The Importance of Being Discrete: Learning Actions through Interaction

Gary King GWKING@CS.UMASS.EDU
Tim Oates OATES@CS.UMASS.EDU
Computer Science Department
University of Massachusetts, Box 34610, Amherst, MA 01003-4610

Abstract

A robotic agent experiences a world of continuous multivariate sensations and chooses its actions from a continuous action space. Therefore, hand-coding knowledge sufficient for successful planning in uncertain, dynamic environments is a difficult task. We present a method whereby an unsupervised robotic agent learns to discriminate discrete actions out of its continuous action parameters. These actions are discriminated because they lead to qualitatively distinct outcomes in the robot's sensor space. Once found, these actions can be used by the robot as primitives for planning and further exploration of its world. We present results gathered using a Pioneer 1 mobile robot.

1. Introduction

Robots typically have continuous state and action spaces but effective planners typically require discrete states and actions. For example, the Pioneer 1 mobile robot has a pair of independent drive wheels and a variety of sensors including seven sonars and a CCD camera. To move, the robot must select a speed for its right and left wheels from an infinite range of possible parameters. While it acts, the values returned from its sonars, camera and other sensors will transition through a subset of an infinite number of states. As far as the robot can tell, every one of its possible wheel speed settings is a different action and every one of its distinct sensor readings is a different state. If a planner was forced to plan using these as primitive actions and states, it would obviously be unable to devise any useful plans in a reasonable amount of time.

Of course, many of these wheel speed settings lead to qualitatively similar outcomes. The robot will go forward, backwards, turn left or right or not move at all. We can examine the robot's behavior and categorize its actions because *we* have already categorized these continuous domains into discrete chunks. However, providing a robot with knowledge of our categories by hand-coding primitive actions and states is tedious, error prone, and must be tuned to each particular model of robot. Lastly, since the robot's sensing and effecting abilities are not equivalent to our own, we may be unable to provide distinctions which are optimally effective for the robot as it attempts to interact with and control its environment.

Below, we present a method whereby an unsupervised robotic agent can learn qualitively distinct regions of the parameters that control its actions. In our model, the robot begins with a finite number of distinct controllers, each of which is parameterized over zero or more dimensions. Using our method, a robot will be able to learn for itself which regions of the parameter spaces of its controllers lead to what sensory outcomes. These regions can then become the discrete primitive actions which the robot can use to plan. The layout of the paper is as follows: we first describe our robotic agent–the Pioneer 1 mobile robot–and the primitive controllers we created for it; then we describe our method and the experimental results that validate it. Lastly, we set this method in the context of existing and future work.

2. Method

We can view the sensor data collected by the robot as being generated by distinct activities or processes. For example, a process may involve the robot going forward, turning to the right, spinning sharply left or doing nothing at all. Our problem falls into two pieces. The first is to take the set of continuous multivariate time series generated by the robot and discover the distinct activities which created it and which activity generated which time series. In essence we want to discover how many different kinds of things the robot did and which thing goes with which time series. The second problem is to use this information to divide the parameter space(s) of the controller(s) that generated each activity into possibly overlapping regions. These regions build upon the robot's innate controllers and we use them to form the robot's primitive actions.

2.1 Framework

Although the method we propose is quite general, we explicate it in the context of our experimental work with the Pioneer mobile robot. We provide a robot with three distinct controllers:

- $\Psi_{RL}(r, l)$–a left-right wheel speed controller. By varying r and l, the robot sets its right and left wheel speeds.
- $\Psi_{\emptyset}$–a null controller that does nothing.
- Ψ_{OS}–a controller designed specifically to seek out and move the robot towards open space.

We then let the robot randomly select controllers and parameters and execute them for a brief time–typically between 10 and 20 seconds. The data recorded by the robot during each experience is saved along with the controller type and its parameters, if any. We call the complete set of robot experiences $\mathcal{E}$. Note that qualitatively distinct controller/parameter settings should generate trajectories of qualitatively distinct sensor readings as outcomes. For example, going forward will typically cause the forward facing sonar's distances to go down, the sizes of objects in the visual field to grow and the translational velocity to be positive. Other actions will produce very different readings. The next section describes how we can learn which of these sensor time series are associated with the different kinds of activities in which the robot engages.

