Inverse Reinforcement Learning for Following Instructions

Nicholas Roy
CSAIL
MIT
Cambridge, MA 02139

Thomas Kollar
CSAIL
MIT
Cambridge, MA 02139

Stefanie Tellex
Media Lab
MIT
Cambridge, MA 02139

Introduction

In order to achieve higher levels of autonomy, robots need the ability to interact naturally with humans in unstructured environments. One of the most intuitive and flexible interaction modalities is to allow a human teammate to instruct a robot with natural language commands. In order to follow natural language directions, a robot needs to convert symbolic natural language instructions to low level actions and observations. We would like a system that can take a command from a teammate such as “Follow me to the kitchen” and generate a sequence of actions that corresponds to the desired motion through the environment.

Our approach to solving this problem is inspired by inverse optimal control or inverse reinforcement learning [8]. Inverse RL uses a corpus of execution traces for a specific action generated by an external signal to learn a model that can be used for decision making in the future. For example, watching the execution traces of a human teammate can be used to recover the teammate’s reward function [7]. However, unlike inverse reinforcement learning, we would like to learn a model that is conditioned on the semantic content of the human instructions so that for example, the planner is using a reward function that is dependent on the instruction “Follow me to the kitchen”.

From Commands to Behavior

If C is a cost function, a_i are actions, s_i are states of the world, L is the natural language command, and D is the parameters of the model, then following instructions is the optimum of a cost function,

$$\underset{a_1 \ldots a_T}{\operatorname{argmin}} \; C(a_1 \ldots a_T | L; D) \tag{1}$$

$$\text{where} \quad C(a_1 \ldots a_T | L; D) \triangleq log(p(s_1 \ldots s_T | L; D)) \tag{2}$$

We assume that actions of the robot following instructions are deterministic, allowing us to map the action sequence $a_1, \ldots a_T$ to a trajectory or state sequence $s_1, \ldots s_T$.

The first challenge is that conditioning on L introduces a problem of data sparsity; natural language instructions can have almost arbitrary structure and almost arbitrary vocabulary. Using inverse reinforcement learning to learn a likelihood model of arbitrary language will require an intractable amount of training data. To overcome this difficulty, we have shown previously [3] that we can use a shallow semantic structure called the *spatial description clause* (SDC) [2, 4, 5] to parse the instructions L to a grammar composed of clauses consisting of a verb v_k, e.g. “go”, “meet”, “follow”, coupled with a spatial relation sr_k, e.g. “past”, “through”, “towards” and a landmark l_k, e.g., “kitchen”, “office”, “street corner”. The verb describes the intended action, and the spatial relation and the landmark describe where the action should take place. If $L = \{\text{sdc}_1, \ldots, \text{sdc}_K\}$, then we can write

$$p(s_1 \ldots s_T | L; D) = \prod_{k=1}^{K} \sum_{\Phi_k} p(\text{sdc}_k | s_{\phi_{k,1:n}}; D) p(\phi_k) p(s_1 \ldots s_T; D), \tag{3}$$

where each sdc_k is given as

$$p(\text{sdc}_k | s_{\phi_{k,1:n}}; D) = p(sr_k, l_k | s_{\phi_{k,1:n}}; D) \cdot p(v_k, l_k | s_{\phi_{k,1:n}}; D), \tag{4}$$

and $\phi_{k,1:n}$ assigns a subset of n states from $s_{1:T}$ to sdc_k. In essence, $s_1, \ldots s_T$ describes a trajectory of the robot; $p(v_k, l_k | s_{\phi_{k,1:n}}; D)$ describes the likelihood of a piece of the trajectory given the verb in the kth SDC, and $p(sr_k, l_k | s_{\phi_{k,1:n}}; D)$ describes the likelihood of a piece of the trajectory occurring in a specific location of the environment given the spatial relation and the landmark. The prior $p(s_1, \ldots s_T; D)$ enforces the sequential nature of instructions; for example, state sequences that are not contiguous are lower probability, although not zero probability because human directions may skip steps, or may even have a different representation of space. By restricting the linguistic structure to SDCs, we are able to factor the original distribution and learn models for the individual terms.

In order to learn models of verbs, e.g., “Go” for navigation, “Meet” to approach another moving entity in the environment, we collect a set of example trajectories, extract features of the trajectory and use standard supervised learning to build a generative model. Similarly, to learn models of spatial relations such as “towards” or “through”, we again extract features from example trajectories and use standard supervised learning to build a generative model.

