Lifelong / Meta / Transfer Learning

Emma Brunskill
Stanford
RL Summer School 2018
Learning to Solve a New (RL) Task
Most RL Agents Start From Scratch
Cornerstone of Intelligence Behavior: Use Prior Experience To Solve New Tasks

Emma Brunskill        Stanford University              @aiforhi
Transfer / Multi-task / Meta RL
Common Settings

Transfer:

[Diagram of a car and a motorcycle]
Common Settings

Transfer:

Lifelong:
Common Settings

Transfer:

Lifelong:

Multitask:
Common Settings

Transfer:

Lifelong:

Multitask:

Many → Many:
Common Settings

Transfer:

Lifelong:

Multitask:

Many → Many:
Tabular vs Function Approximation
Evaluating Success in Transfer RL

![Graph showing performance over training time with annotations for time to threshold, asymptotic performance, jumpstart, and transfer/no transfer thresholds.]

---

Taylor & Stone JMLR 2009

Emma Brunskill  Stanford University  @aiforhi  https://cs.stanford.edu/people/ebrun/
Also, Provably Better Learning?

![Graph showing performance over training time with labels for time to threshold, asymptotic performance, jumpstart, transfer, no transfer, and threshold performance.](Taylor & Stone JMLR 2009)
Two Core Parts of Multi-Task / Meta RL

• Summarize experience across tasks
• Use summary to improve learning in new task
Two Core Parts of Multi-Task / Meta RL

• Summarize experience across tasks
  • As dynamics / rewards models?
  • As value functions?
  • As policies?
• Use summary to improve learning in new task
Two Core Parts of Multi-Task / Meta RL

• Summarize experience across tasks
• Use summary to improve learning in new task
Rest of This Talk

• Summarize experience across tasks
  • As a finite set of tasks (clustering)
  • As a low dimensional subspace
  • As a set of parameters near to desired set

• Use summary to improve learning in new task
  • As initialization to standard RL algorithm
  • To new RL algorithm to direct exploration
Rest of This Talk

• Summarize experience across tasks
  • As a finite set of tasks (clustering)
  • As a low dimensional subspace
  • As a set of parameters near to desired set

• Use summary to improve learning in new task
  • As initialization to standard RL algorithm
  • To new RL algorithm to direct exploration
Setting

Lifelong

Tabular
All Tasks Very Different
All Tasks Identical

Emma Brunskill        Stanford University              @aiforhi             https://cs.stanford.edu/people/ebrun/
Finite Set of Tasks
Nikolaidis et al. HRI 2015
No apriori “labels” of similarity

Before try to learn this, if we knew the set of tasks, does it improve RL?
Two Core Parts of Multi-Task / Meta RL

• Summarize experience across tasks
  • As a finite set of tasks (clustering)
  • As a low dimensional subspace
  • As a set of parameters near to desired set
• Use summary to improve learning in new task
  • As initialization to standard RL algorithm
  • To new RL algorithm to direct exploration
If Know New Task is 1 of M Known Tasks, Can That Provably Improve Performance? (Spoiler: Yes!)
RL with Policy Advice

Azar, Lazaric, Brunskill, ECML 2013

• Assumptions: New task sampled from M tasks
• Evaluation goal: Provably improve performance
• Approach: Leverage known M set of policies
RL with Policy Advice

Azar, Lazaric, Brunskill, ECML 2013

\( \pi_1 \)  \hspace{2cm} \pi_2 \hspace{2cm} \pi_3
Quick Recap: Evaluating Performance

\[ \rho(\pi_k) \]

Return

Episodes

Emma Brunskill        Stanford University              @aiforhi             https://cs.stanford.edu/people/ebrun/
Regret Bounds

Return $\rho(\pi_k)$ vs. Episodes

Optimal return $\rho^*$
Regret Bounds: \( R(T) = T \rho^* - \sum_{k=1}^{T} \rho(\pi_k) \)
Provably Better Learning w/M Policies

Azar, Lazaric, Brunskill, ECML 2013

• Regret $\propto \sqrt{M}$ (independent of domain size)
Sequential Transfer

• Assumptions: New task sampled from M tasks
• Evaluation criteria: Provably speed learning
• Approach: Leverage known M set of models
RL → (Active) Classification

Brunskill & Li, UAI 2013
Maintain Hypothesis Set of Potential Identity of Current Task

Brunskill & Li, UAI 2013

Act in current task
\(<s_1, a_1, r_1, s_2, a_2, r_2, ...>\)
Direct Exploration to Quickly Identify Task*

Brunskill & Li, UAI 2013

Act in current task
<s_1,a_1,r_1,s_2,a_2,r_2,s_3,a_3,r_3>
Grid World Example: Directed Exploration
Intuition: Why Should This Speed Learning?