2.2 Learning Distinctive Outcomes for a Controller

Given $\mathcal{E}$, we search for distinctive outcomes by first uniformly sampling fixed length subsequences of length L, called L-sequences, from the data. We then form k clusters from the L-sequences using Dynamic Time Warping (DTW) (Sankoff & Kruskal, 1983) as a distance measure. DTW is a generalization of classical algorithms for comparing discrete sequences (e.g. minimum string edit distance (Cormen et al., 1990)) to sequences of continuous values. The k centroids of the clusters found, C_i, partition the space of L-sequences, with each centroid standing in for all of the L-sequences that are most similar to it. In effect, the centroids discretize the continuous sensor space and form an alphabet which can be used to tokenize any other experience.

We next divide $\mathcal{E}$ into two sets for each controller: one set contains experiences that occurred while the controller was running; the other experiences that occurred while some other controller was running. For each centroid, we can determine the probability that C_i occurred when the controller was running, $p(C_i|\Psi)$, and the probability that C_i occurred when the controller was not running, $p(C_i|\overline{\Psi})$. If $p(C_i|\Psi)$ is significantly different from $p(C_i|\overline{\Psi})$ then the centroid is *distinctive* for Ψ. Centroids that occur more frequently than by chance (under the null hypothesis that the occurrence does not depend on the controller) are called positively distinctive centroids for Ψ and are denoted by $\Psi(C_i)^+$. Centroids that occur less frequently are negatively distinctive centroids and are denoted by $\Psi(C_i)^-$. Centroids which are neither positively nor negatively distinctive are said to be neutral with respect to the controller. As positively distinctive centroids occur more often in the presence of Ψ, we infer that Ψ causes them: that the sensor trajectories similar to $\Psi(C_i)^+$ are the outcomes of running Ψ. Typically, the inference that a causes b requires that a and b covary, that a occurs before b and that other potential causes of b are controlled (Suppes, 1970). As our method does not account for the last item, some of the causal inductions will be incorrect and further effort will need to go into resolving them.

2.3 From Distinctive Outcomes to Distinctive Actions

For each centroid in $\Psi(C_i)^+$, we examine the experiences in $\mathcal{E}$ and see if the centroid occurs more frequently than by chance. We accomplish this by comparing the number of occurrences of L-sequences similar to the centroid in the experience to that expected given the overall population density of the centroid in $\mathcal{E}$. If C_i occurs frequently in an experience, then we say that the experience is distinctive for the centroid. The set of distinctive experiences for each centroid is $\mathcal{E}_{C_i}$. We will denote the parameters of the distinctive experiences for a centroid as P_{C_i}. We can plot P_{C_i} for each controller colored by the centroid. For example, figure 1 shows one particular division of Ψ_{RL}'s parameter space. This plot shows left and right wheel speed parameters associated with data collected from the Pioneer-1 while running Ψ_{RL}. Each of these robot experiences is labeled with one of six distinctive centroids. For example, the experiences labeled with the small x's all have wheel speeds that are generally below

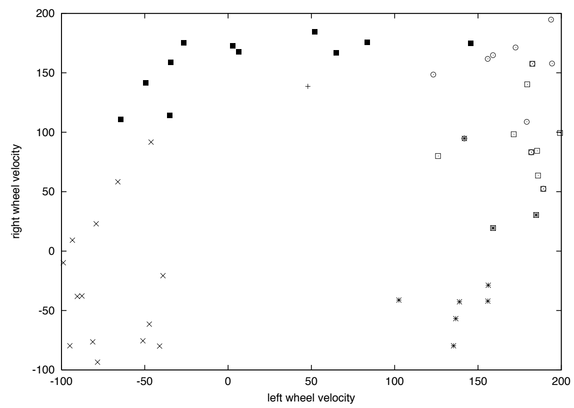

Figure 1. Scatter-plot of Left and Right wheel velocities labeled by centroid.

zero. The center portion of the plot is empty because our method did not find any distinctive outcomes for these experiences. Notice that each of the prototypical centroids is associated with a subset of the entire parameter space and that the subsets appear to be well separated.

In general, there are several possible outcomes for the distributions of controller parameters derived from individual centroids C_i and from pairs of centroids C_j and C_k. We first list the possibilities and then provide intuitions for their meanings:

1. P_{C_i} has a uniform distribution across the entire parameter space.
2. P_{C_i} has a non-uniform distribution–some parameter values lead to C_i more frequently than others. This distribution may be uni-modal, bimodal or more complex.
3. P_{C_j} and P_{C_k} are well separated (note that this can only occur if the individual distributions are non-uniform to begin with).
4. P_{C_j} and P_{C_k} overlap significantly.

