Abstraction for Analogical Reasoning in Robotic Agents

Tesca Fitzgerald, Andrea Thomaz, and Ashok Goel

School of Interactive Computing,
Georgia Institute of Technology
tesca.fitzgerald@cc.gatech.edu,
athomaz@ece.utexas.edu,
goel@cc.gatech.edu

Abstract. Analogical reasoning has been implemented in agents to use information about past experiences to guide future action. A similar process for analogical reasoning would enable a robot to reuse past experiences, such as the tasks that an interactive robot could learn from a human teacher. However, in a robotics domain, analogical reasoning must be performed over the knowledge obtained from sensor input and must result in actionable output in the form of the robot's joint configurations. Thus, robotics provides a challenging domain for analogical reasoning, as abstraction plays a necessary role in representing the robot's task knowledge. We explore the problem of abstraction for a robot which performs analogical reasoning, and provide an example task illustrating how abstraction is necessary for a robot to reason over learned tasks.

Keywords: Case-based agents, robotics, process-oriented CBR

1 Introduction

In the context of an agent which learns to execute tasks, analogical reasoning can be used to transfer known task knowledge from a source task to address an unfamiliar, target problem. A similar process can be applied to a robotic agent, where the problem the robot seeks to address is a task environment observed in the real-world, and the task plan the robot executes consists of a series of motor commands. After observing a new target problem, the robot would identify a relevant task plan from memory, using a method such as that described in [2], and then reuse the known task plan to address the target problem. However, the problem of reusing learned actions to address an unfamiliar problem is still a difficult challenge [10]. Actions can be learned in a way such that small adaptations in the task requirements can be made, such as moving objects from their original location in the learned task [6], or to respond to larger changes in the target task by collecting additional experience in the target domain [12, 11]. However, these approaches transfer task knowledge by directly adapting the action representation, and do not address the task at a higher-level of abstraction, such as at the

In Proceedings of the ICCBR 2016 Workshops. Atlanta, Georgia, United States of America

goal-level. This limits the number of target problems in which a task representation can be reused. Case-based approaches have been used to enable action transfer at the strategic level in domains such as RoboCup soccer [4, 3, 5, 7], and in problems requiring case retrieval and adaptation based on both surface features and action goals [9]. However, transferring action strategies assumes that the task knowledge has already been represented at a pre-determined level of abstraction.

By implementing an analogical reasoning approach to task transfer, a robot could reuse task knowledge in a wider range of target problems. However, a challenge of applying analogical reasoning to the robotics domain is that it must operate over knowledge represented at a low-level of abstraction. This problem makes it difficult to identify high-level analogs between a source and target problem, such as similar relationships between the robot's actions and the objects it observes, and then use this analogy to address the target problem. We introduce the challenges of applying analogical reasoning in a robotic agent, and describe preliminary results demonstrating how the level of abstraction at which task knowledge is represented affects how it can be transferred.

2 Problem Domain

We focus on the problem of enabling analogical reasoning for a robot which learns to complete real-world tasks on a tabletop workspace, such as pouring a cup or stacking a set of blocks. We assume that the robot is taught the task by a human teacher who moves the robot's arm through the action of completing the task. This results in a representation of the task containing two forms of input:

- A point-cloud representation of the observed scene, consisting of $p_{x,y} = <r, g, b, d>$ containing the color (in RGB format) and depth of each x, y coordinate in the observed frame.
- A joint-space trajectory recorded during the task demonstration, containing a set of joint configurations $d_t = <j_0, j_1, ..., j_n>$ where j_n is the position of a single joint in the robot's arm at the timestamp t.

Both the source and target environments are perceived via the overhead camera. However, while an action is learned for the source task, the robot does not yet have an action model which can be executed to achieve the task in the target environment. This representation occurs at a *low-level of abstraction* because it does not present any relation between the task goal, the observed objects, and/or the demonstrated action. Thus, the analogs between a source and target problem are not initially apparent, and it becomes essential to abstract the task representation at some level in order to perform analogical reasoning.

