Unsupervised Perceptual Rewards for Imitation Learning

Pierre Sermanet, Kelvin Xu, and Sergey Levine
ICLR 2017 workshop track
Unsupervised Perceptual Rewards for Imitation Learning

Pierre Sermanet, Kelvin Xu* & Sergey Levine
Google Brain
{sermanet,kelvinxx,slevine}@google.com
ICLR committee final decision

ICLR 2017 pcs
6 Feb 2017    ICLR 2017 conference paper521 acceptance    readers: everyone

Comment: Quality, Clarity:

The work is well motivated and clearly written -- no issues there.

Originality, Significance:

The idea is simple and well motivated, i.e., the learning of reward functions based on feature selection from identified subtasks in videos.

pros:
- the problem is difficult and relevant; good solutions would have impact

cons:
- the benefit with respect to other baselines for various choices, although the latest version does contain updated baselines
- the influence of the initial controller on the results
- the work may gain better appreciation at a robotics conference
Main Contributions

• A method for perceptual reward learning from only a few demonstrations of real-world tasks

• Unsupervised discovered of intermediate steps of reward function
Main Contributions

• **The first vision-based reward learning method** that can learn a complex robotic manipulation task from a few human demonstrations in **real world experiments**.

• Demonstration that the learned visual representations inside a pre-trained deep model are general enough to be directly used to represent goals and sub-goals for manipulation skills in new scenes **without retraining**.
Prior related work

• Tried to learn image-based reward functions, mostly using a raw target image and measuring distance to the image as reward function

• Required many more training examples
Demonstrator
(human or robot)
Demonstrator
(human or robot)

Few demonstrations
Demonstrator (human or robot)

Few demonstrations

Unsupervised discovery of intermediate steps
Demonstrator (human or robot)

Few demonstrations

Unsupervised discovery of intermediate steps

Feature selection maximizing step discrimination across all videos
Real-time perceptual reward for multiple intermediate steps
Real-time perceptual reward for multiple intermediate steps

Learning agent with Reinforcement Learning
Demonstrator
(human or robot)

Few demonstrations

Unsupervised discovery of intermediate steps

Feature selection maximizing step discrimination across all videos

Real-time perceptual reward for multiple intermediate steps

Learning agent
with Reinforcement Learning
general high-level features

pretrained deep model (e.g. Inception)
Deep learning in a nutshell

Source: http://www.biomedcentral.com/content/figures/1472-6750-7-53-2-l.jpg
Learning how to represent visual input

Basic building blocks of primate visual system

Lowest layer of a neural network trained on natural images

Higher level representations

Features learned by 4 deep nets in 4 different categories

Convolutional feature maps

The architecture of LeNet5

Source: Coursera Neural Networks course by Geoffrey Hinton, lecture 5c
Google’s Inception network

- Very deep
- Learns many layers of representation

Pretrained on the ImageNet dataset

• 1.2 million training images

Pretrained on the ImageNet dataset

- Learns high level features
- Generalizes to new situations
Pouring training set

11 examples
Use the features from the network

- Segment the video to minimize the variance of each feature within each segment
Use the features from the network

- Segment the video to minimize the variance of each feature within each segment

<table>
<thead>
<tr>
<th>dataset (training)</th>
<th>method</th>
<th>2 steps</th>
<th>3 steps</th>
<th>average</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>step 1</td>
<td>step 2</td>
<td>average</td>
</tr>
<tr>
<td>door</td>
<td>ordered random steps</td>
<td>59.4%</td>
<td>45.6%</td>
<td>52.5%</td>
</tr>
<tr>
<td></td>
<td>unsupervised steps</td>
<td>84.0%</td>
<td>68.1%</td>
<td>76.1%</td>
</tr>
<tr>
<td>pouring</td>
<td>ordered random steps</td>
<td>65.2%</td>
<td>66.6%</td>
<td>65.9%</td>
</tr>
<tr>
<td></td>
<td>unsupervised steps</td>
<td>92.3%</td>
<td>90.5%</td>
<td>91.6%</td>
</tr>
</tbody>
</table>
Use the features from the network

- Try splitting into different number of segments
Use the features from the network

• Select the most discriminative features

• Features, times steps assumed independent

• Reward function is made up of (log of) Gaussians fit to each feature
Use the features from the network

- Select the most discriminative features
- Features, times steps assumed independent
- Reward function is made up of (log of) Gaussians fit to each feature
Combine intermediate rewards into a single reward function

- Won’t know intermediate step boundaries at run time
- Partially reward intermediate steps but save most of the reward for the last step
Figure 5: **Rewards from human demonstration only.** Here we show the rewards produced when trained on humans only (see Fig. 11). In 5a, we show the reward on a human test video. In 5b, we show what the reward produces when the human hands misses opening the door. In 5c, we show the reward successfully saturates when the robot opens the door even though it has not seen a robot arm before. Similarly in 5d and 5e we show it still works with some amount of variation of the door which was not seen during training (white door and black handle, blue door, rotations of the door).
ICLR committee final decision

ICLR 2017 pcs
6 Feb 2017  ICLR 2017 conference paper521 acceptance  readers: everyone

Comment: Quality, Clarity:

The work is well motivated and clearly written -- no issues there.