We will formalize these notions below but the intuitions should be clear. In the concrete terms of $\Psi_{RL}(r, l)$, item 1 indicates that although the outcome occurs more frequently when Ψ_{RL} is running, it does not depend on the parameters of Ψ_{RL}. Item 2 indicates that the occurrence of the centroid depends on r and l. If the distribution is uni-modal, then only one range of r and l leads to this outcome; if it is more complex, then two or more ranges lead to it. This corresponds to a different regions of the parameter space having the same outcome.

Items 3 and 4 both require that the outcomes C_j and C_k depend on the choice of r and l. If the parameter ranges for the two outcomes overlap significantly, then this corresponds to a single action leading to two (or more) different outcomes. This may be due to the context in which the two action occurs.

2.4 Knowing when an action is discrete

We can divide the parameter space of a controller into uniform cells and create a histogram of the number of occurrences of P_{C_i} in a cell. We can create a similar histogram of the total number of experiences with parameters in a cell regardless of centroid. We can use these histograms to form a discrete probability distribution of the probability that a given range of parameters leads to the distinctive outcome (C_i). We wish to determine if the distribution is significantly different from that expected by random chance. The null hypothesis is that the parameter values have no effect on the outcomes and that the distribution obtained from P_{C_i} is uniform. We can test H_0 for each C_i by building a sampling distribution of the Kullback-Leibler distances between randomly generated distributions of the number of experiences containing C_i. elements and the true uniform distribution. The discrete Kullback-Leibler distance or average variance measures how much one probability distribution differs from another:

$$d(p_1, p_2) = -\sum_x p_1(x) ln \frac{p_1(x)}{p_2(x)}$$

Once we have obtained the distribution of the distance measures, we can use randomization testing to see if the actual distribution derived from P_{C_i} is significant.

If P_{C_i} is significantly different from the non-uniform distribution, then we can use randomization testing again on each of the cells in the distribution. In this case, we build the sampling distribution for the cells of the histogram using the Kullback-Leibler distance of the probability value in each cell as compared to the uniform probability distribution. We then look for cells whose Kullback-Leibler score is significantly different from that expected under H_0. These cells are the ones who contribute highly to P_{C_i}'s significance. They define the discrete action which leads to outcome C_i.

2.5 Summary of the Method

In summary, our method is as follows. Given a set of parameterized controllers for a mobile robot and a set of sensors:

1. Randomly select a controller and run it with randomly selected parameters. While it is running, record the data that it generates and save this along with the type of controller and its parameter values.
2. Sample fixed length subsequences uniformly from the data generated and form clusters.
3. For each cluster centroid, C_i, and controller, Ψ, determine if the probability of the centroid occurring while Ψ is running, $p(C_i|\Psi)$, differs significantly from the probability of the centroid occurring while Ψ is *not* running, $p(C_i|\overline{\Psi})$.
4. Determine the distinctive experiences for each of Ψ's positively distinctive centroids. Use these to create probability distributions for P_{C_i}, the parameters of the experiences that lead to outcome C_i.
5. Use randomization testing and the discrete Kullback-Leibler distance to find centroids that are dependent on the parameters of Ψ and the regions of the parameter-space that lead to the centroid.

The regions found are ranges of parameter values that typically result in specific outcomes of sensory trajectories. They are candidates for primitive actions of the mobile robot.

3. Experiment

3.1 Method

We collected 120 experiences using $\Psi_{RL}(r,l)$ (96-experiences), $\Psi_{\emptyset}$ (12-experiences) and Ψ_{OS} (12-experiences). The distribution was weighted towards Ψ_{RL} as this controller was the focus of our experiment. The r and l parameters for Ψ_{RL} were uniformly sampled between -100 and 200 so as to obtain more forward-moving experiences than backward-moving experiences. The robot operated in a convex space containing numerous small objects with which it could interact. Intervention was required once during the data collection when the robot became stuck on power conduit lines attached to one of the walls of the space.

In the analysis that follows we used the following subset of sensors: heading, right-wheel-velocity, left-wheel-velocity, translational-velocity and rotational-velocity. The Pioneer keeps track of its heading and assumed position by dead reckoning. It determines its right and left wheel velocities, translational and rotational velocities via proprioceptive feedback from its wheel encoders. The values of its sensors are recorded every 10-milliseconds.

