Learning Curriculum Policies for Reinforcement Learning

Sanmit Narvekar and Peter Stone
Department of Computer Science
University of Texas at Austin
{sanmit, pstone} @cs.utexas.edu
Successes of Reinforcement Learning

Approaching or passing human level performance

BUT

Can take *millions* of episodes! People learn this **MUCH** faster
People are able to learn a lot of complex tasks very efficiently
Example: Quick Chess

- Quickly learn the fundamentals of chess
- 5 x 6 board
- Fewer pieces per type
- No castling
- No en-passant
Example: Quick Chess
Quick Chess is a curriculum designed for people
We want to do something similar automatically for autonomous agents
Curriculum learning is a complex problem that ties task creation, sequencing, and transfer learning.
Value Function Transfer

- Initialize Q function in target task using values learned in a source task

- Assumptions:
  - Tasks have overlapping state and action spaces
  - OR an inter-task mapping is provided
    - Existing related work on learning mappings

\[ Q_{\text{source}}(s,a) \]

Image credit: Taylor and Stone, JMLR 2009
Reward Shaping Transfer

• Reward function in target task augmented with a shaping reward \( f \):

\[
\begin{array}{c}
\text{New Reward} \\
\text{Old Reward} \\
\text{Shaping Reward}
\end{array}
\]

\[
r'(s, a, s') = r(s, a, s') + f(s, a, s')
\]

• Potential-based advice restricts \( f \) to be difference of potential functions:

\[
f(s, a, s') = \Phi(s', \pi(s')) - \Phi(s, a)
\]

• Use the value function of the source as the potential function:

\[
\Phi(s, a) = Q_{source}(s, a)
\]
The Problem: Autonomous Sequencing

• Existing work heuristic-based, such as examining performance on the target task, and using heuristics to select next task

• In this work, we use learning to do sequencing
Sequencing as an MDP

Curriculum Task

Task 1
- Environment
- RL Agent
- Action
- State Reward

Task 2
- Environment
- RL Agent
- Action
- State Reward

Task N
- Environment
- RL Agent
- Action
- State Reward

Curriculum State
Curriculum Reward

University of Texas at Austin  Sanmit Narvekar 11
Sequencing as an MDP

- **State space** $S^C$: All policies $\pi_i$ an agent can represent
- **Action space** $A^C$: Different tasks $M_j$ an agent can train on
- **Transition function** $p^C(s^C, a^C)$: Learning task $a^C$ transforms an agent’s policy $s^C$
- **Reward function** $r^C(s^C, a^C)$: Cost in time steps to learn task $a^C$ given policy $s^C$
Sequencing as an MDP

A policy $\pi^c: S^c \rightarrow A^c$ on this curriculum MDP (CMDP) specifies which task to train on given learning agent policy $\pi_i$.

Essentially training a teacher.

How to do learning over CMDP?

How does CMDP change when transfer method changes?
Learning in Curriculum MDPs

- Express raw CMDP state using the weights of base agent’s VF/policy
- Extract features so that similar policies (CMDP states) are “close” in feature space

Graphical representation:

- Extract Raw CMDP State Variables → Extract Features → Function Approximation and Learning

University of Texas at Austin  Sanmit Narvekar  14
Example: Discrete Representations

- CMDP states 1 and 2 encode very similar policies, and should be close in CMDP representation space
Example: Discrete Representations

- One approach: use tile coding
- Create a separate tiling on a state-by-state level
- When comparing CMDP states, the more similar the policies are in a primitive state, the more common tiles will be activated
- Each primitive state contributes equally towards the similarity of the CMDP state
Continuous CMDP Representations

• In continuous domains, weights are not local to a state

• Needs to be done separately for each domain
  • Neural networks
  • Tile coding
  • Etc...

• If the base agent uses a linear function approximator, one can use tile coding over the parameters as before
Changes in Transfer Algorithm

- Transfer method directly affects CMDP state representation and transition function
- CMDP states represent “states of knowledge,” where knowledge represented as VF, shaping reward, etc.
- Similar process can be done if knowledge parameterizable
Experimental Results

• Evaluate whether curriculum policies can be learned

• Grid world
  • Multiple base agents
  • Multiple CMDP state representations

• Pacman
  • Multiple transfer learning algorithms
  • How long to train on sources?
Grid world Setup

Agent Types

• Basic Agent
  • State: Sensors on 4 sides that measure distance to keys, locks, etc.
  • Actions: Move in 4 directions, pickup key, unlock lock

• Action-dependent Agent
  • State difference: weights on features are shared over 4 directions

• Rope Agent
  • Action difference: Like basic, but can use rope action to negate a pit

CMDP Representations

• Finite State Representation
  • For discrete domains, groups and normalizes raw weights state-by-state to form CMDP features

• Continuous State Representation
  • Directly uses raw weights of learning agent as features for CMDP agent
Basic Agent Results

![Graph showing cost to learn target task over CMDP episodes]

- **Cost to Learn Target Task**
- **CMDP Episodes**

Legend:
- no curriculum
- Narvekar et al. (2017)
- finite state representation
- continuous state representation
- naive length 2 representation
- naive length 3 representation
Action-Dependent Agent Results

![Graph showing cost to learn target task over CMDP episodes]

- no curriculum
- Narvekar et al. (2017)
- finite state representation
- continuous state representation
- naive length 2 representation
- naive length 3 representation
Rope Agent Results

![Graph showing the cost to learn target task over CMDP episodes. The graph includes data for different representations and a curriculum approach.](image)

- **No Curriculum**
- **Narvekar et al. (2017)**
- **Finite State Representation**
- **Continuous State Representation**
- **Naive Length 2 Representation**
- **Naive Length 3 Representation**
Pacman Setup

Agent Representation

• Action-dependent egocentric features

CMDP Representation

• Continuous State Representation
  • Directly uses raw weights of learning agent as features for CMDP agent

Transfer Methods

• Value Function Transfer
• Reward Shaping Transfer

How long to train on a source task?
Pacman Value Function Transfer

![Graph showing the cost to learn target tasks with different representations.

- Blue line: no curriculum
- Green line: continuous state representation
- Red line: naive length 1 representation
- Teal line: naive length 2 representation]

CMDP Episodes vs. Cost to Learn Target Task
Pacman Reward Shaping Transfer

Cost to Learn Target Task

CMDP Episodes

- no curriculum
- Svetlik et al. (2017)
- continuous state representation
- naive length 2 representation
How long to train?

![Graph showing cost to learn target task vs CMDP Episodes]
Related Work

Restrictions on source tasks
• Florensa et al. 2018, Riedmiller et al. 2018, Sukhbaatar et al. 2017

Heuristic based sequencing
• Da Silva et al. 2018, Svetlik et al. 2017

MDP/POMDP based sequencing
• Matiisen et al. 2017, Narvekar et al. 2017

CL for supervised learning
Summary

• Generalize/Formulate curriculum generation as an MDP

• Demonstrate curriculum policies can be learned, and is robust to:
  • Learning agent state/action representation
  • CMDP representations
  • Transfer algorithm used