} - Q'_{HoldBall} \geq \Delta$

THEN    PREFER $Shoot$ TO $PassNearest$   AND
        PREFER $Shoot$ TO $MoveAhead$
```

We have recently been investigating other mechanisms for extracting transfer advice. Because our current learned models are linear functions of features, we can place *Q*-function expressions directly into the advice as shown above. However, we would also like to apply transfer advice to non-linear models like neural networks or support vector machines, which do not fit so easily into our advice language.

Table 1: An algorithm to create transfer advice.

```
GIVEN
  A learned model of Task A AND
  A mapping from Task A to Task B
DO
  for each $a \in Actions(TaskA)$ generate advice:
    IF for each $b \neq a \in Actions(TaskA) : Q'_a - Q'_b \geq \Delta$
      THEN PREFER $a'$ TO $b'$ in Task B
```

One approach we are investigating uses Inductive Logic Programming (ILP) [1]. ILP systems learn concepts from training examples in the form of first-order logic rules. With ILP, we learn rules from transcripts of KeepAway games and use those to create advice for BreakAway using the same kind of user-provided mapping as before. The use of first-order logic provides an additional advantage: we can have a variable *?teammate*, for example, that applies to any of the learner's teammates, instead of a particular one as was necessary in the previous transfer rule. This makes it possible to create more general rules.

Here is an example of an ILP rule learned for BreakAway:

```
IF      dist_between_less_than(me,goalie,10) and
        dist_between_greater_than(me,goal_left,10) and
        dist_between_in_range(me,?teammate,3,25)
THEN    PREFER pass_to(?teammate) TO move_ahead(me)
```

We have presented results for the first transfer approach [4] and are currently working on the second.

SUMMARY

Transfer is a critical capability for any system attempting to learn in a complex domain, since learning each new task from scratch would be extremely time-consuming and redundant. We have developed mechanisms for extracting and applying learned knowledge using human-provided advice about the parallels between the old task and the new one. We automatically create transfer advice in an action preference format that we have developed, and we can also incorporate direct advice about the new task in this format.

Advice-taking can be an useful tool for achieving knowledge transfer. It supplies the learner with a partial policy for deciding which actions to take, particularly for situations in which the user thinks that old skills will apply. Also, since advice can be refined or discarded by the learner rather than followed absolutely, transfer advice should not be harmful in the long run even if the user's guidance is imperfect. In future work we plan to examine new transfer methods that use advice.

REFERENCES

[1] N. Lavrac and S. Dzeroski. *Inductive Logic Programming*. Ellis Horwood, New York, 1994.

[2] R. Maclin, J. Shavlik, L. Torrey, T. Walker, and E. Wild. Giving advice about preferred actions to reinforcement learners via knowledge-based kernel regression. In *AAAI*, 2005.

[3] P. Stone and R. Sutton. Scaling reinforcement learning toward RoboCup soccer. In *ICML*, 2001.

[4] L. Torrey, J. Shavlik, T. Walker, and R. Maclin. Using advice to transfer knowledge acquired in one reinforcement learning task to another. In *ECML*, 2005.