Imitation Learning as Cause-Effect Reasoning

Garrett Katz[1](✉), Di-Wei Huang[1],
Rodolphe Gentili[2,3,4], and James Reggia[1,3,4,5]

[1] Department of Computer Science, University of Maryland, College Park, USA
{gkatz,dwh,reggia}@cs.umd.edu
[2] Department of Kinesiology, University of Maryland, College Park, USA
rodolphe@umd.edu
[3] Neuroscience and Cognitive Science Program,
University of Maryland, College Park, USA
[4] Maryland Robotics Center, University of Maryland, College Park, USA
[5] Institute for Advanced Computer Studies,
University of Maryland, College Park, USA

Abstract. We propose a framework for general-purpose imitation learning centered on *cause-effect reasoning*. Our approach infers a hierarchical representation of a demonstrator's intentions, which can explain *why* they acted as they did. This enables rapid generalization of the observed actions to new situations. We employ a novel causal inference algorithm with formal guarantees and connections to automated planning. Our approach is implemented and validated empirically using a physical robot, which successfully generalizes skills involving bimanual manipulation of composite objects in 3D. These results suggest that cause-effect reasoning is an effective unifying principle for cognitive-level imitation learning.

Keywords: Artificial general intelligence · Imitation learning · Cause-effect reasoning · Parsimonious covering theory · Cognitive robotics

1 Introduction

During early childhood development, humans and other primates gain procedural knowledge in large part through *imitation learning* (IL) [15]. Implementing this general-purpose ability in robots will facilitate their wide-spread use. It will also mitigate the risks associated with artificial general intelligence, since a human is kept in the loop when shaping a robot's behavior.

Cognitive science tells us that IL should and does involve understanding the *intentions* of a teacher (or "demonstrator"), in addition to their actions [2,11]. Inferring a teacher's intent can be viewed as a form of cause-effect reasoning: How do hidden intentions *cause* the observed actions? Artificial intelligence (AI) researchers have studied cause-effect reasoning, also known as abductive inference, and its utility for inferring an agent's intentions [4]. However, the connection to robotic IL is largely unexplored. Most IL research has focused on sensorimotor control, with minimal cognitive processing (e.g., [1,3,20]). While certain

B. Steunebrink et al. (Eds.): AGI 2016, LNAI 9782, pp. 64–73, 2016.
DOI: 10.1007/978-3-319-41649-6_7

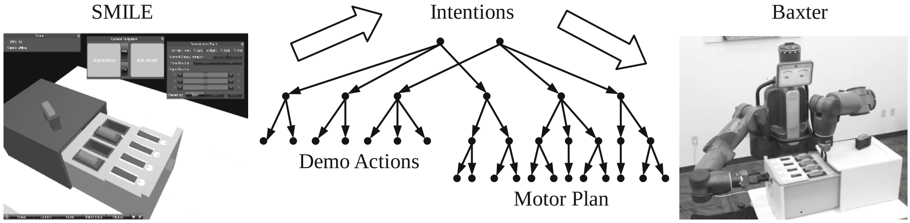

Fig. 1. Our IL framework. Demonstrations are recorded in SMILE (left) and imitated by a robot (right). Hierarchical intentions (center, explained below) are inferred bottom-up using causal reasoning (left block arrow). During imitation, intentions are decomposed top-down into new actions and ultimately motor plans (right block arrow).

cognitive abilities have been modeled for IL (e.g. [6,8,9,14]), to our knowledge, the utility of cause-effect reasoning in particular has not been studied in depth.

Here we hypothesize that cause-effect reasoning is central to cognitive-level, general-purpose IL, and propose a causal IL framework to test this hypothesis (Fig. 1). Using a novel abductive inference algorithm with formal guarantees, our approach constructs a parsimonious explanation for an observed demonstration, in which hypothesized intentions explain observed actions through hierarchical causal relationships. The intentions at the top of the hierarchy can then be carried out in new situations that require different low-level actions and motor plans. In other words, the system generalizes from a single demonstration. Our framework is validated empirically in a real-world application scenario, where a physical robot (Baxter, Rethink Robotics) learns maintenance skills on a hard-drive docking station. Demonstrations are recorded in a virtual environment called SMILE, developed previously by our research group [13].

