Opening Doors: An Initial SRL Approach

Bogdan Moldovan[1], Laura Antanas[1], and McElory Hoffmann[1,2]

[1] Department of Computer Science, Katholieke Universiteit Leuven, Belgium
[2] Department of Mathematical Sciences, Stellenbosch University, South Africa

Abstract. Opening doors is an essential task that a robot should perform. In this paper, we propose an approach to predict the action of opening doors, together with the action point where the action should be performed. The input of the classifier is a set of bounding boxes of the door and door handle, together with background knowledge in the form of logical rules. Learning and inference are performed with the probabilistic programming language ProbLog. We evaluate our approach on a dataset of doors and encouraging results are obtained.

1 Introduction

In the context of the EU-project on *Flexible Skill Acquisition and Intuitive Robot Tasking for Mobile Manipulation in the Real World*[3] (First-MM), one of the goals of autonomous robots is to perform mobile manipulation tasks in indoor environments. In this setting, an essential condition is that the robot can operate doors during navigation. A complete solution to this general problem requires a system that can solve several tasks: detecting and localising the door and its handle, recognising the grasping points, finding the right actionable point and action and finally, performing the action on the handle. In this work we focus on two of these tasks, that is detecting the actionable point and the action movement itself. We assume that the door and handle are detectable by the robot. This is an object detection problem that has been previously addressed in the literature using several approaches, using either 2D [1] or 3D information [2, 3, 14].

Detecting the action and action points is a challenging manipulation task. It depends on the sensorimotor control of the robot, the type of handle and the different door properties. To be opened, each door requires different actions depending on the side of the door that the robot is approaching. The action also depends on the type of the handle, its relative position to the door and sometimes even the objects around the door. Usually if hinges are detected on the side of the door and the light switch on the opposite side next to the handle, the door needs to be pulled. Similarly, while the shape of the handle can be quite a good indicator of a suitable action point (i.e., knob), sometimes it cannot be detected reliably, for example when an L-shaped handle is confused with a horizontal one. This may directly influence valid graspable points, given the robot hand type, and limit the actionable handle points. In this case, the relative position of the

[3] More information available at: http://www.first-mm.eu

contact points with the door, the relative positions of the candidate action point to the door sides or other points on the handle may play a key role.

In these cases generalisations over opening directions, point positions and types of handles are needed. These are tasks that would benefit from the use of *statistical relational learning* (SRL) [4, 11]. SRL combines logical representations, probabilistic reasoning mechanisms and machine learning. Several existing SRL approaches can be used to solve our problem [8, 12]. However, in this work we consider *probabilistic programming languages* (PPLs), specially designed to describe and infer with probabilistic relational models. This paper proposes the use of ProbLog (a PPL) models in order to predict the action and action point for opening a door, using solely extracted properties of door images. In our logical representation of the domain, every visual scene is mapped to a logical interpretation. ProbLog also models the noisy nature of the detection aspects and the uncertainty of the environment where the robot operates. Finally, we use a learning from interpretations setting [6] to learn and predict the action for opening a door, as well as the action point. The approach is general enough to be able to deal with point clouds as well as 2D visual images. We evaluated our approach on a dataset containing 60 images of doors. The results are promising and motivate us to continue this work with a real robot scenario. Some work on predicting how to best open a door by a robot setting exists [10]. However, it does not make use of relational representations as we do.

The outline of this paper is as follows: in Section 2 we introduce the problem and the approach used to solve it, and in Section 3 we present our learning and inference setting. We show experimental results in Section 4, after which we conclude and mention future directions in Section 5.

2 Problem Modeling

2.1 Problem Description

We first introduce an initial setting for a high-level relational reasoning system that can be used by a robot for opening doors. We assume the robot is able to detect doors and door handles so we assume to have access to boundary boxes in the image for both the door frame and the door handle. Figure 1 presents two such examples of detected door frame (in red) and handle (in blue) with their bounding boxes. Later, we can add prior probabilities on the positions of the frame and handle for a more realistic scenario which involves object detection uncertainty. The setting can be expanded to include other detected objects in the environment to help us identify the action needed to open the door by providing additional relational contextual cues.

