Learning from Limited Demonstrations in High Dimensional Feature Spaces

Sasha Salter

Applied Artificial Intelligence Group, Engineering Department, Oxford University

Email: sasha@robots.ox.ac.uk

Abstract

Reinforcement learning (RL) has recently gained a lot of popularity partially due to the success of deep Q-learning (DQN) on the Atari suite and AlphaGo. In these domains reward function engineering is trivial. Many of the Atari suite video games output a score directly and in AlphaGo the reward signal is simply ±1 dependent on whether you win or lose. For the majority of real-world scenarios designing such a function is unintuitive. In the application of robotics it is desirable to automatically craft the reward such that the robot mimics human behavior for any task.

Inverse Reinforcement Learning (IRL), a supervised-learning technique, learns a cost model from demonstrations. Deep-IRL usually requires a large quantity of training examples from an expert agent whose embodiment is very similar to the learning agent. Obtaining many demonstrations from aesthetically similar robots is generally costly and time-consuming.

This project investigates a Max-Entropy IRL approach that leverages the representational power of Inception, Google's pre-trained deep convolutional neural network (CNN), through transfer learning to learn a reward function from a limited number of demonstrations (in many cases one) and an agent with a different embodiment to the one being trained (a human hand as opposed to robot arm).

Approximate Vision-Based Inverse Reinforcement Learning

Maximum Entropy is able to handle expert suboptimality as well as stocasticity by operating on the distribution over possible expert demonstration trajectories. It ensures the selection of a reward function that does not favour any particular trajectory. This is achieved by maximizing the entropy of the trajectory distribution whilst at the same time matching the observed feature expectations. In MaxEnt-IRL, the expert demonstrations τ are assumed to be drawn from a Boltzmann distribution according to:

$$p(\tau) = p(s_1,\ldots,s_T) = \frac{1}{Z}\exp\left(\sum_{t=1}^{T} R(s_t)\right)$$

Computing the partition function, Z, for high dimensional feature spaces is unfeasible. [1] overcomes this problem by assuming independence for both feature and time dimensions, resulting in a naïve Bayes IRL model that is bias but generalizes well to new and unseen environments.

$$p(\tau) = \prod_{t=1}^{T}\prod_{i=1}^{N} p(s_{it}) = \prod_{t=1}^{T}\prod_{i=1}^{N} \frac{1}{Z_{it}}\exp\left(R_i(s_{it})\right)$$

The ability to generalize well with few demonstrations (including one-shot learning) lends to the use of Google's pre-trained CNN, Inception, for features s_{it}. The activations of several layers within the network are used to describe the scene in an abstract manner that neglect irrelevant factors for the task at hand. The Inception network has been trained to classify a wide variety of images and hence should lend itself favorably to generalization.

Deep RL in high dimensional feature spaces in general takes millions of training steps to learn a meaningful policy. The sparsity of the obtained rewards strongly affects the rate of learning. Hierarchical RL, via the detection of *subgoals,* can reduce this sparsity. [1] detects subgoals via a form of clustering and selects a subset of the features to further reduce over fitting by ranking them as follows:

$$z_i^g = \alpha\left|\mu_i^g - \mu_i^{\neg g}\right| - \left(\sigma_i^g + \sigma_i^{\neg g}\right)$$