2 Demonstrating Hard-Drive Maintenance

Our current work has focused on a learning scenario we call the "hard-drive docking station." A robot must learn to maintain a docking station for several hard-drives subject to hardware faults (Fig. 1, right). Each drive slot has an LED fault indicator and a switch that must be toggled when changing drives. The goal is to replicate a teacher's *intentions*, on the basis of just one demonstration, in new situations that require different motor plans. For example, if the teacher discards a faulty drive and replaces it with a spare, so must the robot, even when a different slot is faulty and the spare is somewhere else. Due to the robot's physical constraints, it may need to use entirely different motor actions than the teacher, such as using different arms, or handing off objects between grippers. For experimental purposes, we used faux 3D-printed "hard-drives," and an Arduino controller for the LEDs and switches. For testing and development, we have also used a toy block scenario: The teacher stacks blocks in various patterns such as letters, and the robot must replicate those patterns even when extraneous blocks are present and the important blocks are in completely different initial positions.

To capture human demonstrations, we use SMILE, the virtual environment shown in Fig. 1 (left). In SMILE a user can manipulate objects with intuitive GUI controls and record their actions [13]. The recording is output in both video format and a machine-readable event transcript, describing which objects were grasped, with which hands, and real-time changes in object positions and orientations. It contains no indication of the user's intentions. SMILE bypasses the challenge of human motion capture, and is appropriate when *how the human changes objects* is less important than *how the objects change*.

3 Imitation Learning with Causal Inference

3.1 Learning Skills by Explaining Demonstrations

Given a demonstration transcript from SMILE, our system instantiates a causal hierarchy of intentions in a bottom-up fashion to explain what was observed. For compact representation, all intentions are *parameterized*: An intention signature such as "grasp⟨*object*, *gripper*⟩" can be grounded by, for example, binding *object* to the value "drive 1" and *gripper* to the value "left," which signifies that drive 1 is grasped with the left gripper. Low-level intention sequences such as

⟨grasp drive 1 with left, move left above slot 3, lower left, open left⟩

can be caused by higher-level intentions such as "insert drive 1 in slot 3 with left gripper," which in turn may be caused by intentions such as "get drive 1 to slot 3" (with any gripper), and so on. In our real-world robotics domain, real-valued parameters (omitted from the text) are also needed to represent things like the precise orientation and position of left above slot 3. Each individual intention, and the sub-intentions it can cause directly, must be pre-defined by a human referred to as the *domain author*. The highest-level intentions pre-defined in our knowledge base include dock manipulations such as "open dock" and "toggle switch," as well as a generic "get object 1 to object 2" intention, which may cause various sub-intentions such as temporarily emptying grippers, clearing obstructions from object 1 or object 2, and handing off objects between grippers. Note that these high-level intentions are rather general, but not general enough that a single root intention can explain an entire demonstration. The typical demonstration can only be explained by a novel *sequence* of high-level intentions, that was not pre-defined by the domain author.

The goal during learning is to infer such a sequence, correctly ordered and parameterized, given the SMILE demonstration. Our inference mechanism is a novel extension of *Parsimonious Covering Theory* (PCT), a formal computational model of causal reasoning that has been applied in diverse fields such as medical diagnosis, circuit fault localization, natural language processing, and semantic web technology [7,12,17,19], but not to intention inference. In our context, a *cover* is an intention sequence that explains the observed actions. The shortest, top-level covers are considered most *parsimonious*, and used to represent learned skills. The idea that parsimony is a unifying factor in explanation and inference is widely supported in philosophy and cognitive science [5]. Parameters and hierachical structure are our new extensions to PCT.

We formalize the intention inference problem as follows. A *causal intention hierarchy* is a tuple $\mathcal{I} = (S, T, X, V, C)$, where S is the set of possible states of the robot's environment, T is a set of intention signatures (e.g., "grasp," without parameters bound to specific values), and X is a set of possible parameter values (e.g., "drive 1"). The set of all *vertices* is $V \subset S \times T \times X^*$, where * denotes the Kleene closure (i.e., X^* is the set of all finite parameter lists).[1] Each vertex $v \in V$ is a tuple $(s, t, \langle x \rangle)$, representing some intention t with some parameter list $\langle x \rangle$ in some state s. Lastly, $C \subset V \times V^*$ is the *causal relation*. Each element $(u, \langle v \rangle) \in C$ signifies that a parent vertex u might cause the sequence of child vertices $\langle v \rangle$. Note that C may be many-to-many: the same u might cause any of several different $\langle v \rangle$'s and vice-versa. C is depicted by drawing downwards arrows representing causal relations from parents to their children, and horizontal arrows across edges to signify ordering constraints, as illustrated in Fig. 2.