In this setting, we are interested in predicting the high-level (initial) action the robot needs to perform in order to open the door, and where this action should be applied (action point). Once these are determined, the robot can grasp the handle and execute the action. In a more advanced setting it can be imagined that we can generalise over possible grasping points depending on the

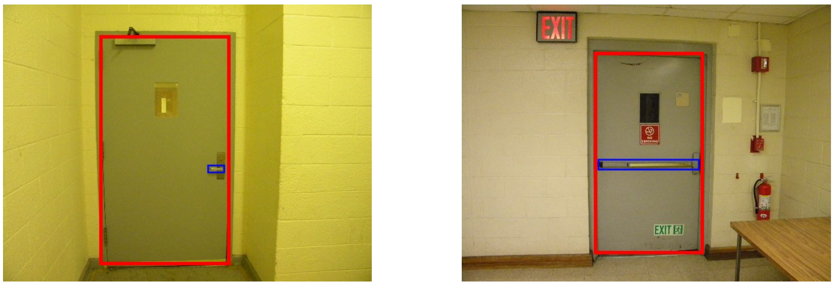

Fig. 1. Annotated doors: **(a)** push down action, **(b)** push in.

specific robot hand. Here, we are just interested in predicting the action/action point pair. We assume that the robot can open a door by pushing it in any of the six 3D directions, labelled as *in*, *out*, *left*, *right*, *up*, *down*, and turning the handle in two directions: *clockwise* and *counterclockwise*. In total, there are eight possible high-level actions. At a high-level, we think of the action point in terms of which end of the handle needs to be acted upon, so we will discretise this into 5 different values: *up*, *down*, *left*, *right* and *centre*. Later, we plan to upgrade our model with exact actionable points.

2.2 Approach

From the bounding boxes for the door frame and handle we obtain a set of positions in the $x - y$ plane for both the door and handle: $(x_{min}, y_{min}, x_{max}, y_{max})$. Based on these we define five features $(F_1, ..., F_5)$, namely: the handle aspect ratio, the handle width relative to the door width (or handle relative width), the handle relative height, the position of the centre of the handle relative to the door frame in the x-axis and in the y-axis. We assume the features are independent and additionally, we discretise them (e.g., handle aspect ratio can take the values: *big-width*, *small-width*, *square*, *small-height*, *big-height*).

Action prediction An initial intuition is that we can use a Naive Bayes classifier [13] in order to predict the action based on these features. Given our computed features $F_1, ..., F_5$ from the observed x and y positions of the bounding boxes of the door frame and handle and using our independence assumption, we can compute the conditional probability of an action A as: $P(A|F_1, ..., F_5) = \frac{P(A)*P(F_1,...,F_5|A)}{P(F_1,...,F_5} = \frac{P(A)*P(F_1|A)*...*P(F_5|A)}{P(F_1)*...*P(F_5)}$ [13] Then, in order to predict A, we compute the maximum a posteriori (MAP) probability estimate as: $\arg\max_A P(A) * \prod_{i=1}^{5} P(F_i|A)$.

However, in a fully propositional setting this requires the learning of many parameters, even in such a small domain with five features, taking values from a small discretised set. We propose to go towards a relational setting, where background knowledge can be used as a set of logical rules to reduce the number of parameters that need to be learnt, and thus the number of learning examples that need to be used, and at a later stage to generalise over our setting.