- If MDPs agree (have same model parameters) for most (s,a) pairs, only a few (s,a) pairs need to visit
  - To classify task
  - To learn parameters (all others are known)
- If MDPs differ in most (s,a) pairs, easy to classify task

Act in it for H steps
\(<s_1, a_1, r_1, s_2, a_2, r_2, s_3, a_3, \ldots s_H>\)
Formalizing RL Learning Speed

\[ \rho(\pi_k) \]

Return

Episodes

Optimal return \( \rho^* \)

Emma Brunskill        Stanford University              @aiforhi             https://cs.stanford.edu/people/ebrun/
Formalizing RL Learning Speed

ρ*
Formalizing RL Learning Speed
Only Count Big Mistakes

$\mathcal{E}$

$\mathcal{N}_\epsilon$ Number of episodes with policies not $\epsilon$-close to optimal

$\rho^*$
Probably Approximately Correct RL

\[ \mathbb{P}(N_\epsilon \leq F(S, A, H, \epsilon, \delta)) \geq 1 - \delta \]

\[ N_\epsilon \quad \text{Number of episodes with policies not } \epsilon\text{-close to optimal} \]
Theorem 1: Given any $\epsilon$ and $\delta$, run Algorithm 1 for $T$ tasks, each for $H = O\left(DSA\left(\max\left(\frac{1}{\Gamma^2} \log \frac{T}{\delta}, SD^2\right)\right)\right)$ steps. Then, the algorithm will select an $\epsilon$-optimal policy on all but at most $\tilde{O}\left(\frac{\zeta V_{max}}{\epsilon (1-\gamma)}\right)$ steps, with probability at least $1 - \delta$, where

$$\zeta = O\left(T_1 \zeta_s + \bar{C} \zeta_s + (T - T_1) \frac{\bar{C}D}{\Gamma^2}\right),$$
How Learn These Clusters?

- Summarize experience across tasks
  - As a finite set of tasks (clustering)
    - As a low dimensional subspace
    - As a set of parameters near to desired set
- Use summary to improve learning in new task
  - As initialization to standard RL algorithm
  - To new RL algorithm to direct exploration
Sequential Multitask Learning Across Finite Set of Markov Decision Processes

Sample a task from finite set of MDPs

Emma Brunskill     Stanford University     @aiforhi     https://cs.stanford.edu/people/ebrun/
Act in it for \( H \) steps:
\[
<s_1,a_1,r_1,s_2,a_2,r_2,s_3,a_3,\ldots s_H>
\]
Again sample a MDP…
Act in it for $H$ steps

$$<s_1,a_1,r_1,s_2,a_2,r_2,s_3,a_3,\ldots,s_H>$$
Series of tasks
Act in each task for H steps
Latent Variable Modeling

MDP R
T=? R=?

MDP G
T=? R=?

MDP Y
T=? R=？

…
Latent Variable Modeling

\[ \langle s_1, a_1, r_1, s' \rangle, \langle s_2, a_2, r_2, s' \rangle, \langle s_3, a_3, r_3, s' \rangle, \langle s_4, a_4, r_4, s' \rangle \]

\[ \text{Observed data} \]

MDP R
\[ T_R, R_R \]

MDP Y
\[ T_Y, R_Y \]

MDP G
\[ T_G, R_G \]
Latent Variable Modeling

$<s_{11}, a_{11}, r_{11}, s_{12}', a_{12}', r_{12}', s_{13}', a_{13}', \ldots>_{1H}$

$<s_{21}, a_{21}, r_{21}, s_{22}', a_{22}', r_{22}', s_{23}', a_{23}', \ldots>_{2H}$

$<s_{31}, a_{31}, r_{31}, s_{32}', a_{32}', r_{32}', s_{33}', a_{33}', \ldots>_{3H}$

$<s_{41}, a_{41}, r_{41}, s_{42}', a_{42}', r_{42}', s_{43}', a_{43}', \ldots>_{4H}$

$\text{MPD R}$

$T_R, R_R$

$\text{MPD Y}$

$T_Y, R_Y$

$\text{Observed data}$

$\text{Latent variable: Underlying MDP identity}$
Latent Variable Modeling

• Formally hard problem
• Expectation Maximization has weak theoretical guarantees
• Recent finite sample bounds on learned parameter estimates
Latent Variable Modeling

Assume for any 2 finite state—action MDPs $M_i$ & $M_j$, there exists at least one state—action pair such that

$$\|\theta_i(\cdot|s,a) - \theta_j(\cdot|s,a)\| > \Gamma$$

Vector of transition & reward parameters for (s,a) for MDP $M_j$

Note: to guarantee $\varepsilon$-optimal performance, very small differences in models are irrelevant. Implies above property always holds in discrete MDPs for some $\Gamma = f(\varepsilon)$
Implications