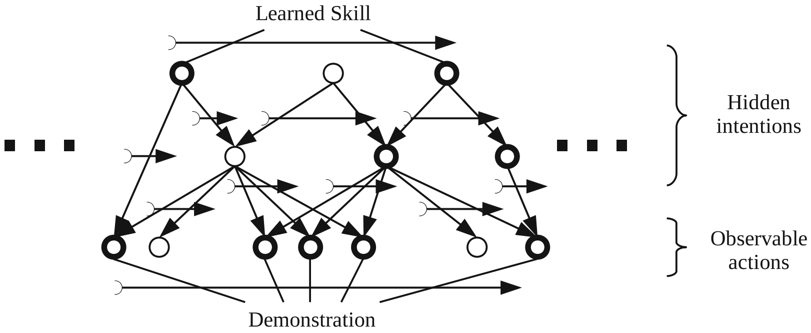

Fig. 2. An example of a causal hierarchy. The schematics are explained in the text.

If X includes floating-point values, it may be infeasible to store C in its entirety. However, only a small subset of parameter values from X will appear in any real-world demonstration, so the relevant portion of C can be constructed and stored online as needed. To this end, we introduce the function $\textsc{causes}(\langle v \rangle) = \{u \mid (u, \langle v \rangle) \in C\}$, which returns only those $u \in V$ that could cause a given sequence $\langle v \rangle$. CAUSES is what the domain author must define.

Another issue is that causal chains from intentions to actions may have different path lengths. For example, in Fig. 2, the leftmost path from the top layer to the bottom layer has length 1, whereas the others have length 2. To identify explanations like this in a bottom-up fashion, we define a more

[1] We denote a finite list of length N by $\langle x_i \rangle_{i=1}^{N} = \langle x_1, x_2, ..., x_N \rangle$. For brevity, a finite list of arbitrary length is denoted simply $\langle x \rangle$, whereas a list of length 1 is always denoted with a single subscript: $\langle x_1 \rangle$.

reflexive, extended CAUSES function as follows: For any singleton sequence $\langle v_1 \rangle$, EXTCAUSES($\langle v_1 \rangle$) = CAUSES($\langle v_1 \rangle$) $\cup \{v_1\}$. For non-singleton sequences $\langle v \rangle$ with length greater than 1, EXTCAUSES($\langle v \rangle$) = CAUSES($\langle v \rangle$).

Finally, we define covers. Informally, a cover is a sequence of intentions, where every observed action is path-connected to some intention in the sequence. An ℓ-cover is a cover where each path has length at most ℓ. Formally, we inductively define covers as follows. A sequence $\langle u_k \rangle_{k=1}^{K} \in V^*$ is a *1-cover* for another sequence $\langle v \rangle \in V^*$, if there is a partition of $\langle v \rangle$ into K consecutive, contiguous subsequences $\langle v \rangle^{(1)}, \langle v \rangle^{(2)}, ..., \langle v \rangle^{(K)}$, such that $u_k \in$ EXTCAUSES($\langle v \rangle^{(k)}$) for every k. A sequence $\langle u \rangle \in V^*$ is an *ℓ-cover* (or simply *cover*) of $\langle w \rangle \in V^*$ if there is some $\langle v \rangle \in V^*$ such that $\langle u \rangle$ is a 1-cover of $\langle v \rangle$ and $\langle v \rangle$ is an $(\ell - 1)$-cover of $\langle w \rangle$.

We may now formally state the intention inference problem as we conceive it. Let $A \subset V$ be a distinguished subset of vertices called *observable actions* (Fig. 2, bottom layer). A *demonstration* is some $\langle a \rangle \in A^*$ (Fig. 2, bottom layer, bold-faced vertices). The *intention inference problem* is to compute the most parsimonious covers of $\langle a \rangle$. Readers may note that a useful parsimony criteria employed by PCT, *irredundancy* [17], is necessarily satisfied by all covers as defined here: no proper subset of a cover is a cover itself. Therefore, beyond irredundancy, we define the *parsimonious* covers, or *explanations*, to be the minimum cardinality top-level covers (Fig. 2, top layer, bold-faced vertices).

