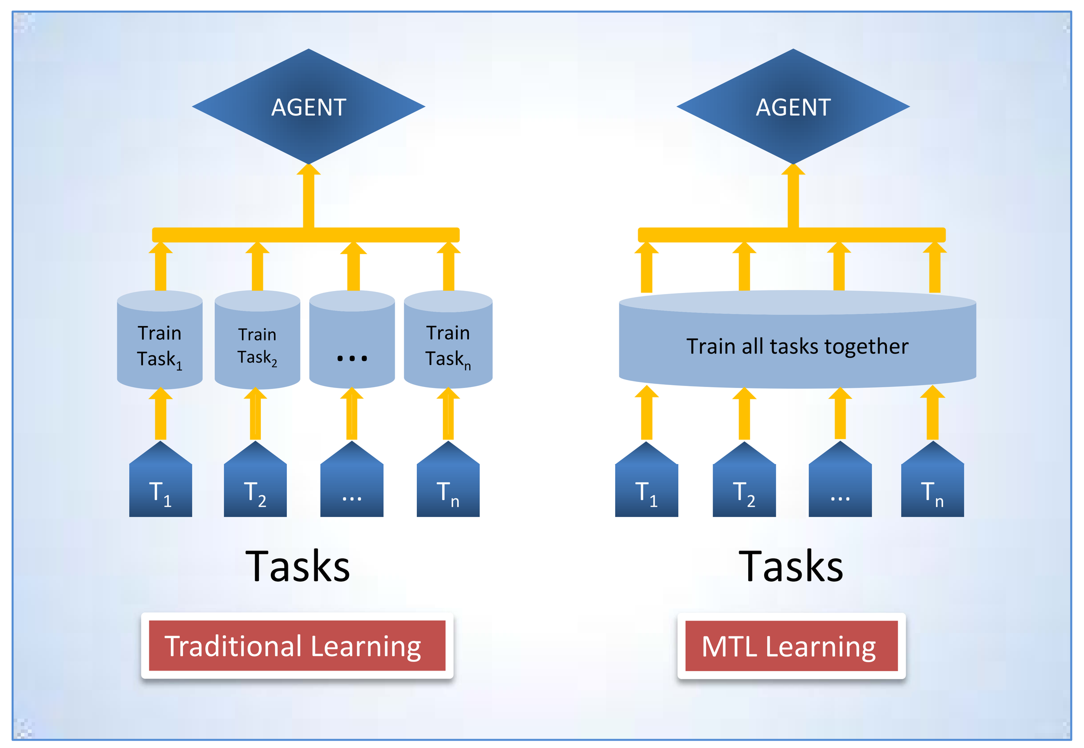
Online Multi-Task Gradient Temporal-Difference Learning

Vishnu Purushothaman Sreenivasan, Haitham Bou Ammar, and Eric Eaton
University of Pennsylvania, Computer and Information Science Department

Motivation

- Reinforcement learning is widely used for design of autonomous systems, but RL agents often require extensive experience to achieve optimal behavior.
- Policies for multiple tasks are often required to be learnt by the agent in order to achieve the overall objective.
- In such scenarios, learning multiple tasks models jointly (called multi-task learning or MTL) produces improved performance but at a large computational cost.

