Categorizing Transfer for Reinforcement Learning

Matthew E. Taylor and Peter Stone

taylorm@usc.edu
Computer Science Department
University of Southern California

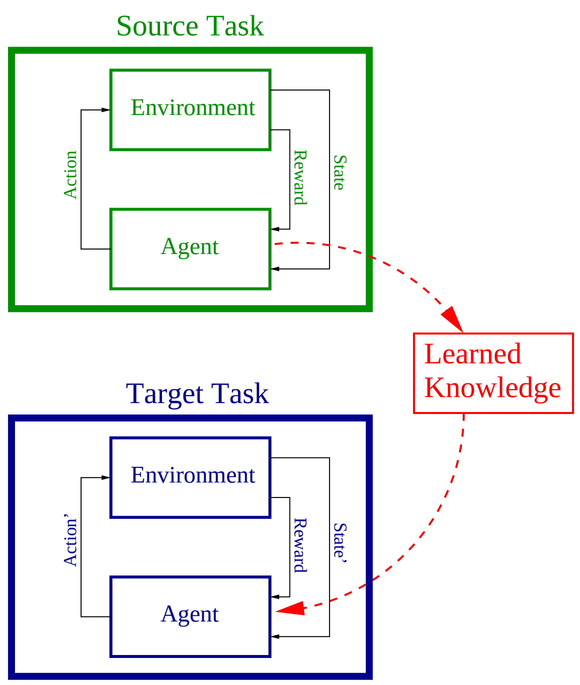
pstone@cs.utexas.edu
Department of Computer Sciences
University of Texas at Austin

Based on the forthcoming JMLR article: *Transfer Learning for Reinforcement Learning Domains: A Survey*

Transfer for Reinforcement Learning (RL)

Goals:

- Learn better performing policies faster
- Make difficult tasks tractable



Distinctions from Other Settings

Transfer Learning (TL)

- Use source task knowledge to learn target task
- Goal 1: Learn target task(s) better with past knowledge
- Goal 2: Learn sequence of tasks better than directly learning final task

Multi-task Learning (MTL)

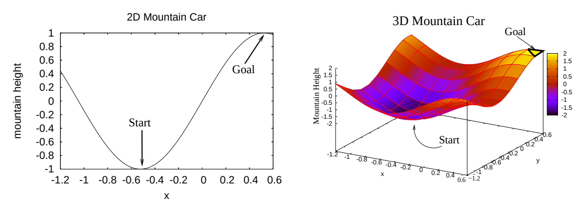
- Fixed (often known) distribution over tasks
- Goal: learn $n + 1^{th}$ task better

Many goals and metrics are used: no standard.

Related paradigms

- Lifelong learning: Tasks may be spatially (and temporally) separated, agents identify new tasks autonomously
- Imitation Learning: Observe an outside actor rather than reuse own knowledge
- Human Advice: Human integrated in the loop to give on-line feedback
- Shaping: Human directs training process (e.g., reward shaping)

Algorithm Differences



Possible task differences (examples)

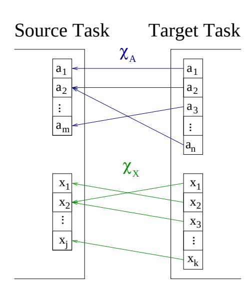
- Transition function
- Start state
- State space
- State variables
- Actions

Source task selection (examples)

- One task, human selected
- Multiple tasks, use all
- Multiple tasks, use some

Type of knowledge transferred

1. Low level examples
   - Instances
   - Q-value function
   - Policy
2. High level examples
   - Partial policies / options
   - Shaping reward
   - Important features
   - Rules



Inter-task mappings

- Not used
- Not needed (per agent learning algorithm)
- Used: allows for different actions and/or state variables
- Learned

Compatible learning methods (examples)

- Temporal difference methods
- Bayesian learning
- Batch
- Relational Reinforcement Learning

Selection of Open Questions

Theoretical results

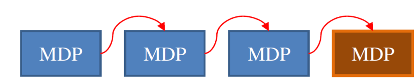
- Majority of results are empirical
- Guarantee improvement for pair of tasks
- Define relationship between amount/quality of knowledge and improvement
- Find an optimal inter-task mapping

Negative transfer

- Transfer can be **harmful** for a pair of tasks
- Identify incompatible pairs of tasks (per TL method)
- Identify when transfer is harming learner (on-line) in target task

Concept drift

- In transfer, new tasks are typically announced and changes are discrete
- What if tasks **change gradually** over time?
- What if agent is not told when it enters a new task?



Task sequence construction

- Given a target task, one may construct/select a sequence of source tasks
- Goal: **reduce total training time**
- What is the best way to select this sequence?
- Meta-planning problem

New Directions

- Transfer in repeated normal form games or stochastic games?

	Cooperate	Defect
Cooperate	-1, -1	-5, 0
Defect	0, -5	-2, -2

- Transfer in POMDPs?
- Learn multiple RL tasks simultaneously (MTL)?
- Develop a domain-independent metric for TL performance?

Acknowledgements

This work has taken place in the Learning Agents Research Group (LARG) at the Artificial Intelligence Laboratory, The University of Texas at Austin. LARG research is supported in part by grants from the National Science Foundation (CNS-0615104), DARPA (FA8750-05-2-0283 and FA8650-08-C-7812), the Federal Highway Administration (DTFH61-07-H-00030), and General Motors.

