Crowdsourcing for closed-loop control

Sarah Osentoski
Department of Computer Science
Brown University
Providence, RI 02912
sosentos@cs.brown.edu

Christopher Crick
Department of Computer Science
Brown University
Providence, RI 02912
chriscrick@cs.brown.edu

Grayin Jay
Department of Computer Science
Brown University
Providence, RI 02912
tjay@cs.brown.edu

Odest Chadwicke Jenkins
Department of Computer Science
Brown University
Providence, RI 02912
cjenkins@cs.brown.edu

Abstract

We present a system for large scale robotic learning from demonstration. We describe a set of software tools for enabling human-robot interaction over the internet and gathering the large datasets that such crowdsourcing makes possible. We show results in which humans teach a robot to navigate a maze over the Internet.

Robots occupy a peculiar place in our culture. We have been building robots in our imaginations for decades, robots that are alternately wondrous or terrifying, always brilliant and consummately skilled. In contrast, real robots are typically brittle and capable of only a few simple constrained tasks. Additionally, only expert programmers, intimately familiar with the particulars of a low-level robotic system, can hope to achieve any kind of complex robot behavior. This paper describes our efforts to apply the lessons of crowdsourcing to robotics to leverage the power and knowledge of a truly large number of end users to create more skilled and robust robot controllers.

We focus on learning from demonstration (LfD), [2, 1] an approach to robot programming in which users demonstrate desired skills to a robot. Nothing is required of the user beyond the ability to complete the task in a way that the robot can interpret. Traditionally, LfD research has been constrained by the number of demonstrations that can be performed; unless a large number of users can interact with the robot only a limited amount of data can easily be gathered. Since users do not usually need specialized skills to demonstrate robot skills, a web-enabled system could be used to collect data from a large number of users. An online system also lifts the burden of training the robot from a single user who may only want to contribute a few demonstrations. Collecting data for closed-loop control is different from many crowdsourcing applications examined in the Machine Learning community, which has typically focused on annotation of text and labeling of images. Task demonstration often requires a significant interaction both in terms of time and information provided to the user. Users not only give the robot instructions, but also evaluate the results and provide new instructions given the outcome.

We describe a recently developed system that allows a large number of users to train a robot to solve a task (in this case maze-navigation) through a video-game style interface. While a single demonstration may contain errors and provides only limited data, the demonstrations from multiple users provide enough data to create a robust policy. There have been a few initial efforts in putting robots on the Internet [4, 9]. These approaches generally allowed people to interact with robots but were not aimed at task learning. Other work examined using crowdsourcing approaches to train robots through game playing environments. Chernova et al examined using a multi-player video game where users collaborate to provide user demonstrations [3]. This work differs from ours in that it relied on real-world user interactions in a museum, rather than an online setting with a potentially global user base. Additionally, users were not providing demonstration on the actual robot. Our system allows users to actually control the robot and does not require special software or plugins other than a web browser.

1 Robot and Web Interface

The robot used in our experiments is an iRobot Create with a FitPC2 small-form-factor computer and a Sony PS3 Eye camera. The computer maintains a wireless connection to the Internet allowing user interactions. The robot is able to move forward, backwards, and rotate. The robot can maneuver through a maze, pictured in Figure 1a The maze has artificial reality (AR) tags placed within it as landmarks. The robot is able to detect the AR tags and use the size of the tag in its visual perception to estimate its distance from each tag. The tags, along with the bump sensors that detect collisions with walls, represent the perceptual space of the robot.

The system builds upon ROS [7], Willow Garage's robot middleware system and leverages rosjs [6], a lightweight Javascript binding for ROS. rosjs exposes the robot's functionality and sensors as web services, as well as providing security and visualization tools. Robot application developers and researchers can create robot controllers and interfaces in the same manner as creating web contentrosjs does not require users to install additional software or plugins beyond a web browser, allowing a large number of users the ability to access the robot. Figure 1 shows the two interfaces that were available to users to drive the robot to the goal location. The first shows the live video feed to the user. The second shows a visualization of the AR tag(s): a blue polygon represents the position of the tag within the robot's field of view. Both visualizations also provided information on whether the robot had hit an obstacle, by flashing red when a bump sensor activated. Subjects were also shown a map of the maze marked with the position of the AR tags. When users finished their demonstrations, they clicked a link and were told if they had successfully navigated the maze, as well as their performance in terms of time and number of collisions.