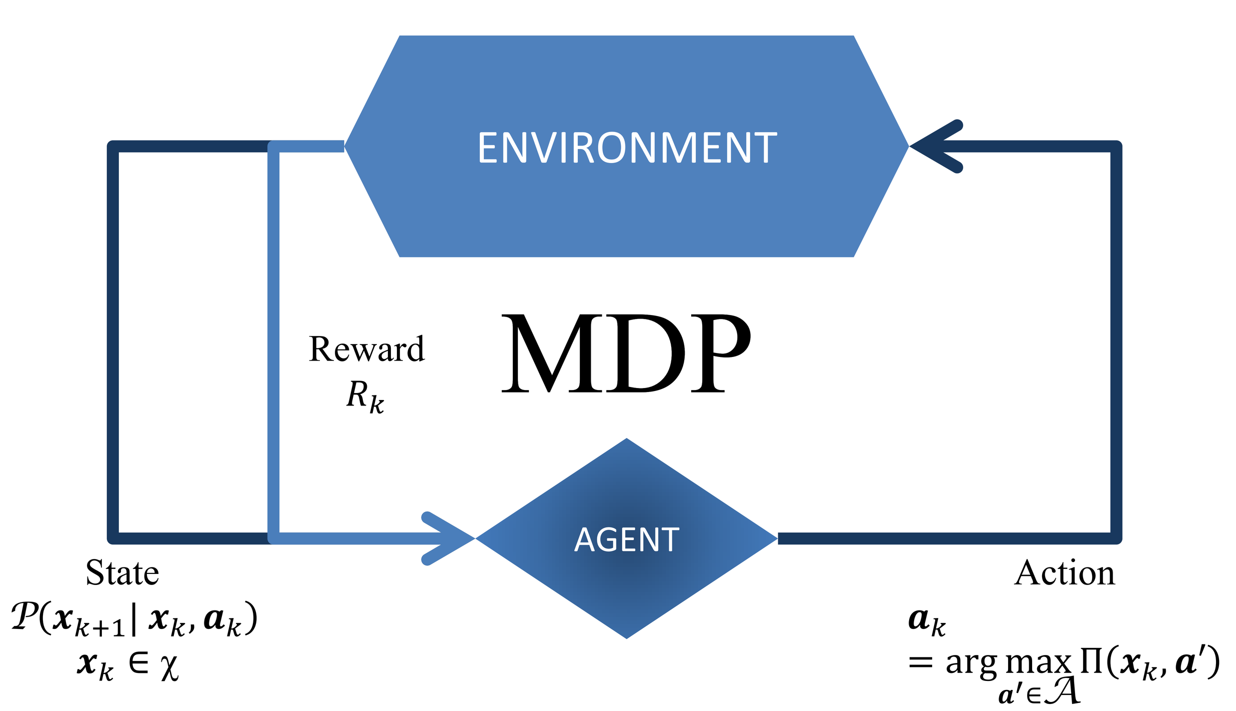


Goal

To design a MTL formulation for RL that

1. reduces the required overall interaction time of the agent with the environment,
2. allows the agent to rapidly learn new tasks by building on prior knowledge.

Background: Reinforcement Learning



Find policy $\Pi^*$by taking greedy actions in value function V^*, to maximize the expected return $= \lim_{n\to\infty} \sum_{k=0}^{n} \gamma^k R_k$

Background: Gradient Temporal Difference Learning

- Value function is approximated by a linear combination of a set of basis functions $\boldsymbol{\Phi}(\boldsymbol{x})$ representing the state space.

$$V = \boldsymbol{\theta}^T \boldsymbol{\Phi}(\boldsymbol{x})$$

V – Value function
$\boldsymbol{\Theta}$ – Parameter vector $\Theta \in \mathrm{R}^n$
$\boldsymbol{\Phi}$ – State basis function $\Phi : \chi \to \mathrm{R}^n$

- Value function estimated from the set $\{(\boldsymbol{\Phi}(\boldsymbol{x}_k), \boldsymbol{\Phi}(\boldsymbol{x}_k'), R_k)\}_{k=1,2,\ldots}$ where,

$\boldsymbol{x}_k$– Current state, $\boldsymbol{x}_k{}'$ – Successor state
$\boldsymbol{\Phi} = \boldsymbol{\Phi}(\boldsymbol{x}_k)$, $\boldsymbol{\Phi}' = \boldsymbol{\Phi}(\boldsymbol{x}_k{}')$

- GTD minimizes the L2 norm of the temporal difference error:

$$J(\boldsymbol{\theta}) = E[\delta\boldsymbol{\phi}]^T E[\delta\boldsymbol{\phi}]$$

by following the gradient of the objective function:

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = E[\boldsymbol{\phi}(\boldsymbol{\phi} - \boldsymbol{\gamma}\boldsymbol{\phi}')^T]^T E[\delta\boldsymbol{\phi}]$$

$$\delta = R_k + \gamma \boldsymbol{\theta}^T \boldsymbol{\phi}' - \boldsymbol{\theta}^T \boldsymbol{\phi}$$

Problem Definition

- Agent learns a series of RL tasks $\mathcal{Z}^{(1)}, \ldots, \mathcal{Z}^{(T max)}$, each of which is an MDP.

$$\mathcal{Z}^{(t)} = \langle\, \chi^{(t)}, \mathcal{A}^{(t)}, \mathcal{P}^{(t)}, R^{(t)}, \gamma^{(t)} \,\rangle$$