3.2 Results

The algorithm described above found several statistically significant ($p < 0.01$) regions of the parameter space of $\Psi_{RL}(r,l)$ including ones that we would label "forward", "backwards", "hard-left", "slow-left" and so forth. Figure 2 below demonstrates several probability distributions linking particular setting of left and right wheel speeds and their distinctive outcomes (C_i).

Each plot in figure 2 shows the action associated with a particular distinctive outcome. The darker cells of each plot indicate the range of parameters that define the action. The first plot shows the action defined by high values of left and right wheel speeds and with the right wheel speed generally higher than the left wheel speed–with what we would label forward motion and turning to the right. Investigation of the distinctive centroid associated with the plot confirms this interpretation. The second plot shows actions with right wheel speeds below zero and the third shows actions with high left wheel velocities and low right wheel velocities. We might label these activities as "backwards to the left" and "forward left turn" respectively. Of course, each atomic action discovered by our method ranges over a large portion of the controller's parameter space. This is due in part to the limited amount of data collected and in part to the noisy environment in which the robot runs. We expect that additional data would allow the atomic actions to become more precise.

We have shown that our method allows an unsupervised mobile robot to interact with its environment and learn discrete actions over the parameter spaces of its controllers.

4. Related and Future Work

The problem of learning action models for the purpose of planning is studied in a variety of forms. Much of this work focuses on simulated domains and assumes discrete state and action spaces and deterministic outcomes of actions (Gil, 1992; Wang, 1995), though some allows for the possibility of probabilistic outcomes (Benson, 1995; Oates & Cohen, 1996). One notable exception is (Pierce, 1995), which describes a method for learning action models given continuous state and action spaces for a simulated robot with noisy sensors.

In stochastic domains with continuous states and

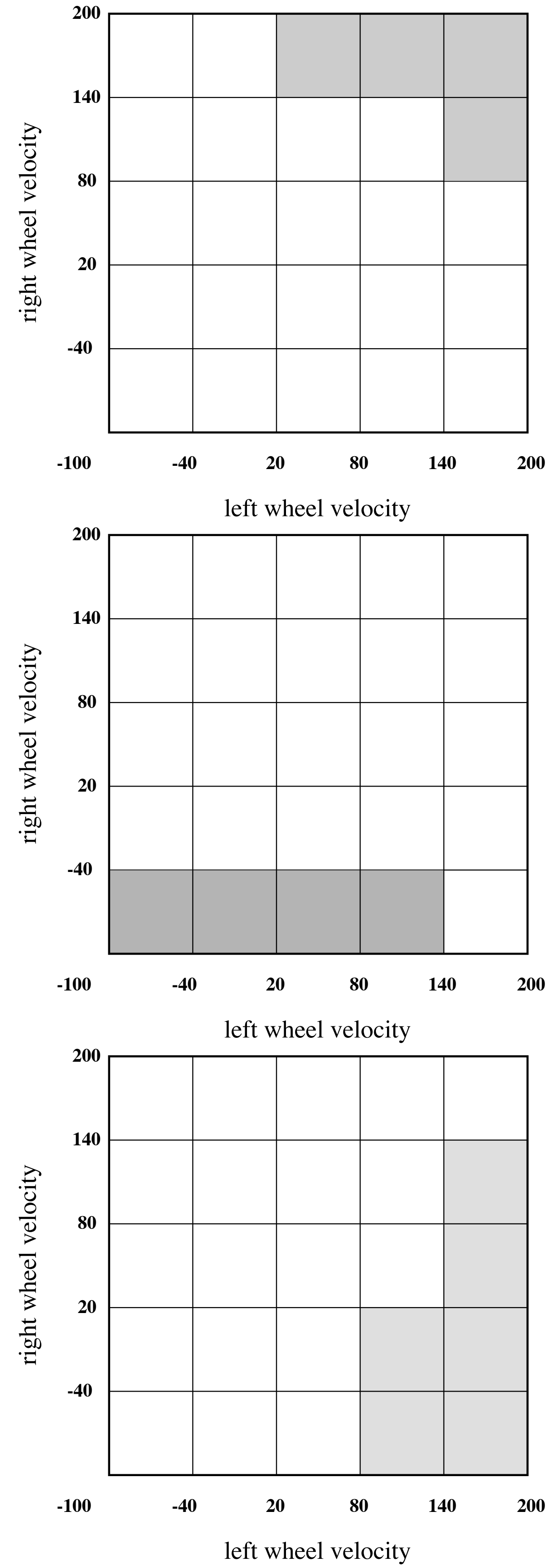

Figure 2. Discrete actions associated with different distinctive outcomes.

discrete actions, reinforcement learning methods can learn reactive control policies (Mahadevan & Connell, 1992), and recent work in this area addresses the case in which both the state and action spaces are continuous (Santamaria et al., 1998). Reinforcement learning has also proven to be effective both in simulated domains and with physically embodied robots. Our work differs from these approaches in that the goal is to learn a declarative action model suitable for use by symbolic planning algorithms (and other cognitive tasks such as natural language generation and understanding (Oates et al., 2000)), not opaque, non-symbolic policies.

