Reinforcement learning (RL) is a key technique for learning through interaction with the environment.

**Problem Definition:**

RL problems are formalized as **Markov Decision Processes (MDPs):** \( \langle S, A, P, R, \gamma \rangle \)

- **S** : State Space
- **A** : Action Space
- **P** : Transition Probability
- **R** : Reward Function
- **\( \gamma \)** : Discount Factor

**Goal**

Learn optimal policy by maximizing

\[
Q(s, a) = \mathbb{E} \left[ \sum_{t=0}^{\infty} \gamma^t R_t \right]
\]
Motivation

**Problem**

Reinforcement learners are slow to learn

**Possible Solution**

Reuse knowledge from other sources

- Learning from Demonstration
- Transfer Learning

**Impressive Results**
Transfer Learning

Pool of source tasks from same domain

New target task

Questions to answer:

1. How to transfer?  
   lots of approaches

2. What to transfer?  
   lots of approaches

3. When to transfer?  
   Less progress has been achieved  
   Needs a task similarity measure

Bou Ammar, Eaton, et al.
RBDist: Similarity Measure Between MDPs
RBDist: Similarity Measure Between MDPs

Our measure is based on **Restricted Boltzmann Machines (RBMs):**

- Set of visible units $\mathcal{V} = \{v^{(1)}, \ldots, v^{(n_v)}\}$
- Set of hidden units $\mathcal{H} = \{h^{(1)}, \ldots, h^{(n_h)}\}$

**RBM Energy Function**

$$E(v, h) = - \sum_{i,j} v^{(i)}h^{(j)}w^{(i,j)} - \sum_i v^{(i)}a^{(i)} - \sum_j h^{(j)}b^{(j)}$$

**Probability distribution**

$$p(v, h) \propto \exp(-E(v, h))$$

Weights are trained using contrastive divergence

Bou Ammar, Eaton, et al.
**RBDist: Similarity Measure Between MDPs**

**Step 1:** Train an RBM to approximate the source task’s dynamics

The RBM learns a generative model that captures the source dynamics.

**Key Idea:** If the dynamics of a source and target domain are similar, the RBM trained on the source task should be able to **reconstruct** trajectories from the target task.

Bou Ammar, Eaton, et al.
RBDist: Similarity Measure Between MDPs

Step 2: Reconstruct target task trajectories by sampling the trained RBM

- **Trajectories from target task**
- **Visible Layer**
- **Hidden Layer**
- **Reconstruction of target trajectories based on source task’s dynamics**

Step 3: Measure reconstruction error of sampled target trajectories as RBDist

\[
\text{RBDist} = \frac{1}{n} \sum_{k=1}^{n} e_k
\]

\[
e_k = L_2 \left( \langle s_2^{(k)}, a_2^{(k)}, s'_2^{(k)} \rangle_0, \langle s_2^{(k)}, a_2^{(k)}, s'_2^{(k)} \rangle_1 \right)
\]

- original tuple
- reconstructed tuple

Bou Ammar, Eaton, et al.
Experiments & Results
Dynamical Systems & Benchmarks

Inverted Pendulum
- Swing and balance pole upright by applying torques

Cart Pole
- Balance pole upright by applying linear forces

Mountain Car
- Control car to reach goal by oscillating around the valley

Bou Ammar, Eaton, et al.
Results: Dynamical Phases

RBDist can automatically cluster dynamical phases

Bou Ammar, Eaton, et al.
Results: Transfer Performance

RBDist correlates with transfer performance
Thank you!

Please send correspondence to:

Haitham Bou Ammar  
haithamb@seas.upenn.edu

Eric Eaton  
eeaton@seas.upenn.edu

This work was supported in part by ONR N00014-11-1-0139, AFOSR FA8750-14-1-0069, and NSF IIS-1149917.