We have derived a provably correct procedure for the intention inference problem during learning, shown in Algorithm 1. The inputs are a causal intention hierarchy $\mathcal{I}$ as defined above (encoded as a CAUSES function) and a demonstration $\langle a \rangle$. The algorithm incrementally constructs all covers in a bottom-up fashion, accumulating all ℓ-covers in a set $H^{(\ell)}$ during the ℓ^{th} iteration (lines 4–17). Each layer $H^{(\ell)}$ is populated by finding all 1-covers of all child sequences from the previous layer $H^{(\ell-1)}$ (lines 5–14). The 1-covers for every such child sequence are also constructed incrementally: 1-covers for the leading sequences up to index $k - 1$ are used to construct the 1-covers up to index k, as k ranges over the full sequence (lines 6–12). This step checks every partition point $j \leq k$, and concatenates every 1-cover of the leading subsequence up to $j - 1$ with every cause of the trailing subsequence from j to k (lines 8–11). This pairwise concatenation operation (line 10), denoted by $\oplus$, is defined as follows:

$$Y \oplus Z = \{\langle y_1, ..., y_M, z \rangle \mid \langle y_m \rangle_{m=1}^{M} \in Y \text{ and } z \in Z\} \ .$$

The leading covers up to k are accumulated in sets $G^{(1)}, ..., G^{(j)}, ...G^{(k)}$, which are used to populate $G^{(k+1)}$.

It can be shown (Sect. 4) that once k reaches N, $G^{(N)}$ contains all 1-covers of the full child sequence. These get added to the current $H^{(\ell)}$ (line 13). The top-most $H^{(\ell)}$ is returned when no new covers are found (line 16). The most parsimonious covers can then be extracted from $H^{(\ell)}$ in a post-processing step. This design choice allows researchers to compare alternative parsimony criteria (beyond irredundancy) without modifying the core algorithm.

Algorithm 1. The Intention Inference Algorithm.

```
procedure COVER($\mathcal{I}$, $\langle a \rangle$)        ▷ $\mathcal{I}$ is supplied implicitly through CAUSES
    $H^{(0)} \leftarrow \{\langle a \rangle\}$        ▷ Start with the demo $\langle a \rangle$
    for $\ell \leftarrow 1, 2, ...$ do        ▷ Bottom-up cover construction
        $H^{(\ell)} \leftarrow \emptyset$        ▷ Begin finding $\ell$-covers
        for $\langle v_i \rangle_{i=1}^{N} \in H^{(\ell-1)}$ do        ▷ Process each child sequence
            for $k \leftarrow 0, 1, ..., N$ do        ▷ Process child sequence incrementally
                $G^{(k)} \leftarrow \emptyset$        ▷ Begin covering up to $k$
                for $j \in \{1, ..., k\}$ do        ▷ Check all leading-trailing splits
                    $U \leftarrow$ EXTCAUSES($\langle v_i \rangle_{i=j}^{k}$)        ▷ Get trailing causes
                    $G^{(k)} \leftarrow G^{(k)} \cup (G^{(j-1)} \oplus U)$        ▷ Append to leading covers
                end for
            end for
            $H^{(\ell)} \leftarrow H^{(\ell)} \cup G^{(N)}$        ▷ Add full covers to next layer
        end for
        if $H^{(\ell)} = H^{(\ell-1)}$ then        ▷ Check for any new covers
            return $H^{(\ell)}$        ▷ No new covers, terminate
        end if
    end for
end procedure
```

