Statistical Relational Learning of Object Affordances for Robotic Manipulation

Bogdan Moldovan[1], Martijn van Otterlo[1], Plinio Moreno[2], José Santos-Victor[2], and Luc De Raedt[1]

[1] Department of Computer Science, Katholieke Universiteit Leuven, Belgium
[2] Electrical & Computer Engineering, Instituto Superior Técnico, Portugal

Abstract. We present initial results of an application of statistical relational learning using ProbLog to a robotic manipulation task modeled using affordances. Affordances encompass the action possibilities on an object, so previous works have presented models for just one object. However, in scenarios where there are multiple objects that interact between each other, it is very useful to consider the advantages of the statistical relational learning.

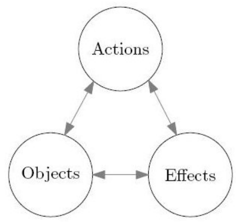
1 Introduction

Robotics is a vast and active area seeking to develop mobile, *physical* agents capable of reasoning, learning and manipulating their environment. Early approaches such as in Shakey used logical representations such as STRIPS, and many more approaches use various other kinds of symbolic knowledge representation [2, 9]. In addition to symbolic (or, semantic) methodologies, the physical aspect of robots requires dealing with various kinds of *uncertainty* typically not handled by symbolic formalisms. These aspects include interpreting noisy sensors, processing image streams from cameras, controlling noisy physical actuators for manipulation, and in general, solving many *grounding* and *anchoring* problems. Therefore, much of current robotics research is concerned with *probabilistic reasoning and learning* techniques [10] instead of symbolic representations. *Statistical relational learning* (SRL) [7, 1] combines logical representations, probabilistic reasoning and machine learning. Several recent works have explored the use of SRL and have shown how to effectively combine probabilistic and logical methods in robotics domains, e.g. see the kitchen scenario in [3].

We outline initial results of using SRL, in particular ProbLog [8], in a manipulation scenario with an iCub robot. We extend recent results in *imitation learning* in which (video) demonstrations of object manipulation (e.g. by a human) are used to learn *affordances* [4–6]. Affordances are a way to structure the robot's environment in terms of *what it can do with specific objects*. We extend this model by proposing an initial approach towards generalization, using probabilistic logical affordance models. This can be done by the use of SRL in a multiple object scenario, which allows for the generalization over objects and actions by allowing the use of already learned one and two object models together with known logical rules to be used for inference. This allows for a greater flexibility in modeling complex multi-object environments for robot manipulation than the previous approaches [4–6] of modeling the scene with a specific Bayesian Network (BN). Next, we explain the one-object case and in Section 3 we discuss the learning of manipulation skills. Section 4 describes the relational extension and initial results after which we conclude with planned extensions, mostly carried out in the context of the EU-project on *Flexible Skill Acquisition and Intuitive Robot Tasking for Mobile Manipulation in the Real World* (First-MM).

2 Affordance-based Models

We first discuss the affordance learning setting of [4–6]. Affordances capture *action opportunities* (e.g. *what can one do with an object?*) to structure the environment. The typical setup involves a robot (with an arm) and a table with physical objects (cubes, balls, etc.). Three main aspects of the approach are *actions* (A), *object properties* (O) and *effects* (E). These, and their relationships, are presented in Figure 1. Actions are physical manipulation skills that can be applied to objects and include *grabbing* (and releasing), *pushing* (in a plane away from the robot), and *tapping* (sideways). Object properties are aspects that can be measured from perceptual devices (such as vision) and involve color, shape, and size. Effects are measurable features that change once an action is performed, e.g. the velocity of the hand, the velocity (or distance between) between the hand and the object, etc. Depending on which data is available to the robot and which aspects need to be predicted, the model can also be used for planning and other tasks, see the table in Figure 1.



Inputs	Outputs	Function
(O, A)	E	Effect prediction
(O, E)	A	Action recognition/planning
(A, E)	O	Object recognition/selection

Fig. 1. Affordances model: relations between objects, actions, effects [5, 6].