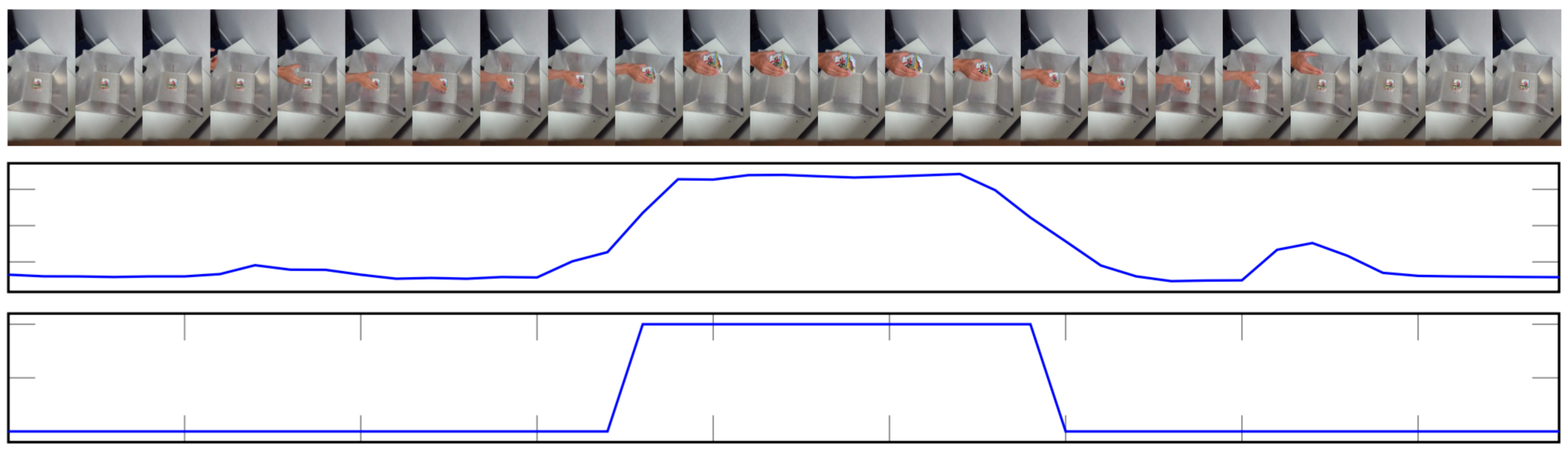
And retaining the top *n* features according to how they are ranked. The final reward function takes the following form, awarding higher rewards to latter subgoals to encourage the agent to explore them:

$$R_i(s_{it}) = \sum_{g=2}^{m}\left(1+\frac{\left(s_{it}-u_{ig}\right)^2}{2\sigma_{ig}^2}\right)^{-1}\cdot 2^{g-1}$$