3.2 Imitation and Generalization

Once a parsimonious cover for the observed demonstration has been found and saved, the robot is ready to generalize the learned skill to new situations. When asked to imitate, the robot begins with visual processing to identify object properties and relationships in the new scene. Our current implementation uses simple computer vision techniques as a baseline. Next, the objects found by visual processing must be matched with the corresponding objects in the original demonstration. For example, consider the toy block IL scenario with three blocks, two of which get manipulated by the demonstrator. When the robot sees three blocks in a new situation, it does not know a priori which one should be treated as "block 1" from the demonstration, which should be treated as "block 2", and which is extraneous. A simple algorithm computes the one-to-one object matching that best preserves salient properties (shape, color) and relationships (part-whole, atop-below). For example, now consider the drive maintenance IL scenario. Suppose that in the demonstration, slot $1^{(demo)}$ is occupied and LED $1^{(demo)}$ is red, and in the new scene, slots $2^{(new)}$ & $3^{(new)}$ are occupied but only LED $3^{(new)}$ is red. Slots and LEDs $1^{(demo)}$ & $3^{(new)}$ will be matched, rather than $1^{(demo)}$ & $2^{(new)}$, since the configuration of colors, part-whole relationships, and atop-below relationships is better preserved. The matching algorithm is based on greedy weighted bipartite matching. Note that the matching only compares the initial state in the demo with the initial state during imitation. Incorporating the inferred intentions is a potential research direction, although similar problems involving plan reuse are known to be hard [16].

Once matching is complete, the parameter bindings in the top-level intentions are updated to point to the corresponding objects in the new scene. AI planning techniques can then be used in a top-down manner to plan a sequence of low-level motor commands that carry out these intentions. In particular, we employ *Hierarchical Task Network* (HTN) planning, in which high-level tasks are decomposed into lower-level sub-tasks and ultimately executable actions [10]. It turns out that intentions in our causal hierarchy can be mapped directly onto the formal HTN notion of tasks. A corollary is that, if CAUSES formally inverts the HTN planning operators, then Algorithm 1 formally inverts the HTN planning algorithm. To our knowledge, this is the first provably correct inversion.

Like the causal relation, HTN operators can map the same parent intention onto several alternative child sequences. These represent alternate strategies for carrying out the parent intention, some of which may be more or less appropriate depending on the current state of the environment. The HTN planner can search each branch, simulating its effects on the environment, and avoid branches that fail. Consequently, the resulting actions planned for the new situation may differ significantly from the observed actions in the demonstration. For example, suppose "block 1" was grasped and released by the teacher's left hand during demonstration, but its matching block is only reachable by the robot's right gripper in the new situation. When the HTN planner decomposes the high-level "get block 1 to ..." task, it will find the branch most suitable to the new situation, namely picking up with the right and handing off to the left. This is an example of the bimanual coordination supported in our implementation.

The robot's capacity for generalization boils down to this fact that the same parent intention can cause any of several alternative sub-intention branches, in a way sensitive to the current state of the environment. The results of just one branch are observed in the demonstration, but many other branches exist that are more appropriate for other situations. Inferring the teacher's intentions exposes these other branches, and the higher up the hierarchy, the more branches get exposed. So cause-effect inference of parsimonious covers is central to one-shot generalization. Moreover, the lowest-level HTN operators can invoke motion planning routines, which convert target gripper positions into joint angles that respect the physical constraints of the robot. As a result, the causal hierarchy can be extended to a level deeper than the object-centric actions recorded in SMILE, producing concrete motor plans suitable for physical robot execution.

4 Theoretical and Empirical Results

Algorithm 1 is sound and complete, although a naive set implementation for $H^{(\ell)}$ and $G^{(k)}$ may lead to intractable storage requirements. PCT mitigates this sort of problem using storage-efficient data structures called *generators*. Given space limitations, we cannot describe our generator-based implementations here. To convey the key ideas of the correctness proof, we provide a sketch for Algorithm 1 as is, assuming correct generator-based implementations of the relevant set operations. The complete proof will be included in a forthcoming publication.

Theorem 1. *Let H be the return value of* $\textsc{cover}(\mathcal{I}, \langle a \rangle)$. *Every element of H is a cover of $\langle a \rangle$; every cover of $\langle a \rangle$ is an element of H.*