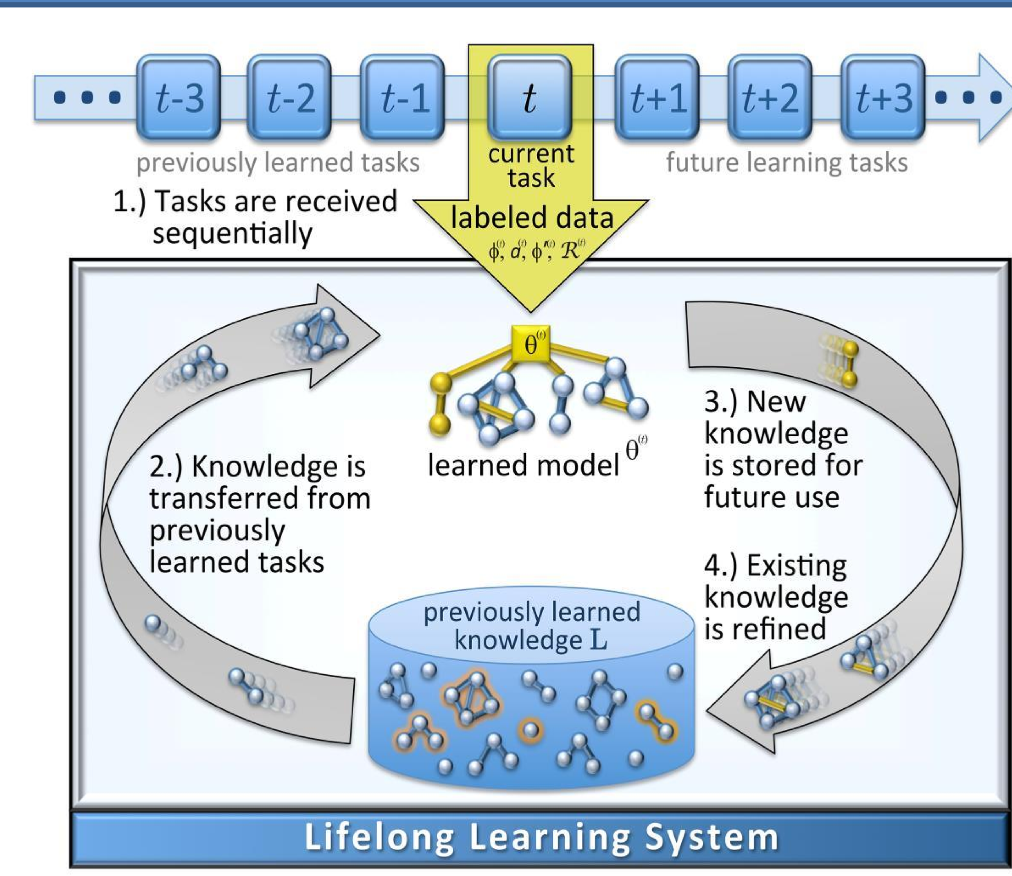
  - Tasks may be revisited any number of times and in any order.
  - Agent does not know the total number of tasks a priori.
- The goal is to learn an optimal set of value functions

$$V^* = \left\{ V^*_{\{\boldsymbol{\theta}^{(1)}\}}, \ldots, V^*_{\{\boldsymbol{\theta}^{(Tmax)}\}} \right\}$$

with corresponding parameter vectors $\boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(Tmax)}$.

- We consider the model-based RL setting, which can be readily extended to a model-free scenario.

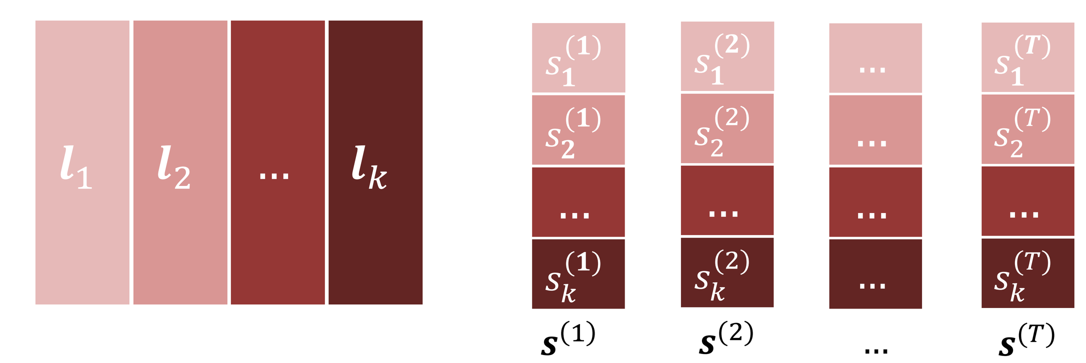
Online Multi-Task Learning Process



Approach

Maintain a library of k latent components $\boldsymbol{L} \in \mathrm{R}^{\{d \times k\}}$ that is shared among all the tasks and forms a basis for representing the parameter vector of the task models.

