Project 16: Background Review

Chris Paxton

April 4, 2014

1 Introduction

I am investigating the methods by which we can use methods from inverse optimal control to improve human capabilities and assist surgical procedures. To this end, I looked at papers from groups that have looked into learning from demonstration and techniques for learning from demonstration that could be adapted to a real-world robotics task.

2 Continuous Inverse Optimal Control with Locally Optimal Examples

A major limitation in applying existing methods for inverse optimal control (IOC) to robotic examples is the high dimensionality of these examples. I am interested in addressing this by looking in to recent work by Levine et al [2]. This paper an interesting inverse optimal control paper that goes into an method for inverse optimal control on continuous valued data rather than discrete valued data, as many previous papers assume (e.g. [4]). This work also has the added advantage of using locally optimal, rather than globally optimal examples: this means that if the human is limited by clumsy controls the system might be able to learn a better approach.

2.1 Summary

The classic maximum entropy IOC problem as defined in prior work [4] is given in Equation (1). We assume that robot control is deterministic and that we have a finite horizon.

$$P(u|x_0) = \frac{1}{Z} \exp\left(\sum_t r(x_t, u_t)\right) \tag{1}$$

Computing the partition function Z is incredibly costly. To evaluate Eq. (1) without computing Z, the authors apply the Laplace approximation and model the distribution locally as a Gaussian. This corresponds to assuming that the expert performed a local, not global, optimization. Eq. (1) is rewritten as an integral in the continuous case, where $r(u)$ denotes the sum of the rewards along a given path (x_0, u).

$$P(u|x_0) = e^{r(u)} \left(\int e^{r(\bar{u})} d\bar{u}\right)^{-1} \tag{2}$$

In this case, $r(\bar{u})$ is the sum of all rewards along a different path; the equation is the sum of the log-linear rewards along one path (x_0, u) over the sum of the log-linear rewards of all possible paths $(x_0, \bar{u})$. The authors approximate this probability with a second order Taylor expansion:

$$r(\bar{u}) \approx r(u) + (\bar{u} - u)^T \frac{\delta r}{\delta u} + \frac{1}{2}(\bar{u} - u)^T \frac{\delta^2 r}{\delta u^2}(\bar{u} - u) \tag{3}$$

With the gradient $\frac{\delta r}{\delta u}$ as g and the Hessian $\frac{\delta^2 r}{\delta u^2}$ as H, Eq. (1) can be rewritten as in Eq. (4). We use the Laplace approximation to rewrite this as a Gaussian, with n as the number of dimensions in the space of actions u.

$$\begin{aligned} P(u|x_0) &\approx e^{r(u)} \left(\int e^{r(u) + (\bar{u}-u)^T g + \frac{1}{2}(\bar{u}-u)^T H (\bar{u}-u)} d\bar{u}\right)^{-1} \\ &= e^{\frac{1}{2} g^T H^{-1} g} + |-H|^{\frac{1}{2}} (2\pi)^{-\frac{n}{2}} \end{aligned} \tag{4}$$

This in turn gives us the approximate log-likelihood in Eq. (5), which can be optimized directly using any number of methods:

$$\mathcal{L} = \frac{1}{2} g^T H^{-1} g + \frac{1}{2} \log|-H| - \frac{n}{2} \log 2\pi \tag{5}$$

This equation is the primary contribution of the paper. The authors also provide an efficient method to compute the linear system $H^{-1}g$, a key part of Equation (5) and its gradient, and then examine efficient methods for computing the log-likelihood and the gradient of the log-likelihood.

It is possible to use a linear or nonlinear kernel for the reward function. If solving for a nonlinear reward function, the authors represent the reward function as a Gaussian. In this case, they jointly optimize the log-likelihood from Equation (5) plus the Gaussian kernel likelihood from Equation (6), where F is a set of inducing feature points (from the expert demonstrations) and the output of the Gaussian reward function y is learned.

$$\log P(y, \lambda, \beta|F) = -\frac{1}{2} y^T K^{-1} y - \frac{1}{2} \log |K| + \log P(\lambda, \beta|F) \tag{6}$$

Here K is the Gaussian covariance matrix such that $K_{ij} = k(f^i, f^j)$, and λ and β are the parameters of the Gaussian kernel function k given in Equation (7).

$$k(f^i, f^j, \lambda, \beta) = \beta \exp\left(-\frac{1}{2} \sum_k \lambda_k \left[(f_k^i - f_k^j)^2 + 1_{i \neq j} \sigma^2 \right] \right) \tag{7}$$

This is applied in the paper to a couple of sample problems, including a simulated driving task and a multi-robot arm movement task in two dimensions. The authors showed that they were better able to learn the underlying reward function in the arm manipulation task.

2.2 Analysis

The paper's assumptions are all reasonable to make in our test case: we have a model of the world that tells us where the robot can go, and we are trying to accomplish a single task in a short amount of time.