Learning to imitate manipulation actions is accomplished through several steps. First, data is collected from the robot performing several actions on different objects. All features for A, O and E are measured for each demonstration, and then processed, discretized and aggregated to acquire a dataset where all features have a relatively small (ordinal) set of values. A probabilistic model can then be learned that captures the dependencies between the three types of features. This can be used for various purposes; e.g. for imitation learning; the robot observes a human manipulating an object (with properties O), observes the effects of the demonstrated action (E) and computes the *most likely* action causing E and O, i.e. $\arg\max_A P(A|E, O)$. Note that the robot has now computed how to imitate a certain effect on an object in terms *of his own action repertoire* which is obviously different from that of the human.

3 Learning Relational Skills and Experiments

The setting we have just described focuses on learning to manipulate a single object. The data (E, A and O) is used to learn a (propositional) BN and single actions for each object (e.g. *large cubes* should always be tapped to the left). We envision a more general setting in which manipulation skills involve i) multiple objects, ii) object interactions while manipulating, iii) behaviors depending on spatial configurations of objects, and iv) sequential actions.

A typical example in the First-MM project is the *shopping scenario*. Part of it is depicted in Figure 2. The robot is given a shelf, where several objects are already present and their object properties (e.g. shape, location, orientation, etc.) are known. An additional object is present in front of the shelf, and the task of the robot is to place this object onto it, in a context of multiple other objects. We set up experiments where an iCub robot has to interact with one or two objects on the shelf. The *active* object is the one the robot acts upon, and the *passive* object may interact with the active one through the robot's actions. Once the setting is extended to more than one object, object interactions occur (e.g. pushing an object into another object), and the (spatial) relationships between objects have to be taken into account. We ran 87 experiments, 37 replicating the one object experiment in [4–6] and 50 involving two objects. Figure 3(l) illustrates the two objects setting. On the left is the robot's hand (white) and on the black table there's a yellow rectangular prism (active), and a blue rectangular prism (passive). The experiments were recorded using a top-view camera. The videos were processed in order to extract features (e.g. object shape, color, location, etc.), to compute the feature values for groups of frames, and then to cluster and discretise the values. From this data we first learn a BN using the K2 algorithm (using Matlab), and then similar to [6, 5], we learn the parameters (i.e. the probabilities) of this BN. Figure 3(r) shows a subset of the BN, involving the relations between the action, the magnitude of the displacement of the active object, and the displacement orientation of the two objects.

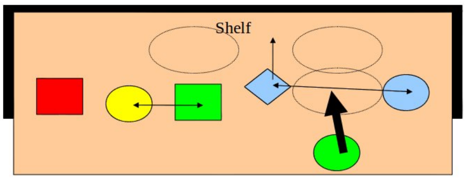


Fig. 2. Relational interactions on a shelf, with colored objects and possible goal locations.

Learning an affordance model for these situations and the corresponding BN, and the use of SRL allows to achieve a generalization over multiple objects in this setting. Using SRL for affordances allows the robot to learn high-level skills, including motion planning, from low-level components such as the actions and their effects on objects with given properties. The ultimate goal of this research is the temporal aspect of the setting, in which a plan consists of a set of actions. To imitate the plan and learn necessary manipulation skills, the robot needs to recognize *individual* actions in the plan. The rest of the paper will focus on recognizing individual actions.

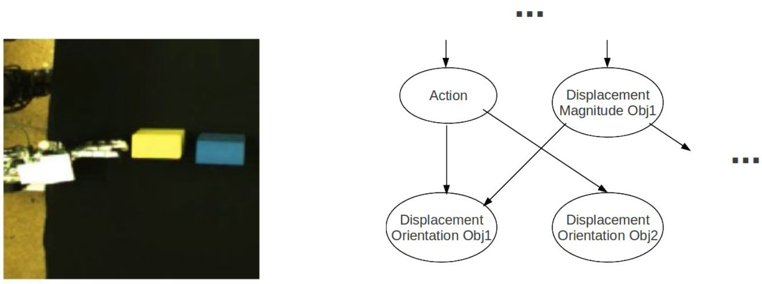


Fig. 3. **(l)** video still of the data showing the robot hand and two objects, **(r)** part of the Bayesian network obtained from structure learning.