Each trajectory is modelled in relation to a specific landmark, e.g., “the kitchen” or ”the streetcorner”. We assume the robot has the ability to recognize a set of predefined concepts, using a pre-trained parts-based object recognizer [1]. However, the specific landmark used by the human teammate may be an unknown synonym for a known concept, or may not even be a known concept. However, the relation between unknown words and known words can be learned from large databases. In practice we use the tags of Flickr images to learn the correlations between known and unknown concepts; image tags have the property that concepts that are correlated in images tend to be correlated in the physical world. Therefore, given an unknown landmark in a set of instructions (e.g., “microwave”) that correlates highly with a known concept (e.g., “kitchen”) in image tags, we learn to expect the new concept in the same locations we see the known concept.

A second challenge is that the cost function depends on the likelihood of each spatial description clause sdc$_k$, and in turn depends on an unbounded number of states $s_{\phi_{k,1:n}}$. As a result, the planning problem is not first-order Markov, and the reward function may in fact be of arbitrary order, which precludes standard planning techniques such as dynamic programming. However, since the reward function is the log-likelihood of the instructions, the planning problem is equivalent to inference in a loopy (possibly fully connected) undirected graph, and akin to [6] we can apply standard inference techniques to solve the planning problem.

Preliminary results shown in table 1 indicate good performance on some commands, such as simply following directions to navigate through an environment (90%). Bringing another person to a specific location performs less well. ‘bring” involves more complex event structure than other verbs: the robot must first approach the person, then take them somewhere else. This structure is difficult for a simple feature-based classifier to model. An example output of one step of the inference appears in Figure 1, for “Meet the person at the kitchen.” The system uses inferences over possible paths of the robot and the person to find the most likely path according to our model, and takes the corresponding action. It does this at each timestep, so the overall output is a trajectory for the robot through the environment, replanning at each step in response to the person’s actions.

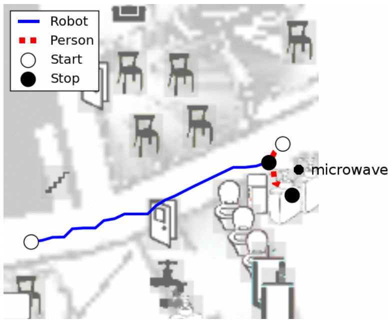

Fig. 1: Top scoring plan for “Meet the person at the kitchen.” The system searches over action sequences the robot could take, combined with possible trajectories for the person. It outputs the most likely pair of trajectories for the robot and person.

Go	Follow	Avoid	Meet	Bring	**Overall**
90%	80%	78%	70%	29%	**69**%

Table 1: Accuracy of our algorithm for various verbs, over a corpus of 46 commands.

References

[1] P. Felzenszwalb, D. Mcallester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *CVPR-2008*, June 2008.

[2] Ray S. Jackendoff. *Semantics and Cognition*, pages 161–187. MIT Press, 1983.

[3] Thomas Kollar, Stefanie Tellex, Deb Roy, and Nicholas Roy. Toward understanding natural language directions. In *Proceedings of the ACM/SIGART International Conference on Human-Robot Interaction*, Osaka, Japan, 2010.

[4] Beth Levin. *English Verb Classes and Alternations: A Preliminary Investigation*. University Of Chicago Press, September 1993.

[5] Leonard Talmy. The fundamental system of spatial schemas in language. In Beate Hamp, editor, *From Perception to Meaning: Image Schemas in Cognitive Linguistics*. Mouton de Gruyter, 2005.

[6] M. Toussaint and A. Storkey. Probabilistic inference for solving discrete and continuous state Markov Decision Processes. In *Proceedings of the 23rd international conference on Machine learning*, page 952. ACM, 2006.

[7] Brian Ziebart, Nathan Ratliff, Garratt Gallagher, Christoph Mertz, Kevin Peterson, J. Andrew (Drew) Bagnell, Martial Hebert, Anind Dey, and Siddhartha Srinivasa. Planning-based prediction for pedestrians. In *Proc. IROS 2009*, October 2009.

[8] Brian D. Ziebart, Andrew Maas, J. Andrew (Drew) Bagnell, and Anind K. Dey. Maximum entropy inverse reinforcement learning. In *Proceeding of AAAI 2008*, July 2008.

Topic: Control
Preference: oral/poster