Proof (sketch). Suppose on lines 7–11 that for $j \leq k$, each $G^{(j-1)}$ contains precisely the 1-covers of $\langle v_i \rangle_{i=1}^{j-1}$. Then any element of $G^{(j-1)} \oplus U$ added to $G^{(k)}$ on line 10 must be a 1-cover of $\langle v_i \rangle_{i=1}^{k}$ by definition. Conversely, any 1-cover of $\langle v_i \rangle_{i=1}^{k}$ has its last contiguous child sub-sequence start at some index $\tilde{\jmath} \leq k$. This cover consists of leading causes in $G^{(\tilde{\jmath}-1)}$, and a trailing cause in $\textsc{extcauses}(\langle v_i \rangle_{i=\tilde{\jmath}}^{k})$. Therefore it will be added to $G^{(k)}$ on line 10 when $j = \tilde{\jmath}$. It follows that $G^{(k)}$ will contain all and only the 1-covers of $\langle v_i \rangle_{i=1}^{k}$ by line 11, and by induction on k, $G^{(N)}$ will contain precisely the 1-covers of the full $\langle v_i \rangle_{i=1}^{N}$ by line 12. Consequently, $H^{(\ell)}$ receives every 1-cover of $\langle v_i \rangle_{i=1}^{N}$ on line 13. Assuming $H^{(\ell-1)}$ contains every $(\ell-1)$-cover of $\langle a \rangle$, it follows that $H^{(\ell)}$ has accumulated every ℓ-cover of $\langle a \rangle$ by line 14. Now by induction on ℓ, every cover of $\langle a \rangle$ has been found when the algorithm returns on line 16. Termination in finite time can be guaranteed under reasonable conditions on $\mathcal{I}$. □

To ascertain the practical utility of these theoretical results, we performed an initial assessment of our framework using the dock scenario. Four different skills were taught to the robot. Each skill was demonstrated twice in SMILE, using different initial states for the maintenance dock each time. Algorithm 1 was used to infer intentions in each demonstration. Finally, the robot was asked to imitate each demonstration four times, again using different initial dock states each time. The result is 8 demonstrations total and 32 imitation trials total. In every demonstration and trial, the initial dock states were automatically and randomly generated, varying the number and position of spare drives, which slots were occupied, and which LEDs were red. The robot was taught four skills: (1) discarding a red LED drive, (2) replacing (and discarding) a red LED drive with a spare on top of the dock, (3) replacing (and discarding) a red LED drive with a green one, and (4) swapping a red LED drive with a green one.

On every demonstration, Algorithm 1 terminated in a matter of minutes (see Table 1), so while time complexity is a theoretical concern it was not prohibitive in practice. Nevertheless, developing a more efficient algorithm that scales to more complex examples is the subject of future work. Additionally, inspection showed that in all new situations, the robot was generating a suitable, correct plan of low-level actions to execute. Unfortunately, our physical robot failed midway through execution in 31.25 % of the trials due to sensorimotor errors (see Table 2). For example, spare drive locations as determined by visual processing would be too inaccurate for a successful grasp (visual failures), or a drive would be misaligned with a slot and not inserted properly (motor failures). These issues are due to both our simplistic sensorimotor processing and limited accuracy in Baxter's hardware as compared to more expensive robots. Nevertheless, the key result is that the cognitive learning process produced correct plans in 100 % of the trials. Sensorimotor processing is not our primary focus here so we do not consider the execution fail rate to be a significant objection to this work (although we are currently working to improve the sensorimotor processing).

Table 1. Run times in minutes of Algorithm 1. d is the length of the input $\langle a \rangle$, i.e. the number of steps recorded in the SMILE event transcript.

Skill	Demo 1	Demo 2
Remove red drive	0.03 ($d = 7$)	0.10 ($d = 10$)
Replace red with spare	2.31 ($d = 14$)	2.52 ($d = 14$)
Replace red with green	2.52 ($d = 15$)	2.47 ($d = 15$)
Swap red with green	0.72 ($d = 16$)	0.73 ($d = 16$)

Table 2. Frequencies of success and failure during imitation trials.

Class	Frequency
Planning failures	0
Vision failures	3
Motor failures	7
Successful trials	22

5 Conclusion

We have introduced a general-purpose, cognitive-level IL framework, based on hierarchical cause-effect reasoning. We validated our framework on a modest set of IL tasks, suggesting that using causal knowledge to infer a teacher's intentions, rather than copying their actions, is a promising approach to one-shot IL. Future work should evaluate our approach on more complex and varied tasks, with controlled end-user studies. The computational complexity of our algorithms should be reduced, and more reasoning should be shifted from the domain author into the algorithms themselves. Formal links with more modern hierarchical planners, such as Hierarchical Goal Networks [18], will be sought. Lastly, although our system accumulates a database of inferred top-level intention sequences, these sequences are not fed back into the hierarchy so that they can become *sub*-intentions of even higher-level parents. We hope to extend our framework in this direction so that a teacher can enrich the robot's knowledge base over time.