$$\boldsymbol{\theta}^{(t)} = \boldsymbol{L}\, \boldsymbol{s}^{(t)}$$



Given T tasks, the MTL objective function is

$$e_T(\boldsymbol{L}) = \frac{1}{T} \sum_{t=1}^{T} \min_{\boldsymbol{s}^{(t)}} \left[J(\boldsymbol{\theta}^{(t)}) + \mu \left\| \boldsymbol{s}^{(t)} \right\|_1 \right] + \lambda \|\boldsymbol{L}\|_F^2$$

Eliminating Dependence on All Trajectories

- The above equation is not jointly convex in $\boldsymbol{L}$ and $\boldsymbol{s}^{(t)}$'s.
  - Approximating the loss function $J(\boldsymbol{\theta}^{(t)})$ with the second order Taylor expansion around the optimal single-task solution $\alpha^{(t)}$.
  - Computation of $\alpha^{(t)}$is performed using GTD.

$$e_T(\boldsymbol{L}) = \frac{1}{T} \sum_{t=1}^{T} \min_{\boldsymbol{s}^{(t)}} \left[\left\| \boldsymbol{\alpha}^{(t)} - \boldsymbol{L}\boldsymbol{s}^{(t)} \right\|_{\boldsymbol{\Gamma}^{(t)}}^2 + \mu \left\| \boldsymbol{s}^{(t)} \right\|_1 \right] + \lambda \|\boldsymbol{L}\|_F^2$$

$$\boldsymbol{\alpha}^{(t)} = \arg\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta}^{(t)}) \qquad \Gamma^{(t)} = \nabla_{\boldsymbol{\theta}^{(t)}, \boldsymbol{\theta}^{(t)}} J(\boldsymbol{\theta}^{(t)})$$

Eliminating the Reoptimization of Other Tasks

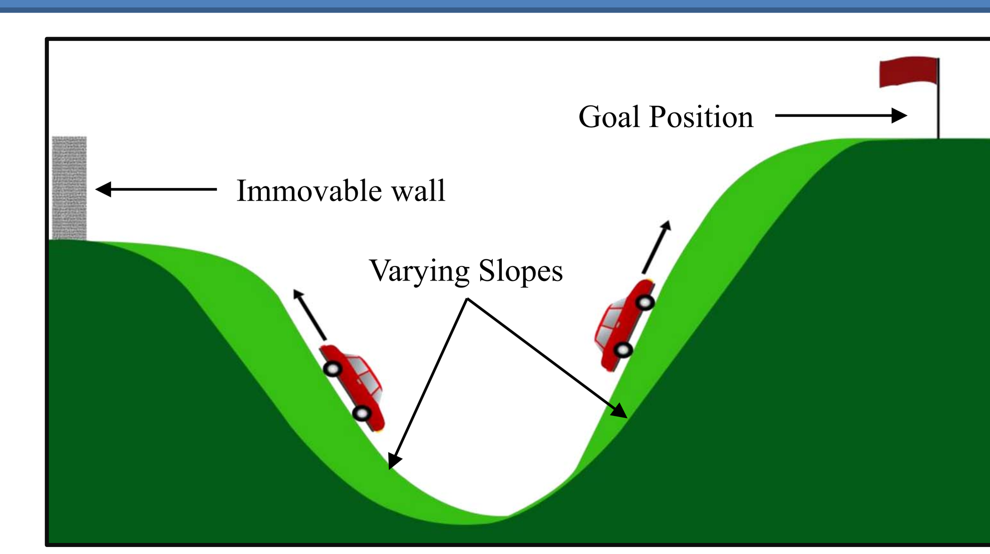
- Modify the MTL objective function by eliminating minimization over all $s^{(t)}$'s.
- Updating $s^{(t)}$'s only when training on task t.

$$s^{(t)} \leftarrow \arg\min_{\boldsymbol{s}^{(t)}} \left(\boldsymbol{L}_m, \boldsymbol{s}^{(t)}, \boldsymbol{\alpha}^{(t)}, \boldsymbol{\Gamma}^{(t)} \right)$$

$$\boldsymbol{L}_{m+1} \leftarrow \arg\min_{\boldsymbol{L}} \frac{1}{T} \sum_{t=1}^{T} l\left(\boldsymbol{L}, \boldsymbol{s}^{(t)}, \boldsymbol{\alpha}^{(t)}, \boldsymbol{\Gamma}^{(t)} \right) + \lambda \|\boldsymbol{L}\|_F^2$$