• Assume can visit any part of the decision making task an unbounded number of times
• If time horizon per task sufficiently long, can learn $O(\Gamma)$-accurate task parameters with high probability
→ Can correctly cluster tasks
Enables Provably Faster Learning in Finite Set of Tasks
Setting

Multitask:

Tabular
Multi-task RL

Or all customers using Amazon, or patients, or robot farm...
Provably Speeding Multitask RL

Guo and Brunskill, AAAI 2015

• Assumptions: K tasks sampled from M tasks
• Evaluation goal: Provably improve performance
• Approach: Quickly cluster and then share
Emma Brunskill  Stanford University  @aiforhi  https://cs.stanford.edu/people/ebrun/
Cluster Tasks
Going Forward
Share Data
Across Similar Tasks

Emma Brunskill       Stanford University       @aiforhi       https://cs.stanford.edu/people/ebrun/
If Clusters are Well Separated, → Cluster Quickly and Provably Speed Learning
Latent Variable Modeling for Provably Improved RL

• Separability assumptions
  – Concurrent RL (Guo & B., AAAI 2015)
  – Multi-task RL options learning (Li & B. ICML 2014)
  – Continuous-state multi-task RL (Liu, Guo & B. AAMAS 2016)

• Method of moments
  – Multi-task bandits (Azar, Lazaric and B NIPS 2013)
  – Multi-task Contextual latent bandits (Zhou and B, IJCAI 2016)
Offline Evaluation of Online Latent Contextual Bandit for News Personalization
Zhou and Brunskill IJCAI 2016
Two Core Parts of Multi-Task / Meta RL

• Summarize experience across tasks
  • As a finite set of tasks (clustering)
  • **As a low dimensional subspace**
    • As a set of parameters near to desired set

• Use summary to improve learning in new task
  • As initialization to standard RL algorithm
  • To new RL algorithm to direct exploration
Settings

Lifelong:

Multitask:

Function Approximation

Emma Brunskill        Stanford University              @aiforhi             https://cs.stanford.edu/people/ebrun/
Hidden Parameter MDPs: Smooth Latent Space Over Models

Doshi-Velez and Konidaris  IJCAI 2016

\[(s_d' - s_d) \sim \sum_k^K z_{kad} w_{kb} f_{kad}(s) + \epsilon\]
\[\epsilon \sim N(0, \sigma^2_{nad})\]
More Robust Hidden Parameter MDPs

Killian, Konidaris, Doshi-Velez. NIPS 2017

→ Use Bayesian Neural Networks for modeling the dynamics
Better Transfer on HIV Simulator Across Patients

Killian, Konidaris, Doshi-Velez. NIPS 2017
Smooth Latent Policy Space

Ammar, Eaton, Luna, Ruvolo, IJCAI 2015
Smooth Latent Policy Space for Cross Domain Transfer

Ammar, Eaton, Luna, Ruvolo, IJCAI 2015

- Set of policies with shared basis set of parameters
- Can be used to do cross domain transfer (different state & actions)
Two Core Parts of Multi-Task / Meta RL

• Summarize experience across tasks
  • As a finite set of tasks (clustering)
  • As a low dimensional subspace
  • As a set of parameters near to desired set

• Use summary to improve learning in new task
  • As initialization to standard RL algorithm
  • To new RL algorithm to direct exploration
Setting

Many $\rightarrow$ Many:

\[
\{\text{car, camel}\} \rightarrow \{\text{motorcycle, camel}\}
\]

Function Approximation
Inspiration: Pretraining

Slide from Sergey Levine
Review: Single Task Policy Gradient

\[ \theta'_i = \theta - \alpha \nabla_{\theta} \mathcal{L}_{\mathcal{T}_i}(f_{\theta}) \]

\[ \mathcal{L}_{\mathcal{T}_i}(f_{\phi}) = -\mathbb{E}_{x_t, a_t \sim f_{\phi}, q_{\mathcal{T}_i}} \left[ \sum_{t=1}^{H} R_i(x_t, a_t) \right] \]
How to Choose Initial Parameters to Speed Learning?

\[
\theta'_i = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)
\]

\[
\mathcal{L}_{\mathcal{T}_i}(f_\phi) = -\mathbb{E}_{x_t, a_t \sim f_\phi, q_{\mathcal{T}_i}} \left[ \sum_{t=1}^{H} R_i(x_t, a_t) \right]
\]
Parameters for Faster Future RL