4 ProbLog Modeling and Results

SRL [7], a subfield of AI, studies the combination of logical representations, probabilistic reasoning mechanisms and machine learning. *Probabilistic programming languages* (PPL) are programming languages specially designed to describe and infer with probabilistic relational models. The PPL ProbLog is a probabilistic extension of the Prolog logic programming language, where facts are annotated with probabilities and for which several inference methods are available. [8] Additionally, Prolog style logical rules can be used for defining (general) *background knowledge* to answer probabilistic queries.

We continue with the obtained experimental data described in Section 3 before, and add relational properties between objects (e.g. initial relative distance or orientation between two objects) in the two objects scenario as well as relational effects (e.g. final relative distance or orientation between two objects), and model it using ProbLog. The model supports inference for action recognition in this relational extension of [4–6]. Our approach has the advantage that data obtained from the one-object experiments can already be generalized to multiple objects through the use of variables that refrain from referring to specific, hardcoded objects. ProbLog rules generalize over the object displacement magnitude and orientation. Thus later this learning setting can be extended to more than the one and two objects that the experiments investigated, to the full shelf scenario. Knowing that grabbing and moving an object does not involve a second object, the displacement of this main object can be generalized by using the data already obtained from the one object experiment. ProbLog can be used for modeling the relations of the learned Bayesian network and parameters. As an example, in the subnet from Figure 3(r), the following ProbLog statement using annotated disjunctions and the learned parameters models part of the relation between robot action and the magnitude of the displacement of the main object and the displacement orientation of that object:

$$0.8947 :: \texttt{dispOri}(\texttt{ObjMain}, 5); 0.1053 :: \texttt{dispOri}(\texttt{ObjMain}, 7) \\ \quad \leftarrow \texttt{action}(\texttt{ObjMain}, \_, 3), \texttt{dispMag}(\texttt{ObjMain}, 1).$$

This says that if the action type is tap ("3") *and* the displacement magnitude of the main object is small ("1"), then there's a probability of 0.8947 of the displacement orientation of the main object to be in a North (N) direction ("5"), while there's a probability of 0.1053 of it being in an East (E) direction ("7").

The full relation between the robot action and the displacement orientation of the secondary object is modeled as:

$\mathsf{0.0345} :: \mathtt{dispOri}(\mathtt{ObjSec}, 1); \mathsf{0.9655} :: \mathtt{dispOri}(\mathtt{ObjSec}, 7) \leftarrow \mathtt{action}(\_, \mathtt{ObjSec}, 3).$
$\mathsf{0.0476} :: \mathtt{dispOri}(\mathtt{ObjSec}, 1); \mathsf{0.0952} :: \mathtt{dispOri}(\mathtt{ObjSec}, 3);$
$\quad \mathsf{0.6190} :: \mathtt{dispOri}(\mathtt{ObjSec}, 5); \mathsf{0.1429} :: \mathtt{dispOri}(\mathtt{ObjSec}, 6);$
$\quad \mathsf{0.0952} :: \mathtt{dispOri}(\mathtt{ObjSec}, 7) \leftarrow \mathtt{action}(\_, \mathtt{ObjSec}, 4).$

Logical rules are used to specify general behavior. In the example, when an object is grabbed and moved, any other object in the scene remains unchanged (displacement magnitude and orientation are 0), which is modeled as:

$\mathtt{dispMag}(\mathtt{ObjSec}, 0) \leftarrow \mathtt{action}(\mathtt{ObjMain}, \mathtt{ObjSec}, 1).$
$\mathtt{dispOri}(\mathtt{ObjSec}, 0) \leftarrow \mathtt{action}(\mathtt{ObjMain}, \mathtt{ObjSec}, 2).$

General logical rules keep the relations as generic as possible. Similar to the examples presented above, the whole two-object model can be modeled using ProbLog. Because of limited data, not every relation is caught by structure learning. But using ProbLog is effective here too, as additional relations between objects, actions and effects, or constraints in the system can be modeled additionally with the use of logical rules.