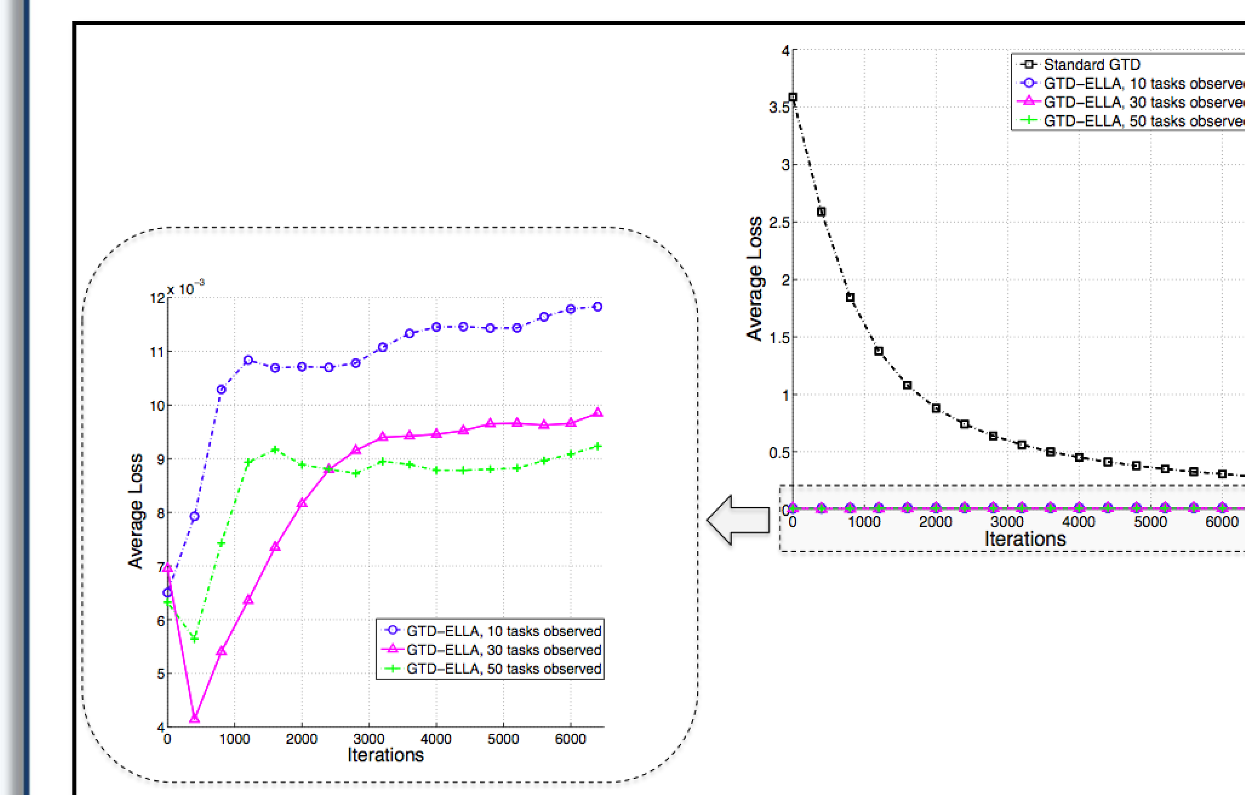
where $l(\boldsymbol{L}, \boldsymbol{s}^{(t)}, \boldsymbol{\alpha}^{(t)}, \boldsymbol{\Gamma}^{(t)}) = \mu\|\boldsymbol{s}\|_1 + \|\boldsymbol{\alpha} - \boldsymbol{L}\boldsymbol{s}\|^2$ and $\boldsymbol{L}_m$ corresponds to the value of the latent basis at the m^{th} iteration.

Mountain Car Tasks



Preliminary Results

- We evaluated GTD-ELLA on multiple tasks in the mountain car (MC) domain.
- State is given by position and velocity, represented by 6 radial basis functions linearly spaced across both the dimensions.
- Parameters:
  - Position is bounded between 1.2 and 0.6.
  - Velocity is bounded between -0.07 to 0.07.
  - Rewards of -1 in all states except goal state at which reward is 0.
- Generated 75 tasks by randomizing the valley slope which also changes the valley position.
- We trained GTD-ELLA on different number of task to learn L and evaluation was conducted on 25 unobserved MC tasks using either GTD-ELLA or standard GTD(0).
- The results indicate GTD-ELLA significantly improves RL performance when training on new tasks. Further, as the agent learns more tasks, its overall performance improves.



Future Work

- Extend the GTD-ELLA algorithm to a model-free RL setting.
- Support transfer between tasks with different feature spaces

Acknowledgements

This work was partially supported by ONR grant #N00014-11-1-0139 and AFOSR grant #FA8750-14-1-0069

Contact

Vishnu Purushothaman Sreenivasan
2nd Year, Robotics MSE, University of Pennsylavania
email : visp@seas.upenn.edu