Our representation of outcomes as prototypical time series is based on earlier work on clustering time series (Oates, 1999). Several other recent approaches to identifying qualitatively different regimes in time series data include (Agrawal et al., 1995; Cohen et al., 1999; Keogh & Pazzani, 1998).

Future work will remove a number of limitations of the current method. In particular, rather than representing outcomes of actions as fixed-length prototypes, we will apply the algorithm described in (Oates, 1999) to identify and represent outcomes of variable duration. Also, having identified discrete actions and their outcomes, it becomes possible to go back to the time series data and search for features of the environment that condition the outcome probabilities. In terms of classical planning operators, we will identify preconditions. Another limitation of the current method is that sensor groups are pre-specified. Ideally, the robot would determine which sets of sensors should be grouped together because patterns in those sensors capture outcomes of invoking actions. We plan to explore the utility of a simple generate and test paradigm to this problem, with the test phase involving statistical hypothesis tests of the form previously described.

5. Acknowledgments

This research is supported by DARPA contract DASG60-99-C-0074 and DARPA/AFOSR contract F49620-97-1-0485. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright notation hereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements either expressed or implied, of the DARPA or the U.S. Government.

References

Agrawal, R., Lin, K., Sawhney, H. S., & Shim, K. (1995). Fast similarity search in the presence of noise, scaling and translation in time series databases. *Proceedings of the 21st International Conference on Very Large Databases.*

Benson, S. (1995). Inductive learning of reactive action models. *Proceedings of the Twelfth International Conference on Machine Learning* (pp. 47–54).

Cohen, P. R., Ramoni, M., Sebastiani, P., & Warwick, J. (1999). Unsupervised clustering of robot activities: A bayesian approach. To appear in *Proceedings of the Fourth International Conference on Autonomous Agents.*

Cormen, T. H., Leiserson, C. E., & Rivest, R. L. (1990). *Introduction to algorithms.* The MIT Press.

Gil, Y. (1992). *Acquiring domain knowledge for planning by experimentation.* Doctoral dissertation, Carnegie Mellon University.

Keogh, E., & Pazzani, M. J. (1998). An enhanced representation of time series which allows fast and accurate classification, clustering and relevance feedback. *Working Notes of the AAAI-98 workshop on Predicting the Future: AI Approaches to Time-Series Analysis* (pp. 44–51).

Mahadevan, S., & Connell, J. (1992). Automatic programming of behavior-based robots using reinforcement learning. *Artificial Intelligence*, *55*, 189–208.

Oates, T. (1999). Identifying distinctive subsequences in multivariate time series by clustering. *Proceedings of the Fifth International Conference on Knowledge Discovery and Data Mining* (pp. 322–326).

Oates, T., & Cohen, P. R. (1996). Searching for planning operators with context-dependent and probabilistic effects.

Oates, T., Eyler-Walker, Z., & Cohen, P. R. (2000). Toward natural language interfaces for robotic agents: Grounding linguistic meaning in sensors. To appear in *Proceedings of the Fourth International Conference on Autonomous Agents.*

Pierce, D. M. (1995). *Map learning with uninterpreted sensors and effector.* Doctoral dissertation, University of Texas, Austin.

Sankoff, D., & Kruskal, J. B. (Eds.). (1983). *Time warps, string edits, and macromolecules: Theory and practice of sequence comparisons.* Reading, MA: Addison-Wesley Publishing Company.

Santamaria, J. C., Sutton, R. S., & Ram, A. (1998). Experiments with reinforcement learning in problems with continuous state and action spaces. *Adaptive behavior*, *6*, 163–218.

Suppes, P. (1970). *A probabilistic theory of causality.* Amsterdam: North Holland.

Wang, X. (1995). Learning by observation and practice: An incremental approach for planning operator acquisition. *Proceedings of the Twelfth International Conference on Machine Learning.*