After modeling the whole setting in ProbLog, we performed inference in order to do action recognition. It is assumed that the robot has knowledge of the object properties (O), and it can observe the effects (E), and it needs to infer which action (A) was performed. This resumes to querying for the condiditonal probabilities $P(A|O, E)$ in the ProbLog model for each of the 4 possible actions. As an example, an instance of action recognition that we run had:

O: Object1=*Cube* Object2=*Cylinder*
Initial Relative Distance=*Big* Initial Relative Orientation=*NE*
E: Displacement Object1=*Medium* Displacement Object2=*Small*
Displacement Orientation Object1=*N*
Displacement Orientation Object2=*E* Contact Area=*Medium*
Predicted Action: Grasp, Release: 0% **Tap: 86.588%** Push: 13.412%
In this case, the action would be recognized by the robot as being a Tap.

One of the main advantages of using a probabilistic logic language is that it makes learning and inference so flexible by generalizing over the specific objects. Given that most object interactions in this setting involve two objects, the existing two objects model is a good approximation for the general shelf setting. Assuming no multi-way (> 2) interactions at the same time, extending this model to more than two objects on the shelf is easy, being enough to add all the new object property values to the model, and then do inference. Eventually, multi-way interactions can also be learned from experiments, just as the two objects interactions were, and this added to the model for it to become more exact. This can be used to find the best action the robot can do to place an object at the wished location given a configuration of objects around it.

5 Conclusion and Future Work

We described an initial approach towards generalization of robotic affordance model learning in a probabilistic relational setting. Moving to multi-object scenes requires expressive representation schemes to generalize over specific configurations of objects. Future work will involve the learning of full manipulation skills and generalization over more than two objects in a multiple object scenario, and planning given partial knowledge about the environment. One direction in the multiple object setting is that of *activity recognition* and *imitation learning.* Here, the robot detects the object properties and effects and tries to predict which action was performed. This involves recognizing the low-level "atomic actions" involving just one or two objects, by employing the learned models (for either 1 or 2 blocks). The whole demonstrated behavior consists of sequences of such actions, which would allow learning high-level manipulation strategies from demonstration by first distinguishing between their component low-level actions. A second interesting direction is to use the affordance models for *planning* of manipulation strategies. The long term goal is to go towards a full shelf/shopping scenario, in which the robot is instructed where and how to place objects by a human.

Acknowledgements. Bogdan Moldovan is supported by the IWT (agentschap voor Innovatie door Wetenschap en Technologie). This work is supported by the European Community's 7th Framework Programme, grant agreement First-MM-248258.

References

1. L. De Raedt. *Logical and Relational Learning.* Springer, 2008.
2. J. Hertzberg and R. Chatila. AI reasoning methods for robotics. In B. Siciliano and O. Khatib, editors, *Springer Handbook of Robotics.* Springer, 2008.
3. D. Jain, L. Mösenlechner, and M. Beetz. Equipping robot control programs with first-order probabilistic reasoning capabilities. In *ICRA*, 2009.
4. V. Krunic, G. Salvi, A. Bernardino, L. Montesano, and J. Santos-Victor. Affordance based word-to-meaning association. In *ICRA*, 2009.
5. Manuel Lopes, Francisco S. Melo, and Luis Montesano. Affordance-based imitation learning in robots. In *IROS*, 2007.
6. Luis Montesano, Manuel Lopes, Alexandre Bernardino, and Jos Santos-Victor. Learning object affordances: From sensory-motor coordination to imitation. *IEEE Transactions on Robotics*, 24:15–26, 2008.
7. Luc De Raedt and Kristian Kersting. Probabilistic inductive logic programming. In *Probabilistic Inductive Logic Programming*, pages 1–27, 2008.
8. Luc De Raedt, Angelika Kimmig, and Hannu Toivonen. Problog: a probabilistic prolog and its application in link discovery. In *IJCAI.* AAAI Press, 2007.
9. F. Stulp and M. Beetz. Combining declarative, procedural, and predictive knowledge to generate, execute, optimize robot plans. *Robotics and Autonomous Systems*, 56:967–979, 2008.
10. S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents).* MIT Press, 2005.