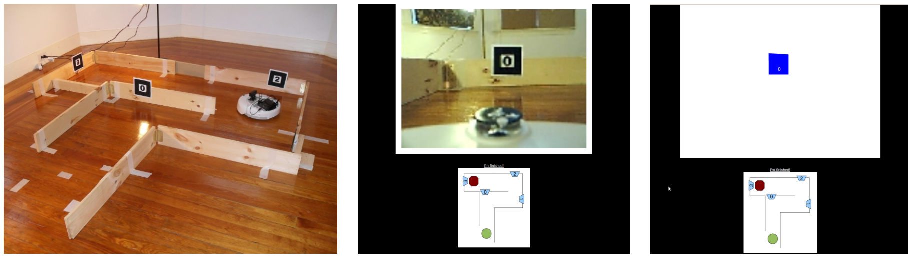


Figure 1: Users teleoperate the robot through the maze, using one of two web-based visualizations.

2 Experimental Results

132 individual subjects participated in the robot training task from all over the world. Subjects could participate as many times as they liked and a total of 276 demonstrations were collected. In the context of human-robot interaction studies, this is a very large subject pool. Our hope is that the availability of rosjs and similar services will encourage more such studies with even larger numbers of participants. We first analyzed how the difference between the two interfaces influenced the user's effectiveness in demonstrating the task to the robot. Humans were better at navigating the maze using in the video. On average, those completing the maze with the video feed were able to do so 16.03 seconds, or 36.3%, faster than those who could see the blue squares. While users with video were faster at negotiating the maze, they were *less* able to maneuver without running into the walls. Drivers with video were more than twice as likely to crash into a wall – 1.15 crashes per demonstration as opposed to 0.45. Part of this might be explainable by the fact that, since they were driving faster, they were more limited by their reaction time or by network latency, transmitting video requires more bandwidth than sending a few coordinates used by rosjs to construct a visualization.

We examined if the data from one visualization modality was more effective than the other. We used ID3 [8], a decision tree learning algorithm to construct decision trees that served as the robot's

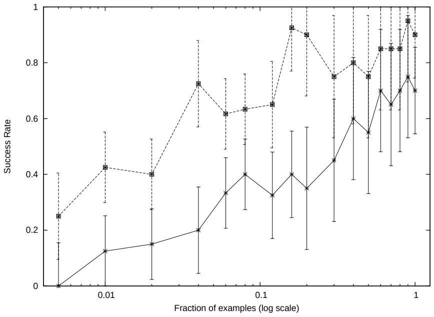
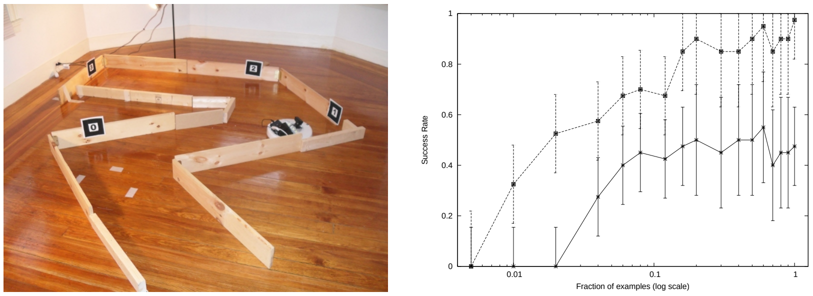


Figure 2: Robot performance as a fraction of training trials seen. Top line represents a robot trained on tag-only data; bottom line is a robot trained by people with access to video. 1320 total trials.

policy. ID3 prefers smaller decision trees and chooses to split selecting the attribute for which the entropy is minimum. We examined how robot performance was affected by the type of visualization and the percentage of training trials seen. For each trial, the robot was given access to a percentage of the training corpus and built a decision tree based upon that data. The robot was started in approximately the same position each time, facing the tag indicating the first turn about 1.3 meters away. A trial was considered successful if the robot stopped in the correct goal position, facing the final tag and less than 0.4 meters away. If the robot stopped elsewhere, or began performing repetitive sequences of actions which did not make progress toward the goal, the trial was marked a failure. Figure 2 shows the results of these experiments. A log scale is used since most of the difference occur when there are fewer training examples available. We also examined how well the training generalized to a similar but not identical task. To do this we created another maze, shown in Figure 3a. Successful navigation through the new maze required the same qualitative sequence of steps – the tags were in the same order and the turns were in the same direction as before – the maze's layout was otherwise very different. Right angles were replaced by acute and obtuse ones, and the lengths of each maze leg were changed. We allowed the robot to learn on successively larger fractions of the training data. The results are summarized in Figure 3b.