Acknowledgements. This work was supported by ONR award N000141310597. Thanks to Ethan Reggia for building the hard-drive docking station.

References

1. Akgun, B., Cakmak, M., Yoo, J.W., Thomaz, A.L.: Trajectories and keyframes for kinesthetic teaching. In: Proceedings of the 7th Annual ACM/IEEE International Conference on Human-Robot Interaction, pp. 391–398. ACM (2012)
2. Baldwin, D., Baird, J.: Discerning intentions in dynamic human action. Trends Cogn. Sci. **5**(4), 171–178 (2001)
3. Barros, J., Serra, F., Santos, V., Silva, F.: Tele-kinesthetic teaching of motion skills to humanoid robots through haptic feedback. In: IEEERAS Humanoids' 2014 Workshop on Policy Representations for Humanoid Robots (2014)
4. Carberry, S.: Techniques for plan recognition. User Model. User-Adap. Inter. **11**(1–2), 31–48 (2001)
5. Chater, N., Vitányi, P.: Simplicity: a unifying principle in cognitive science? Trends Cogn. Sci. **7**(1), 19–22 (2003)
6. Chella, A., Dindo, H., Infantino, I.: A cognitive framework for imitation learning. Robot. Auton. Syst. **54**(5), 403–408 (2006)

7. Dasigi, V., Reggia, J.: Parsimonious covering as a method for natural language interfaces to expert systems. AI Med. **1**(1), 49–60 (1989)
8. Dindo, H., Chella, A., La Tona, G., Vitali, M., Nivel, E., Thórisson, K.R.: Learning problem solving skills from demonstration: an architectural approach. In: Schmidhuber, J., Thórisson, K.R., Looks, M. (eds.) AGI 2011. LNCS, vol. 6830, pp. 194–203. Springer, Heidelberg (2011)
9. Friesen, A.L., Rao, R.P.: Imitation learning with hierarchical actions. In: 9th International Conference on Development and Learning, pp. 263–268. IEEE (2010)
10. Ghallab, M., Nau, D., Traverso, P.: Automated Planning. Elsevier, Amsterdam (2004)
11. Haikonen, P.: The Cognitive Approach to Conscious Machines. Imprint Academic (2003)
12. Henson, C., Sheth, A., Thirunarayan, K.: Semantic perception: converting sensory observations to abstractions. Internet Comput. **16**(2), 26–34 (2012). IEEE
13. Huang, D.-W., Katz, G.E., Langsfeld, J.D., Oh, H., Gentili, R.J., Reggia, J.A.: An object-centric paradigm for robot programming by demonstration. In: Schmorrow, D.D., Fidopiastis, C.M. (eds.) AC 2015. LNCS, vol. 9183, pp. 745–756. Springer, Heidelberg (2015)
14. Jansen, B., Belpaeme, T.: A computational model of intention reading in imitation. Robot. Auton. Syst. **54**(5), 394–402 (2006)
15. Meltzoff, A., Moore, A.: Imitation of facial and manual gestures by human neonates. Science **198**(4312), 75–78 (1977)
16. Nebel, B., Koehler, J.: Plan reuse versus plan generation. Artif. Intell. **76**(1), 427–454 (1995)
17. Peng, Y., Reggia, J.: Abductive Inference Models for Diagnostic Problem-Solving. Springer, New York (1990)
18. Shivashankar, V., Alford, R., Kuter, U., Nau, D.: The GoDeL planning system: a more perfect union of domain-independent and hierarchical planning. In: Proceedings of the 23rd International Joint Conference on AI. AAAI (2013)
19. Wen, F., Chang, C.: A new approach to fault diagnosis in electrical distribution networks using a genetic algorithm. AI Eng. **12**(1), 69–80 (1998)
20. Wu, Y., Su, Y., Demiris, Y.: A morphable template framework for robot learning by demonstration. Robot. Auton. Syst. **62**(10), 1517–1530 (2014)