The authors validate their method by showing they can learn a very similar reward function to the one used to generate data; especially since this is not a widely used example problem I believe this is not a particularly convincing method to demonstrate the algorithm. The reward function they were trying to match had a number of "valleys" with low reward and a Gaussian peak, meaning that it very closely matched their own Gaussian assumptions.

The simulated driving task was more interesting. In this case the authors were able to very closely mimic held-out human demonstrations of aggressive driving, evasive driving, or tailgating other cars. I think one large problem with this paper is that results seem highly dependent on the kernel function chosen; one must choose a reward function that seems to fit the task, and fine tune parameters like the noise term σ from Equation (7).

One useful property of this work is that it looks into continuous optimal control that can use noisy, locally optimal data to find a solution; this is useful to robotic semi-automation research because it is entirely possible that the human operators will make less than optimal choices simply because they are limited by the controls.

One excellent feature of this paper is that a large amount of supplemental material is provided, including MATLAB source code, videos, and errata. This is online at `http://graphics.stanford.edu/projects/cioc/`, which would make it much easier for anyone interested in implementing or extending the algorithm. It also makes it possible to reproduce their results from ICML.

2.3 Application to Project

I believe that the continuous IOC code described here would fit well as a part of this project. It is an interesting and efficient approach that can deal with high dimensionality navigation tasks in complex environments.

As the paper points out, there are not many good options for IOC that can be applied to complex tasks with high dimensionality. Some previous work has looked into learning solutions to problems with linear dynamics and quadratic rewards (LQR); previous work in inverse reinforcement learning has largely avoided real-world robotic examples. I believe that this approach could be adapted into something useful for my project.

3 Trajectory Transfer

Recent work by Pieter Abbeel's group at UC Berkeley [3] looked at transferring learned skills between different contexts, looking at both a simulated Raven-II surgical robot and a real-world robotic example with a PR2 robot.

4 Summary

The authors look at ways to take an example trajectory from an expert and adapt it to a new context. First, they collect an example and perform a non-rigid registration between the demonstration and the task scene. Then the authors apply this transform to the demonstration trajectory, convert this end effector trajectory into joint positions for the robot to take, and execute the task by moving through this series of joint positions.

The approach is first tested with a simulated Raven-II robot. The Raven-II was remotely operated with a pair of Phantom OMNIs; the operator presses

a foot pedal to divide the task performance into segments. They used markers on the ends of the simulated suturing pad to perform the registration. A second suturing pad was perturbed by translations of 0.25 cm on the x axis and 0.5 cm on the y and z axes, and rotations of 5 degrees. The authors show that the algorithm is able to adapt the demonstrated trajectory in all of these cases.

For the PR2 experiment, a single high-quality RGBD video of a task performance was obtained and annotated. The same method was applied to a large suturing task. In this case, the system did not need to determine correspondence between different contexts: a human operator provided corresponding points.

4.1 Analysis of Paper

This is an interesting paper because it solves the same example problem I want to be able to solve. However, I believe there are a few problems with directly using this approach. It still relies on a completely optimal expert demonstrations, and it makes a very large assumption: that the context of the example performance of the task is exactly the same as the test, with the addition of a non-rigid transformation.

This approach also cannot provide feedback or generate a reward function to follow; this means that we cannot use it to assist a human trainee or surgeon who takes a slightly different path than the expert. An IOC reward function might reveal multiple paths that are the same or only slightly different in cost; the approach described by the authors of this paper only allows the exact replication of a trajectory deformed into a new context.

4.2 Application to Project

This paper is interesting because it shows a simpler and more reliable method for learning from demonstration and proves it with a real-world robotic example. While this approach does not allow the operator to adapt a trajectory to a new environment in every case, it could be combined with something like previous work from Amir Masoud in the CIRL lab [1] to learn a quadratic penalty function around a trajectory and exponential cost functions around features of the environment. If IOC proves impractical, this might be a good foundation for a robust approach for human-machine collaboration.

References

[1] Amir M. Ghalamzan E., Chris Paxton, Gregory Hager, and Luca Bascetta. Robot learning from demonstration: from imitation to emulation. 2014.

[2] S. Levine and V. Koltun. Continuous inverse optimal control with locally optimal examples. In *Proceedings of the 29th International Conference on Machine Learning, ICML 2012*, volume 1, pages 41 – 48, 2012.

[3] John Schulman, Ankush Gupta, Sibi Venkatesan, Mallory Tayson-Frederick, and Pieter Abbeel. A case study of trajectory transfer through non-rigid registration for a simplified suturing scenario. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 4111–4117. IEEE, 2013.

[4] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, and Anind K Dey. Maximum entropy inverse reinforcement learning. In *AAAI*, pages 1433 –1438, 2008.