3 Abstractions

We approach the problem of analogical reasoning in a robotic domain as one of *correctly abstracting* a task demonstration for each reasoning problem. When the robot receives a demonstration of the task, it is stored in memory in the representation $S =< D, T, O, F >$ where:

- D is the set of *sub-skill models* recorded during the demonstration, where $D =< d_0, ..., d_n >$ and n is the number of sub-skills comprising the task. Each subskill d_i is represented as a Dynamic Movement Primitive (DMP) that is trained from the task demonstration as defined by [8]. This skill representation allows the robot to later reproduce a motion trajectory that is similar to the original demonstration, but with modified starting and ending point locations.
- T is the set of *object relations* that express the spatial offset between the end point location of each skill and the locations of objects observed at the start of the demonstration, e.g. $T =<< {}_x t_0, {}_y t_0, {}_z t_0 >, ..., < {}_x t_n, {}_y t_n, {}_z t_n >>$ for n sub-skills.
- O is the set of observed *object labels*, where $O =< o_0, ..., o_i >$ and o_i lists a single object's identifying label.
- F is the set of observed *object features*, where $F =< f_0, ..., f_i >$ and f_i lists the object features associated with object o_i, such as object location, dimensions, hue, affordances, and properties. These features are derived from the point-cloud representation of the environment as observed from an overhead camera, and are derived using the algorithm described in [13].

We derive several abstractions of this representation, as listed in Table 1. In the first abstraction, most of the demonstration representation is retained, with the exclusion of low-level features (such as object locations and orientation) that are specific to the environment in which the task was learned. The second representation further abstracts the task by removing object labels, thus removing the assumption that all instances of the task will contain the same objects. A result of this abstraction is that the robot must additionally be provided with a mapping between objects in the source and target environments before it can execute the task using this representation. We currently provide this mapping manually; however, in future work, we intend to have the robot predict this object mapping by further interacting with the human teacher. In [1], we describe an approach to object mapping in which the robot interacts with the human teacher to receive object "hints", which are correspondences between an object in the source environment and an object in the target environment. Once the robot has received a mapping hint, it uses this correspondence to infer which features should be used to map the remaining objects [1]. In future work, we plan to integrate this approach to object mapping with our approach to similarity-based task abstraction and transfer.

The third abstraction is the highest-level representation, in which only the sub-skill models are retained. By excluding the object relations element of the demonstrated task, which dictates the relation between the robot's hand and each object during the task, the task constraints which were present in the demonstration are no longer enforced. Thus, a new set of task constraints must be introduced so that the task plan can be executed accordingly. We currently

Demonstration	Sub-skills, Object Relations, Object Labels, Object Features
Abstraction 1	Sub-skills, Object Relations, Object Labels
Abstraction 2	Sub-skills, Object Relations
Abstraction 3	Sub-skills

Table 1: Features Present at Each Level of Similarity

(a) Source Environment (b) Displaced Objects (c) Replaced Objects (d) Replaced Scoop

Fig. 1: Variants of the *Scooping* Task Environment

provide these updated constraints manually, with future work addressing the problem of inferring task constraints. Additionally, as in the second abstraction, an object mapping is necessary before the task can be executed.

4 Experiment

We evaluated the impact of abstraction on variations of a scooping task, shown in Figure 1, in which the target environment differed from the source environment in one of three ways: (i) object displacement, in which all objects were retained but moved to different locations, (ii) object replacement, where new objects were introduced and displaced, and (iii) scoop replacement, which altered the constraints of the task since the robot now needed to keep its hand at a different height to account for the change in scoop length.

4.1 Preliminary Results

After providing a single demonstration of the scooping task to the robot learner, we counted the number of successful task executions performed by the robot in each of ten different target environments when using each of the three abstraction levels. These results are recorded in Table 2. The task could be executed consistently in the displaced-objects environments using any of the three abstractions, since these target environments were most similar to the source environment.

