Inverse Reinforcement Learning for Robotic Arm Control

Edward Wagstaff, Dushyant Rao, Markus Wulfmeier, Ingmar Posner

Overview

We wish to derive a control policy for a robotic arm to perform a task (e.g. grasping an object) by learning from human demonstrations. We chose to apply a promising recently published method due to Sermanet, Xu and Levine, and evaluate its effectiveness. For these purposes we chose the simple task of pushing an object off a surface, rather than starting with the more complex task of grasping.

For the purposes of this project we worked in simulation, since working with a real arm is substantially more time consuming. We identified a number of issues with the technique.

Learning a Reward Function

The method we chose to apply generates a reward function for use in reinforcement learning. A set of demonstration videos are provided, and from these the method derives a function mapping from images to rewards. The method uses the features obtained from passing images of the scene through a deep network, as it is hoped that these features will offer good generalisation performance and capture salient aspects of the scene.

Identifying Subtasks

We identify subtasks for each demonstration separately. The number of subtasks must be fixed in advance. Given a split into subtasks, let s_g^2 be the sum of the variance of all features within subtask g. We choose the split which minimises the total variance, i.e.$\text{å}_g s_g^2$. This minimisation may be performed by exhaustively evaluating this quantity for all possible splits, or a more efficient approximate method may be used.

Scoring by Similarity to a Subtask

Once subtasks have been identified, we wish to define a reward function which highly rewards states which are similar to a demonstrated subtask. We first identify a small set of features which discriminate strongly between subtasks (we fix the size of this set at 32). We then define a reward function which gives high reward when the activations of these features are similar to those in the demo.

To identify discriminative features, we score each feature by the following metric:

$$a\left|m_i^+ - m_i^-\right| - s_i^+ - s_i^-$$

Where μ+ and σ+ are the mean and standard deviation of the feature in images corresponding to the subtask, and μ- and σ- are the statistics for all other images. α=5 is chosen empricially.

The final reward function is the following sum over the 32 highest-scoring features:

$$\frac{1}{n}\sum_{i=1}^{32}\left(1+\frac{(s_i - m_i^+)^2}{2s_i^+}\right)$$

Issues Identified

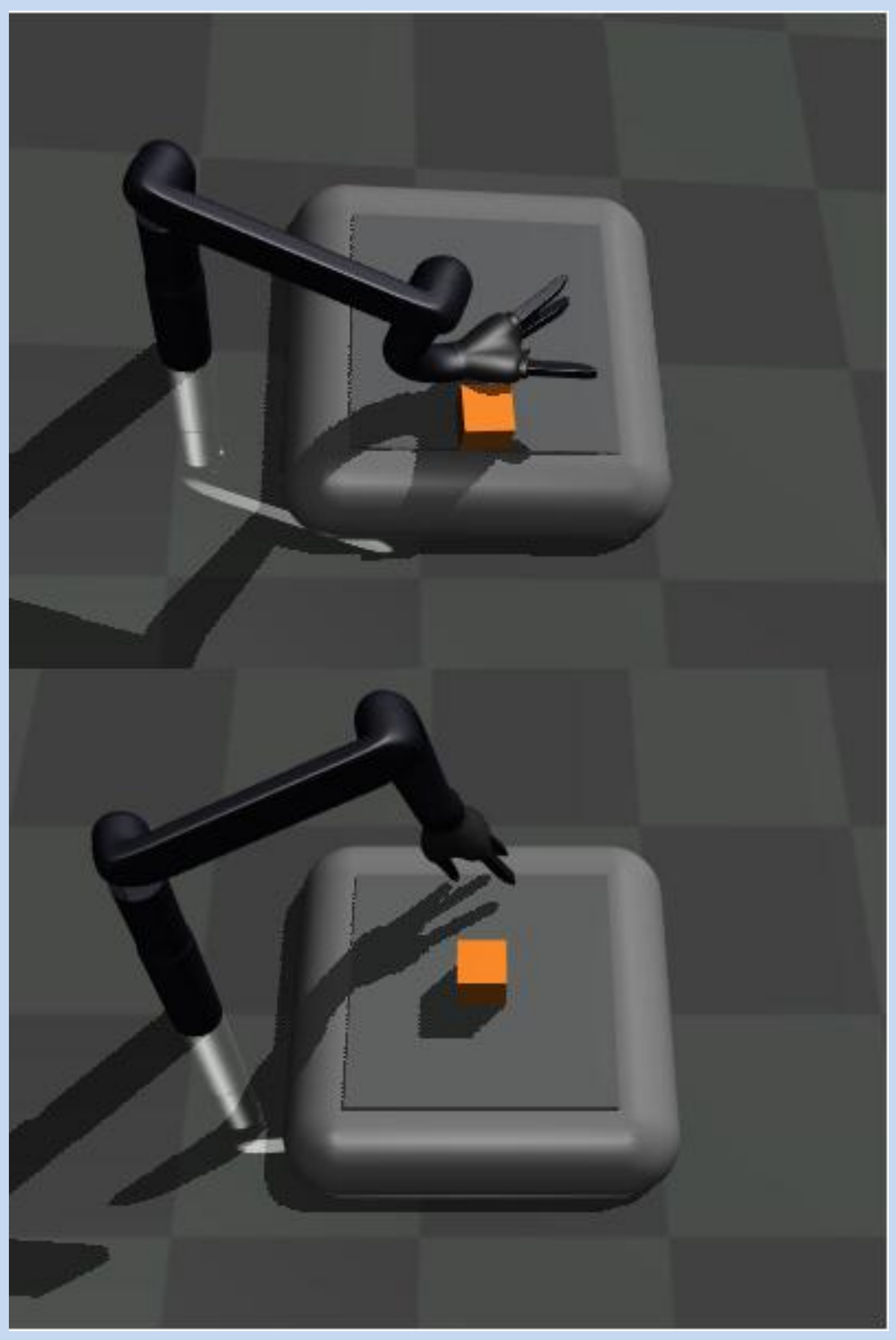
Lack of Specificity

Using reward functions generated by this method, we noted that it was possible to achieve high reward without completing the task as demonstrated. For example, swinging the arm in the same direction as the demonstrations resulted in high reward whether or not the object was actually moved by the arm. This indicates that the reward function is not sufficiently specific, and is either not incorporating features necessary for finer discrimination, or is not penalising divergence from those features sufficiently harshly.

We attempted to address this issue by incorporating “negative demonstrations”. This involved providing a new demonstration which was erroneously highly rewarded, and incorporating it into the statistics μ- and σ- at the feature scoring stage. We found that this modification to the method did not improve peformance.

Inconsistent subtask identification

When incorporating multiple demonstrations, we found that the split into subtasks was not always consistent across demonstrations. The figure to the right illustrates this – the images are from two separate demonstrations, and both have been classified as the start of the third subtask. We cannot simultaneously be similar to both images, so such a situation causes the method to fail. Due to time constraints we were not able to look into mitigating this. Considering all demonstrations at once when splitting into subtasks, and minimising variance over all splits across all demonstrations should improve this issue, but would come at a cost in computational complexity.



References

Sermanet, P., Xu, K., & Levine, S. (2016). Unsupervised perceptual rewards for imitation learning. arXiv preprint