Uncertainty Aware Learning from Demonstrations in Multiple Contexts using Bayesian Neural Networks

Sanjay Thakur, Herke Van Hoof, Juan Camilo Gamboa Higuera, Doina Precup, David Meger

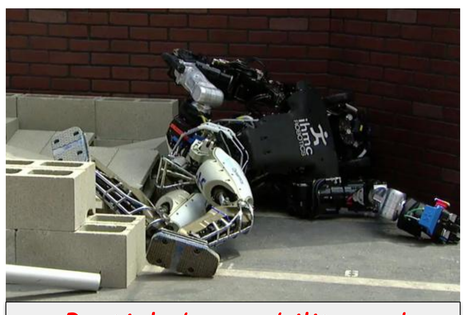
OBJECTIVES

Goal: Train controllers that learn from demonstrations (LfD) to have *principled, and quantified sense of uncertainty* in its decision-making on complex, high-dimensional and *partially observable environments*.

Use case: Base decision-making of an active learner with such a sense of uncertainty *to yield a sample efficient learner* from demonstrations.

MOTIVATION

Data is expensive to obtain



Partial observability and unseen situations cause failure

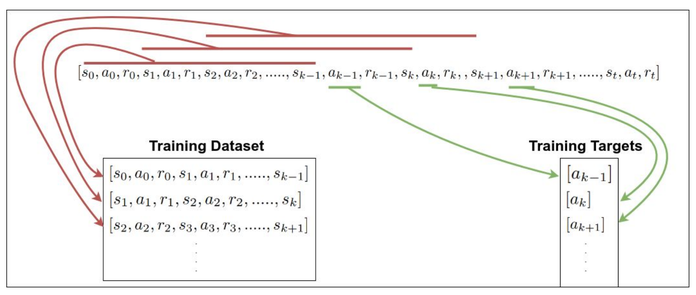
OUR METHODOLOGY

We use a combination of the following techniques:

- **Moving Temporal Windows**
- **Bayes-by-Backprop [1]**
- **Adaptive Threshold**

Moving Temporal Windows

We use temporal windows of the most recent ***k*** steps as inputs.



Allows controller to *identify unobservable changes* due to

- dynamics function,
- reward function

We call *different situations due to such unobservable changes as different contexts*.

Bayes-by-Backprop

- Our learner has to represent uncertainty in its action that it believes the demonstrator would have taken. We do this by *finding a distribution over the weights of a neural network.*
- It uses *variational inference, gaussian reparameterization trick, and Monte-Carlo* sampling to approximate this distribution.

Training:

Variational Parameters: $\vec{\theta} = [\vec{\mu}^T, \vec{\rho}^T]^T$, where $\vec{\Sigma}_{ii} = \log(1+\exp(\vec{\rho}_i))$

Dataset: $\mathcal{D} = \{(x_t, a_t)\}$

Loss function:

$$\mathcal{L}(D, \vec{\theta}) \approx \frac{1}{M} \sum_{i=1}^{M} \log q(\vec{w}^{(i)} \mid \vec{\theta}) - \log \left(P(\vec{w}^{(i)}) P(D \mid \vec{w}^{(i)}) \right)$$

Monte Carlo Samples; Variational Posterior Distribution; Prior Parameter Distribution; Likelihood

$$q(\vec{w}|\vec{\theta}) \sim \mathcal{N}(\vec{\mu}, \vec{\Sigma})$$

Inference:

Optimized Variational Parameters:

$$\vec{\theta}^* = \arg\min_{\vec{\theta}} \mathcal{L}(\mathcal{D}, \vec{\theta})$$

$$\vec{a}_t, \vec{\sigma}_t = \hat{\pi}_{\vec{\theta}^*}(\vec{x}_t), \text{ with } \vec{a}_t = \mathbb{E}\left[\hat{\vec{y}}|\vec{x}_t\right], \vec{\sigma}_t = \sqrt{\mathrm{Var}\left[\hat{\vec{y}}|\vec{x}_t\right]}$$