Originality, Significance:

The idea is simple and well motivated, i.e., the learning of reward functions based on feature selection from identified subtasks in videos.

pros:
- the problem is difficult and relevant: good solutions would have impact

cons:
- the benefit with respect to other baselines for various choices, although the latest version does contain updated baselines
- the influence of the initial controller on the results
- the work may gain better appreciation at a robotics conference
2) I understand the desire to combine the extracted reward function with a simple RL method but believe the used simple controller could potentially introduce a significant bias in the experiments since it requires initialization from an expert trajectory. As a direct consequence of this initialization the RL procedure is already started close to a good solution and the extracted reward function is potentially only queried in a small region around what was observed in the initial set of images (perhaps with the exception of the human demonstrations). Without an additional experiment it is thus unclear how well the presented approach will work in combination with other RL methods for training the controller.
Initialize with kinesthetic demonstrations such as different position or orientation of the door. Following the experimental protocol in prior work (Chebotar et al., 2016), we adapt an imperfect kinesthetic demonstration which we ensure succeeds at least occasionally (about 10% of the time). These demonstrations consist only of robot poses, and do not include images.
Initialize with kinesthetic demonstrations

Demonstrator (human or robot)

Few demonstrations

Unsupervised discovery of intermediate steps

Feature selection maximizing step discrimination across all videos

Real-time perceptual reward for multiple intermediate steps

Learning agent with Reinforcement Learning
Initialize with kinesthetic demonstrations

Path Integral Guided Policy Search

Yevgen Chebotar, Mrinal Kalakrishnan, Ali Yahya, Adrian Li, Stefan Schaal, Sergey Levine

It is important to note that, especially for real robot tasks, it is often not safe or efficient to start with an uninitialized or randomly initialized global policy, particularly when using very general representations like neural networks. Therefore, in our work we initialize the global policies by performing several iterations of standard GPS with local policy sampling using PI² on a fixed set of task instances. In this case, the cost-to-go $S_{i,t}$ in PI² is augmented with a KL-divergence penalty against the global policy as described in [6] (Appendix A), but the optimization is performed using the KL-divergence constraint against the previous local policy. We also initialize the local policies with kinesthetic teaching, to provide the algorithm with the overall structure of the task at the start of training. After initialization, the global

https://www.youtube.com/watch?v=ncp1kY5JV90
Learn a policy with PI$^2$ algorithm

- Relatively simple policy
- Iteratively updated by PI$^2$ algorithm using the reward signal from the video signal
- “Trust region” variant of PI$^2$ ensures small changes to the policy

We employ a relatively simple linear-Gaussian parameterization of the policy, which corresponds to a sequence of open-loop torque commands with fixed linear feedback to correct for perturbations, as in the work of Chebotar et al. (2016). This policy has the form $\pi(u_t|x_t) = \mathcal{N}(K_t x_t + k_t, \Sigma_t)$, where $K_t$ is a fixed stabilizing feedback matrix, and $k_t$ is a learned control. In this case, the state $x_t$ corresponds to the joint angles and angular velocities of a robot, and $u_t$ corresponds to the joint torques.
Real world results
Real world results

![Graph showing success rate for 10 rollouts against iteration number for different methods: Baseline PI-Squared, Our method (2 sub-goals, robot demonstration), Our method (5 sub-goals, robot demonstration), Our method (4 sub-goals, human demonstrations only, slight appearance variations).]
Dependencies: further reading

Path Integral Guided Policy Search
Yevgen Chebotar\textsuperscript{1}  Mrinal Kalakrishnan\textsuperscript{2}  Ali Yahya\textsuperscript{2}  Adrian Li\textsuperscript{2}  Stefan Schaal\textsuperscript{1}  Sergey Levine\textsuperscript{3}


Submitted 1/10; Revised 7/10; Published 11/10

A Generalized Path Integral Control Approach to Reinforcement Learning
Evangelos A.Theodorou
Jonas Buchli
Stefan Schaal\textsuperscript{*}

Department of Computer Science
University of Southern California
Los Angeles, CA 90089-2905, USA

ETHEODOR@USC.EDU
JONAS@BUCHLI.ORG
SSCHAAL@USC.EDU

Dependencies: further reading

Rethinking the Inception Architecture for Computer Vision

Christian Szegedy  
Google Inc.  
szegedy@google.com

Vincent Vanhoucke  
vanhoucke@google.com

Sergey Ioffe  
sioffe@google.com

Jonathon Shlens  
shlens@google.com

Zbigniew Wojna  
University College London  
zbigniewwojna@gmail.com

https://arxiv.org/abs/1512.00567

Maximum Entropy Inverse Reinforcement Learning

Brian D. Ziebart, Andrew Maas, J. Andrew Bagnell, and Anind K. Dey  
School of Computer Science  
Carnegie Mellon University  
Pittsburgh, PA 15213  
bziebart@es.cmu.edu, amaas@andrew.cmu.edu, dbagnell@ri.cmu.edu, anind@cs.cmu.edu