Action point prediction The action point is mostly determined by the type of the already predicted action and the relative position of the centre of the handle in the x-axis. Since the action and this feature are not independent, for the purpose of predicting the action point we define a Bayesian Network (BN) and learn its parameters. We augment our model with background knowledge in the form of logical rules which constrain the action point based on the action and related features. For example, a push in or out on a handle with a big aspect ratio should be done at the centre of the handle. This also helps us reduce the number of parameters that we need to learn. In order to predict the position, we just need to compute: $\arg\max_{Pos} P(Pos|A, F_1, ..., F_5)$. For both tasks we can use ProbLog to compute the probabilities.

3 Learning and Inference

In this section we describe our learning and inference setting. We use ProbLog, a probabilistic extension of the Prolog programming language, where facts are annotated with probabilities and for which several inference methods are available [5]. A labeled fact $p_i :: c_i$ signifies that each ground instance of the fact c_i is true with probability p_i. Additionally, we define the (general) background knowledge as in Prolog, with the aid of logical rules. Once our model is encoded via probabilistic facts and logical clauses, ProbLog can be used to answer probabilistic queries [7].

For learning the parameters, we use the learning from partial interpretations setting within ProbLog (or ProbLog LFI) [6]. Given a ProbLog program $T(p)$ where the parameters $p = \langle p_1, ..., p_n \rangle$ of the probabilistic labeled facts $p_i :: c_i$ in the program are unknown, and a set of M (possibly partial) interpretations $D = I_1, ..., I_M$, known as the training examples, ProbLog LFI finds the maximum likelihood probabilities $\hat{p} = \langle \hat{p_1}, ..., \hat{p_n} \rangle$ such that $\hat{p} = \arg\max_P P(D|T(p)) = \arg\max_P \prod_{m=1}^{M} P_w(I_m|T(p))$, where $P_w(I)$ is the probability of a partial interpretation $I = (I^+, I^-)$ with the set of all true atoms I^+ and the set of all false atoms I^-. ProbLog LFI is also able to learn parameters in the case of partial observations, which is useful to generalise over the cases when the door or handle is not fully observed.

For our prediction task, we first build a ProbLog model of the Naive Bayes classifier, which we augment with logical rules reflecting our background knowledge. In this way, we generalise by reducing the number of parameters to learn. For example, a possible rule which can be used is that if the handle relative width or height is large, the action that needs to be performed is either a push *in* or a push *out* (e.g., Figure 1(b)). This can be encoded in ProbLog in the following way:

$\mathtt{handleRelativeWidth(l)} \leftarrow (\mathtt{A = in; A = out}), \mathtt{hrw(l, A)}, \mathtt{action(A)}.$

where $\mathtt{hrw(l, A)}$ is a probabilistic fact representing the conditional probability of the handle relative width given the action. The model can also be extended later with background contextual knowledge gathered from the environment, like the presence of other objects near the door which could give an indication about how to open it. Furthermore, ProbLog allows us to add priors (e.g., Gaussian) on the x and y axis positions of the detected door frame and handle to model uncertainty in object detection.

We augment this model by encoding the ProbLog model of the BN associated with the action point, together with the respective set of background knowledge in the form of logical rules, which enable us to generalise. For example, any turn action requires the robot to perform a caging grasp of the knob, so grabbing the handle at the centre. This can be encoded as:
$\mathtt{actionpoint(centre)} \leftarrow (\mathtt{A} = \mathtt{turn\_clock}; \mathtt{A} = \mathtt{turn\_counter}), \mathtt{action(A)}.$
Once the parameters are learnt using ProbLog LFI, we can use the ProbLog program for doing inference. We can compute the MAP probability estimate of the action, and afterwards of the action point.

4 Experiments and Preliminary Results

For the purpose of our initial experimental setup, we collected a set of 60 door images. Most of these were taken from the Caltech and Pasadena Entrances 2000 dataset[4] and from the CCNY Door Detection Dataset for Context-based Indoor Object Recognition [15]. To increase variation in the different types of doors and handles we also added a few images from a Google image search. The images were manually annotated with the bounding boxes for the door and handles, as well as the action needed to open the door and the action point. We randomly split this dataset into two sets of 30 images, one to be used for training the ProbLog model by running ProbLog LFI to learn parameters, and one for testing by running inference to make predictions about the action and action point. We ran the experiment five times with different train and test sets and averaged the results. The results of the experiments are summarised in Table 1.