Finn et al., “Model-Agnostic Meta-Learning” ICML 2017

$$\min_{\theta} \sum_{T_i \sim p(T)} \mathcal{L}_{T_i}(f_{\theta_i}) = \sum_{T_i \sim p(T)} \mathcal{L}_{T_i}(f_{\theta - \alpha \nabla_{\theta} \mathcal{L}_{T_i}(f_{\theta})})$$

set of tasks
Model Agnostic Meta-Learning

Finn et al., “Model-Agnostic Meta-Learning” ICML 2017

→ Learn $\theta$ so that it is “close” to good $\theta$ for many tasks:
One gradient step from $\theta$ on task yields high reward
Parameters for Faster Future RL
Finn et al., “Model-Agnostic Meta-Learning” ICML 2017

\[
\min_{\theta} \sum_{T_i \sim p(T)} \mathcal{L}_{T_i}(f_{\theta'}) = \sum_{T_i \sim p(T)} \mathcal{L}_{T_i}(f_{\theta - \alpha \nabla_{\theta} \mathcal{L}_{T_i}(f_{\theta})})
\]

set of tasks

Update meta-parameters \( \theta \) by SGD

\[
\theta \leftarrow \theta - \beta \nabla_{\theta} \sum_{T_i \sim p(T)} \mathcal{L}_{T_i}(f_{\theta'})
\]
MAML for RL

Finn et al., “Model-Agnostic Meta-Learning” ICML 2017

\textbf{Require:} \( p(\mathcal{T}) \): distribution over tasks
\textbf{Require:} \( \alpha, \beta \): step size hyperparameters

1: randomly initialize \( \theta \)
2: \textbf{while} not done \textbf{do}
3: \textbf{Sample batch of tasks} \( \mathcal{T}_i \sim p(\mathcal{T}) \)
4: \textbf{for all} \( \mathcal{T}_i \) \textbf{do}
5: \textbf{Sample} \( K \) trajectories \( \mathcal{D} = \{(x_1, a_1, ...x_H)\} \) using \( f_\theta \) in \( \mathcal{T}_i \)
6: \textbf{Evaluate} \( \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta) \) using \( \mathcal{D} \) and \( \mathcal{L}_{\mathcal{T}_i} \) in Equation 4
7: \textbf{Compute adapted parameters with gradient descent:} \( \theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta) \)
8: \textbf{Sample trajectories} \( \mathcal{D}_i' = \{(x_1, a_1, ...x_H)\} \) using \( f_{\theta_i'} \) in \( \mathcal{T}_i \)
9: \textbf{end for}
10: \textbf{Update} \( \theta \leftarrow \theta - \beta \nabla_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{\theta_i'}) \) using each \( \mathcal{D}_i' \) and \( \mathcal{L}_{\mathcal{T}_i} \) in Equation 4
11: \textbf{end while}
Meta-Learning Parameters

Finn et al., “Model-Agnostic Meta-Learning” ICML 2017
Slide from Sergey Levine

supervised learning: $f(x) \rightarrow y$

supervised meta-learning: $f(D_{\text{train}}, x) \rightarrow y$

model-agnostic meta-learning: $f_{MAML}(D_{\text{train}}, x) \rightarrow y$

$$f_{MAML}(D_{\text{train}}, x) = f_{\theta'}(x)$$

$$\theta' = \theta - \alpha \sum_{(x, y) \in D_{\text{train}}} \nabla_{\theta} \mathcal{L}(f_{\theta}(x), y)$$

Just another computation graph…
Can implement with any autodiff package (e.g., TensorFlow)
But has favorable inductive bias…

Emma Brunskill Stanford University @aiforhi https://cs.stanford.edu/people/ebrun/
Train Meta-Parameters Across Set of Tasks
Model-agnostic meta-learning: accelerating PG

Finn et al., “Model-Agnostic Meta-Learning” ICML 2017

Many nice extensions (including model based)
Very helpful for 1-shot learning in related tasks

Emma Brunskill       Stanford University              @aiforhi             https://cs.stanford.edu/people/ebrun/
Two Core Parts of Multi-Task / Meta RL

• Summarize experience across tasks
  • As a finite set of tasks (clustering)
  • As a low dimensional subspace
  • As a set of parameters near to desired set

• Use summary to improve learning in new task
  • As initialization to standard RL algorithm
  • To new RL algorithm to direct exploration
Open Questions & Directions

• Detecting and recovering from negative transfer
• Changing how to behave in current tasks to improve future performance on later tasks
• Curriculum design and meta-learning
Multi-Task / Meta RL

• Summarize experience across tasks
  • As a finite set of tasks (clustering)
  • As a low dimensional subspace
  • As a set of parameters near to desired set

• Use summary to improve learning in new task
  • As initialization to standard RL algorithm
  • To new RL algorithm to direct exploration