(a) A robot picks its way through a different maze from the one it was taught to navigate.

(b) Robot performance in a new maze, as a function of the fraction of training trials seen. The top line represents tag training data, bottom line is video. 900 total trials.

Figure 3: Experimental results to examine how the learned policy performs on a new maze.

Researchers in data mining and machine translation have been able to take advantage of Google's index of billions of crowdsourced documents to show that simple learning algorithms that focus upon recognizing specific features outperform more conceptually sophisticated ones [5]. We conjecture that similar results will be observed if large amounts of data can be collected for learning from demonstration. We examined how other learning algorithms performed on this dataset. We trained the classifiers on successively larger fractions of training data and tested on the remainder. Our initial results are presented in Figure 4. While no clear pattern is immediately obvious from these results, a more extensive analysis of different learning algorithms, a larger dataset, and a more complex task may help discover the effect of large scale data for learning from demonstration tasks.

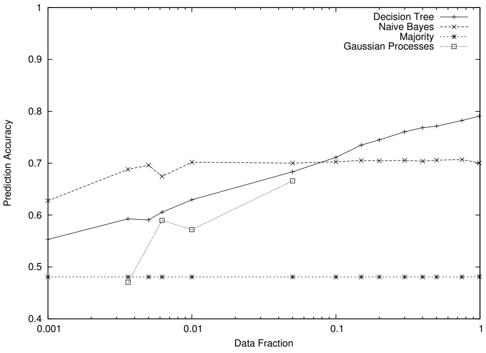


Figure 4: Comparison of the prediction accuracy of several learning algorithms as a function of the percentage of training trials.

3 Conclusions and Future Work

We presented an initial examining the use of crowdsourcing for learning for demonstration. We plan to extend this system to more complex tasks such as manipulation. We plan to study how that teams of people can jointly interact and collaboratively train the robot. Currently our work has focused upon users demonstrating predefined tasks; an exciting avenue of research is to use crowdsourcing to find new potential robotic applications that robotics research has not yet pursued.

4 Acknowledgments

This research was supported in part by the Air Force Office of Scientific Research under grant YIP: FA-9550-09-1-0206. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the AFOSR.

References

[1] B. D. Argall, S. Chernova, M. Veloso, and B. Browning. A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, 57:469–483, 2009.

[2] A. Billard, S. Calinon, R. Dillmann, and S. Schaal. *Handbook of robotics*, chapter Robot programming by demonstration, pages 1371–1394. Springer, 2008.

[3] S. Chernova, J. Orkin, and C. Brazeal. Crowdsourcing hri through online multi-player games. In *Proceedings of the AAAI 2010 Fall Symposium on Dialog with Robots*, 2010.

[4] K. Goldberg, H. Dreyfus, A. Goldman, O. Grau, M. Gržinić, B. Hannaford, M. Idinopulos, M. Jay, E. Kac, and M. Kusahara, editors. *The robot in the garden: telerobotics and telepistemology in the age of the Internet.* MIT Press, 2000.

[5] A. Halevy, P. Norvig, and F. Pereira. The unreasonable effectiveness of data. *IEEE Intelligent Systems*, March/April:8–12, 2009.

[6] S. Osentoski, G. Jay, C. Crick, and O. C. Jenkins. Brown ROS package: reproducibility for shared experimentation and learning from demonstration. In *AAAI-10 Robot Workshop*, 2010.

[7] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng. ROS: an open-source robot operating system. In *Proceedings of the Open-Source Software Workshop of the International Conference on Robotics and Automation*, 2009.

[8] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1:81–106, 1986.

[9] D. Schulz, W. Burgard, D. Fox, S. Thrun, and A. B. Cremers. Web interfaces for mobile robots in public places. *IEEE Robotics and Automation Magazine*, 7:48–56, 2000.