Table 1. Action and action point prediction.

Prediction Task	Total experiments	Avg. Success	Percentage
Action	30	23.6	78.67%
Action Point	30	23	76.67%

The results are promising for our preliminary experiment. We plan to extend our model with the ideas suggested in this paper and perform more extensive experiments that would extend the relational domain of our initial setting.

5 Conclusion and Future Work

We described an initial brief account of using SRL for predicting the action a robot needs to perform in order to open doors. Although the initial model

[4] Available at: http://www.vision.caltech.edu/html-files/archive.html

is limited, there are future ideas on adding more context by considering other objects in the environment and extending the action point prediction to consider multiple interest points on the handle which are relationally related. These can be obtained by automatically detecting different grasping points for the handle [9]. Additionally, we plan to include probabilistic priors on the door frame and handle positions to model real-world detection uncertainty. Furthermore, our model can be extended with a temporal relational aspect to generalise over opening doors that need a sequence of actions (e.g., first push the handle down, then pull the door). Our final goal is to use a realistic simulator, common in the robotics community, where multiple robotic hands can be used to open different doors.

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

References

1. L. Antanas, M. van Otterlo, J. M. Oramas, T. Tuytelaars, and L. De Raedt. A relational distance-based framework for hierarchical image understanding. In *ICPRAM (2)*, pages 206–218, 2012.
2. M. Blum, J. T. Springenberg, J. Wulfing, and M. Riedmiller. A learned feature descriptor for object recognition in rgb-d data. In *2012 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1298 –1303, May 2012.
3. L. Bo, K. Lai, X. Ren, and D. Fox. Object recognition with hierarchical kernel descriptors. In *CVPR*, pages 1729–1736, 2011.
4. L. De Raedt and K. Kersting. Probabilistic inductive logic programming. In *Prob. Ind. Log. Progr.*, pages 1–27, 2008.
5. L. De Raedt, A. Kimmig, and H. Toivonen. ProbLog: A probabilistic Prolog and its application in link discovery. In *IJCAI*, pages 2462–2467, 2007.
6. B. Gutmann, I. Thon, and L. De Raedt. Learning the parameters of probabilistic logic programs from interpretations. In *ECML*, 2011.
7. A. Kimmig, B. Demoen, L. De Raedt, V. S. Costa, and R. Rocha. On the implementation of the probabilistic logic programming language ProbLog, 2011.
8. N. Landwehr, K. Kersting, and L. De Raedt. nFOIL: Integrating Naïve Bayes and FOIL, 2005.
9. P. Moreno, J. Hrnstein, and J. Santos-Victor. Learning to grasp from point clouds. *VisLab Technical Report*, 2011.
10. M. Quigley, S. Batra, S. Gould, E. Klingbeil, Q. V. Le, A. Wellman, and Andrew Y. Ng. High-accuracy 3D sensing for mobile manipulation: Improving object detection and door opening. In *ICRA*, pages 2816–2822. IEEE, 2009.
11. L. De Raedt. *Logical and Relational Learning.* Springer, 2008.
12. L. De Raedt and I. Thon. Probabilistic rule learning. In *ILP*, pages 47–58, 2010.
13. S. Russel and P. Norvig. *Artificial Intelligence: a Modern Approach.* Prentice-Hall, 1995.
14. B. Steder, R. B. Rusu, K. Konolige, and W. Burgard. Point feature extraction on 3D range scans taking into account object boundaries. In *ICRA*, pages 2601–2608, 2011.
15. X. Yang, Y. Tian, C. Yi, and A. Arditi. Context-based indoor object detection as an aid to blind persons accessing unfamiliar environments. In *ACM Multimedia.* ACM, 2010.