Paper	Allowed Task Differences	Source Task Selection	Task Mappings	Transferred Knowledge	Allowed Learners	TL Metrics
Same state variables and actions						
Selfridge et al. (1958)	t	h	N/A	Q	TD	tt†
Asada et al. (1994)	s_i	h	N/A	Q	TD	tt
Singh (1992)	r	all	N/A	Q	TD	ap, tr
Atkeson and Santamaria (1997)	r	all	N/A	model	MB	ap, j, tr
Asadi and Huber (2007)	r	h	N/A	π_p	H	tt
Andre and Russell (2002)	r, s	h	N/A	π_p	H	tr
Ravindran and Barto (2003b)	s, t	h	N/A	π_p	TD	tr
Ferguson and Mahadevan (2006)	r, s	h	N/A	pvf	Batch	tt
Sherstov and Stone (2005)	s_f, t	mod	N/A	A	TD	tr
Madden and Howley (2004)	s, t	all	N/A	rule	TD	tt, tr
Lazaric (2008)	s, t	lib	N/A	I	Batch	j, tr
Multi-Task learning						
Mehta et al. (2008)	r	lib	N/A	π_p	H	tr
Perkins and Precup (1999)	t	all	N/A	π_p	TD	tt
Foster and Dayan (2004)	s_f	all	N/A	sub	TD, H	j, tr
Fernandez and Veloso (2006)	s_i, s_f	lib	N/A	π	TD	tr
Tanaka and Yamamura (2003)	t	all	N/A	Q	TD	j, tr
Sunmola and Wyatt (2006)	t	all	N/A	pri	B	j, tr
Wilson et al. (2007)	r, s_f	all	N/A	pri	B	j, tr
Walsh et al. (2006)	r, s	all	N/A	fea	any	tt
Lazaric (2008)*	r	all	N/A	fea	Batch	ap, tr
Different state variables and actions – no explicit task mappings						
Konidaris and Barto (2006)	p	h	N/A	R	TD	j, tr
Konidaris and Barto (2007)	p	h	N/A	π_p	TD	j, tr
Banerjee and Stone (2007)	a, v	h	N/A	fea	TD	ap, j, tr
Guestrin et al. (2003)	#	h	N/A	Q	LP	j
Croonenborghs et al. (2007)	#	h	N/A	π_p	RRL	ap, j, tr
Ramon et al. (2007)	#	h	N/A	Q	RRL	ap, j, tt†, tr
Sharma et al. (2007)	#	h	N/A	Q	TD, CBR	j, tr
Different state variables and actions – inter-task mappings used						
Taylor et al. (2007a)	a, v	h	sup	Q	TD	tt†
Taylor et al. (2007b)	a, v	h	sup	π	PS	tt†
Taylor et al. (2008b)	a, v	h	sup	I	MB	ap, tr
Torrey et al. (2005) / Torrey et al. (2006)	a, r, v	h	sup	rule	TD	j, tr
Torrey et al. (2007)	a, r, v	h	sup	π_p	TD	j, tr
Taylor and Stone (2007b)	a, r, v	h	sup	rule	any/TD	j, tt†, tr
Learning inter-task mappings						
Kuhlmann and Stone (2007)	a, v	h	T	Q	TD	j, tr
Liu and Stone (2006)	a, v	h	T	N/A	all	N/A
Soni and Singh (2006)	a, v	h	M_a, sv_g, exp	N/A	all	ap, j, tr
Talvite and Singh (2007)	a, v	h	M_a, sv_g, exp	N/A	all	j
Taylor et al. (2007b)*	a, v	h	sv_g, exp	N/A	all	tt†
Taylor et al. (2008c)	a, v	h	exp	N/A	all	j, tr

This table classifies TL methods in terms of the five dimensions. Two entries, marked with a ⋆, are repeated due to multiple contributions. Metrics that account for source task learning time, rather than ignoring it, are marked with a †.

Allowed Task Differences

a	action set may differ
p	problem-space may differ (agent-space must be identical)
r	reward function may differ
s_i	the start state may change
s_f	goal state may move
t	transition function may differ
v	state variables may differ
#	number of objects in state may differ

Source Task Selection

all	all previously seen tasks are used
h	one source task is used (human selected)
lib	tasks are organized into a library and one or more may be used
mod	a human provides a source task that the agent automatically modifies

Task Mappings

exp	agent learns the mappings from experience
M_a	the method must be provided with an action mapping (learns state variable mapping)
N/A	no mapping is used
sup	a human supplies the task mappings
sv_g	method is provided groupings of state variables
T	higher-level knowledge is provided about transfer functions to learn mapping

Transferred Knowledge

A	an action set
fea	task features
I	experience instances
model	task model
π	policies
π_p	partial policies (e.g., options)
pri	distribution priors
pvf	proto-value function
Q	action-value function
R	shaping reward
rule	rules or advice
sub	subtask definitions

Allowed Learners

B	Bayesian learner
Batch	batch learner
CBR	case based reasoning
H	hierarchical value-function learner
LP	linear programming
MB	model based learner
PS	policy search learner
RRL	relational reinforcement learning
TD	temporal difference learner

TL Metrics

ap	asymptotic performance increased
j	jumpstart demonstrated
tr	total reward increased
tt	task learning time reduced