However, in problems where the target environment was more dissimilar from the source environment, an abstracted representation was necessary to correctly execute the task. The replaced-objects environments could only be addressed consistently when the task was represented at the second or third levels of abstraction, and the replaced-scoop environments could only be addressed using the third, most-abstracted representation. These results suggest that as the source and target environment become more dissimilar, abstraction is necessary in order to correctly address the target task.

	Abstraction 1	Abstraction 2	Abstraction 3
Displaced-Object Environments	10/10	10/10	10/10
Replaced-Object Environments	0/10	10/10	9/10
Replaced-Scoop Environments	0/10	0/10	8/10

Table 2: Success Rates for Each Abstraction Applied to the Scooping Task

5 Conclusion

Task transfer becomes a more nuanced problem when the agent's input and output occurs in the domain of low-level perception and action, because the task must first be abstracted using a higher-level representation. The described experiment and preliminary results support the claim that there is a correlation between (i) the level of similarity between the source and target environments and (ii) the level of abstraction that should be used to address a transfer problem. This indicates that different transfer problems may require that knowledge is represented at different levels of abstraction.

In future work, we plan to address the problem of detecting the level of similarity between the source and target environments. Once a level of similarity has been determined for a particular transfer problem, the appropriate level of abstraction for transfer can be selected.

Acknowledgments. This material is based upon work supported by the National Science Foundation Graduate Research Fellowship under Grant No. DGE-1148903.

References

1. Fitzgerald, T., Bullard, K., Thomaz, A., Goel, A.: Situated mapping for transfer learning. In: Fourth Annual Conference on Advances in Cognitive Systems (2016)
2. Fitzgerald, T., McGreggor, K., Akgun, B., Thomaz, A., Goel, A.: Visual case retrieval for interpreting skill demonstrations. In: International Conference on Case-Based Reasoning. pp. 119–133. Springer International Publishing (2015)
3. Floyd, M.W., Esfandiari, B.: A case-based reasoning framework for developing agents using learning by observation. In: Tools with Artificial Intelligence (ICTAI), 2011 23rd IEEE International Conference on. pp. 531–538. IEEE (2011)
4. Floyd, M.W., Esfandiari, B., Lam, K.: A case-based reasoning approach to imitating robocup players. In: FLAIRS Conference. pp. 251–256 (2008)
5. Ontañón, S., Mishra, K., Sugandh, N., Ram, A.: Case-based planning and execution for real-time strategy games. In: Case-Based Reasoning Research and Development, pp. 164–178. Springer (2007)
6. Pastor, P., Hoffmann, H., Asfour, T., Schaal, S.: Learning and generalization of motor skills by learning from demonstration. In: Robotics and Automation, 2009. ICRA'09. IEEE International Conference on. pp. 763–768. IEEE (2009)
7. Ros, R., Arcos, J.L., Lopez de Mantaras, R., Veloso, M.: A case-based approach for coordinated action selection in robot soccer. Artificial Intelligence 173(9), 1014–1039 (2009)

8. Schaal, S.: Dynamic movement primitives-a framework for motor control in humans and humanoid robotics. In: Adaptive Motion of Animals and Machines, pp. 261–280. Springer (2006)
9. Smyth, B., Keane, M.T.: Adaptation-guided retrieval: questioning the similarity assumption in reasoning. Artificial Intelligence 102(2), 249–293 (1998)
10. Taylor, M.E., Stone, P.: Transfer learning for reinforcement learning domains: A survey. The Journal of Machine Learning Research 10, 1633–1685 (2009)
11. Taylor, M.E., Stone, P., Liu, Y.: Transfer learning via inter-task mappings for temporal difference learning. Journal of Machine Learning Research 8(1), 2125–2167 (2007)
12. Taylor, M.E., Whiteson, S., Stone, P.: Transfer via inter-task mappings in policy search reinforcement learning. In: Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems. p. 37. ACM (2007)
13. Trevor, A.J., Gedikli, S., Rusu, R.B., Christensen, H.I.: Efficient organized point cloud segmentation with connected components. Semantic Perception Mapping and Exploration (2013)