Controller action; Predictive uncertainty or Confidence of the controller

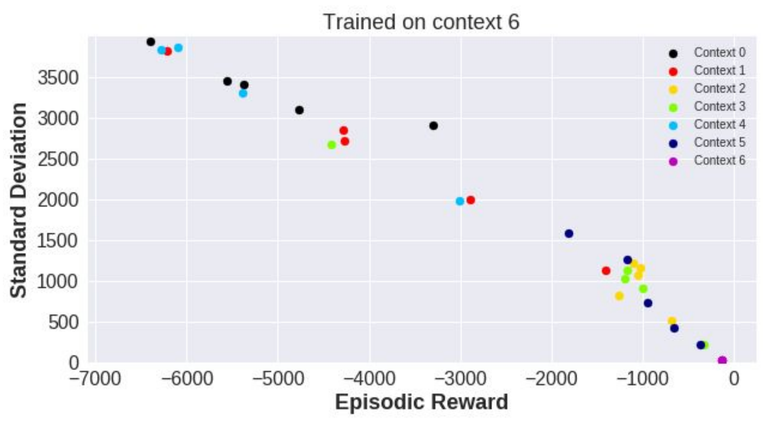
Adaptive Threshold

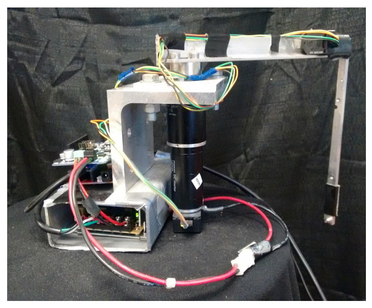
- Value of this threshold is set as the average predictive standard deviation obtained on all seen contexts by the controller. This is to identify familiar contexts from non-familiar contexts.
- Uncertainty is scaled by a factor ***c*** and averaged ***m*** time-steps before comparing with the threshold.

EXPERIMENTS and RESULTS

Real Robotic Pendulum Swing up

We generate *different contexts by changing the mass of the pole*.





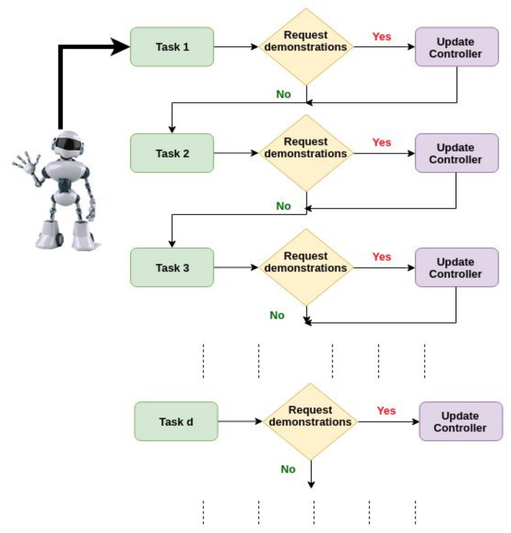
Our mechanism captures the degree of task success through a easy to obtain quantity of predictive standard deviation.

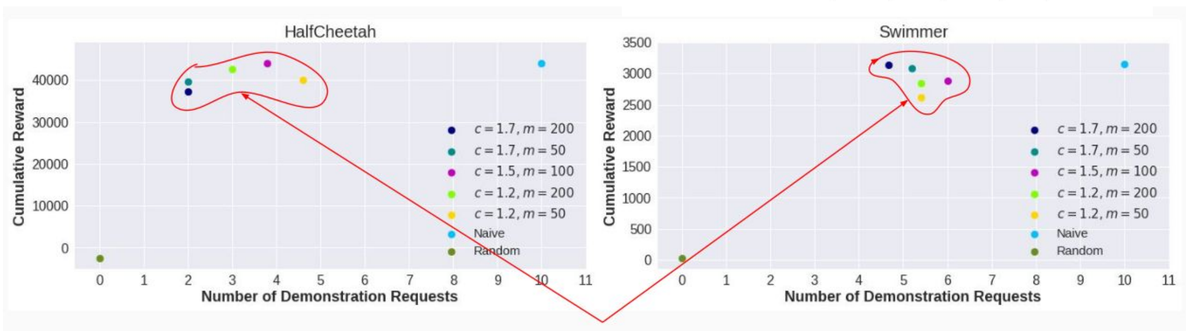
MuJoCo

We generate *different contexts* in HalfCheetah and Swimmer tasks *by changing the masses and lengths of* various body parts.

Task set up:

- Controller faces contexts one after another.
- At each context, depending on the predictive standard deviation, decision can be made whether or not to seek more contet-specific demonstration.





- **Performance close to a naive learner that seeks demonstrations on every context, can be obtained by lesser number of such requests.**
- **Lower c and m leads to seeking more context-specific demonstrations or more conservative behavior without much gain in reward.**

References:

1. Blundell, C., Cornebise, J., Kavukcuoglu, K. and Wierstra, D., 2015. Weight uncertainty in neural networks. *arXiv preprint arXiv:1505.05424*.

Links to our paper, code, full presentation, videos and the blog post are here - https://bit.ly/2O2EaKO

Mila

MRL mobile robotics laboratory @ McGill University

RLLab McGill University