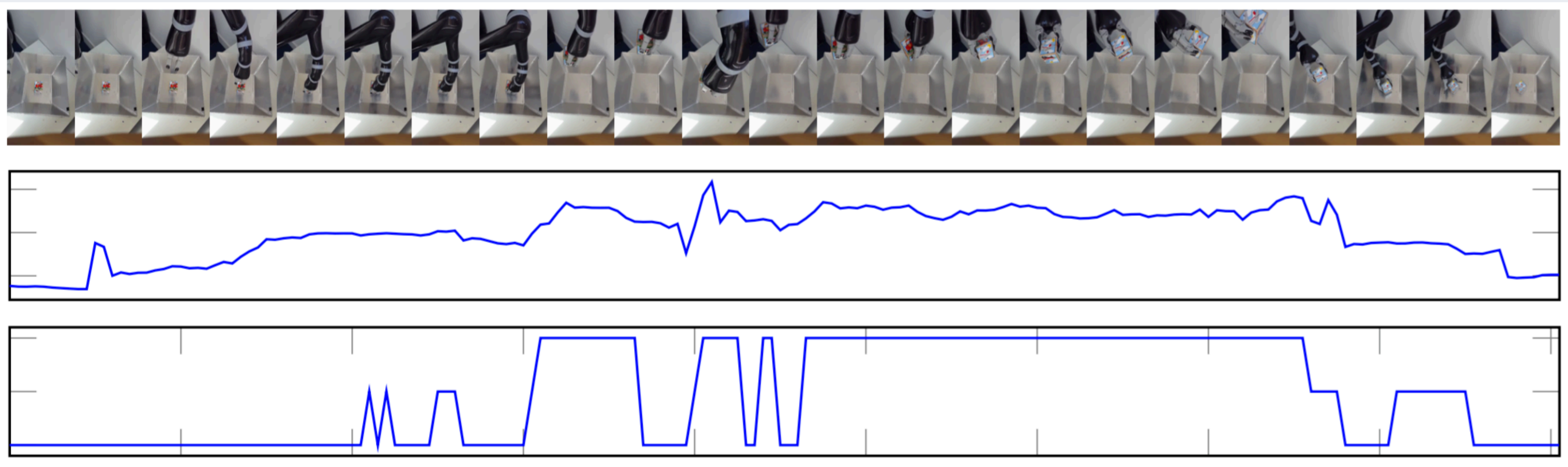
Results

It was decided to test the algorithm on the robotic grasping task of picking up a cube. The demonstration provided to the agent was taken by a human 'expert'. See the video below:

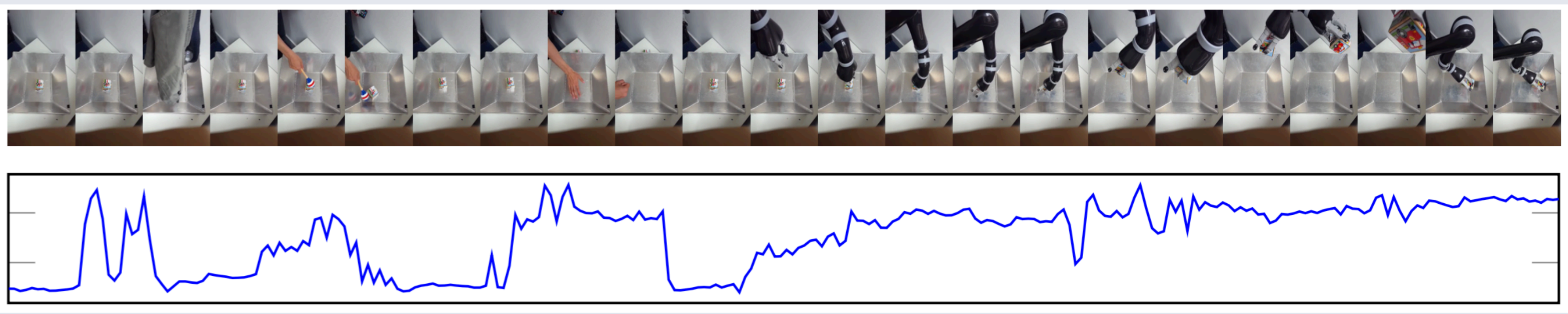
The first devised test was simple, keeping the agent embodiment the same as that of the expert. The results were promising[1]. The highest rewards were awarded when the cube was picked up, and intermediately sized rewards for when the agent approached the cube.



The algorithm was then run in the scenario where both the training and expert agents had different forms (human and robot hand). The learnt reward function was able to generalize well despite the changes in physical appearance.



The final test trajectory attempted to find the caveats associated with this particular IRL approach. The results are shown below and demonstrate that there are some scenarios that this technique cannot handle properly, such as occlusion of the rattle[2].



Conclusions and Further Work

By leveraging the expressivity of Google's pre-trained Inception network, this project was able to create an algorithm that can learn a reward function from limited demonstrations which in turn can be used to train RL.

This techniques is able to generalize well to different embodiments of the learning agent. Unfortunately, it is unclear that the agent is rewarded appropriately for the relevant factors of a given task. Learning from multiple demonstrations partially aided in this respect.

This project aims to further investigate task-specific feature representation that generalize well across domains, train RL in a virtual simulator, and port the learnt policy onto the physical robot arm via domain transfer learning.

Acknowledgements

The author would like to thank Dusyant Rao, Markus Wulfmeier and Ingmar Posner for their continuous support throughout this project.

References

[1] Sermanet, Pierre, Kelvin Xu, and Sergey Levine. "Unsupervised perceptual rewards for imitation learning. (2016).

[1] – All the test video plots show the video (top), the rewards (middle), the estimated subgoal (bottom).

[2] – For this scenario IRL was given a demonstration with a rattle.