Section Outline

- Introduction to RL
- The dimensions of transfer
  - task relatedness
  - transferred knowledge
  - learning algorithms
- Transfer between tasks with same state-action variables
  - From one source task to one target task
  - From many source tasks to one target task
  - Multitask learning: Learning a distribution of tasks
- Transfer between tasks with different state-action variables
  - No explicit mapping
  - Mapping state variables and actions between tasks
  - Learning the inter-task mapping

Problem-Independent Learning

- Konidaris and Barto
  - "Autonomous shaping: Knowledge transfer in reinforcement learning," 2006
  - "Building portable options: Skill transfer in reinforcement learning," 2007
- Agent Space
  - Possibly non-Markovian
  - Capabilities: physical sensors, actuators
- Problem Space
  - Markovian
  - May change between tasks

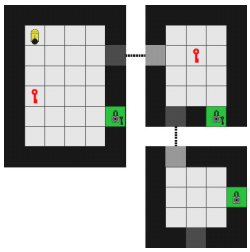
Transfer between tasks with different SxA

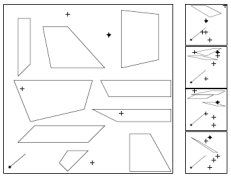
- Problems Arise
  - How to handle new features or actions?
  - Ignore?
  - What if ordering of features/actions change?

- *The idea*: reason over agent-space and problem-space
- *Task difference*: problem-space
- *Transferred knowledge*: shaping reward / options
- *Learning algorithm*: sarsa
- *Metric*: learning speed

Konidaris and Barto (2006, 2007)

- Learn shaping reward in Agent-Space
- Learn agent-space options





Transfer in RRL

"Learning relational options for inductive transfer in relational reinforcement learning," (Croonenborghs et al., 2007)

- Explicit mapping not necessarily needed
- Actions
  - move(A,B)
- State
  - clear(A)
  - above(A,X)

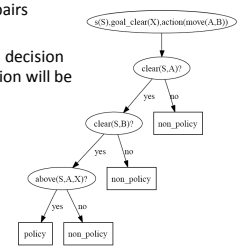
RRL Transfer

- *The idea*: use relational representation
- *Task difference*: # of objects
- *Transferred knowledge*: options
- *Learning algorithm*: TGR
- *Metric*: learning speed

Croonenborghs et al., 2007

Option Learning in Blocksworld

1. use an RRL algorithm to learn to solve this task
2. create training examples of state-action pairs labeled as policy-based or not
3. use the TILDE system to learn a relational decision tree that predicts whether or not the action will be executed by the policy
4. extract a relational policy



clear(X) skill

$$\begin{cases} s(S), goal\_clear(X) : clear(S,A), clear(S,B), above(S,A,X) \rightarrow move(A,B) \\ s(S), goal\_clear(X) : random\_block(X), random\_block(Y) \rightarrow move(A,B) \end{cases}$$

relational policy

Section Outline

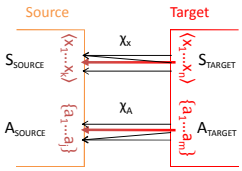
- Introduction to RL
- The dimensions of transfer
  - task relatedness
  - transferred knowledge
  - learning algorithms
- Transfer between tasks with same state-action variables
  - From one source task to one target task
  - From many source tasks to one target task
  - Multitask learning: Learning a distribution of tasks
- Transfer between tasks with different state-action variables
  - No explicit mapping
  - Mapping state variables and actions between tasks
  - Learning the inter-task mapping

Value Function Transfer

- *The idea*: transfer Q-values using inter-task mappings
- *Task difference*: state features and actions
- *Transferred knowledge*: Q-values
- *Learning algorithm*: sarsa
- *Metric*: learning speed+

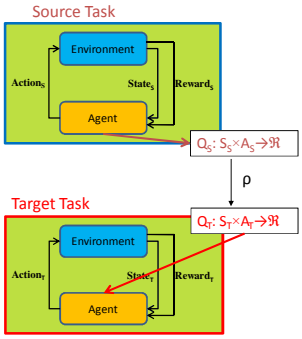
Inter-Task Mappings (Taylor and Stone, 2005)

- χ_x: $s_{target} \rightarrow s_{source}$
  - Given state variable in target task (some x from $s=x_1, x_2, \ldots x_n$)
  - Return corresponding state variable in source task
- χ_A: $a_{target} \rightarrow a_{source}$
  - Similar, but for actions



- Intuitive mappings exist in some domains (Oracle)
- Used to construct transfer functional

Value Function Transfer

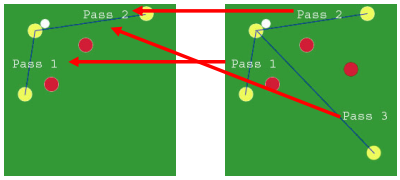


- "Transfer learning via inter-task mappings for temporal difference learning," Taylor et al, 2007
- Q not defined on S' and A'
- ρ(Q(S,A)) = Q'(S',A')
- Action-Value function transferred
- ρ is task-dependant:
- relies on inter-task mappings

Keepaway Hand-coded χA

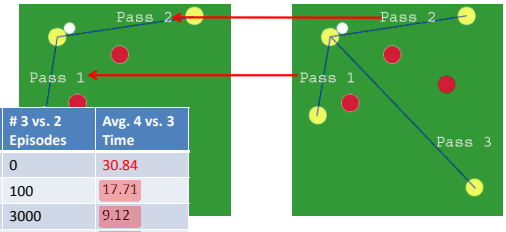
Actions in 4 vs. 3 have "similar" actions in 3 vs. 2

- $Hold_{4v3} \rightarrow Hold_{3v2}$
- $Pass1_{4v3} \rightarrow Pass1_{3v2}$
- $Pass2_{4v3} \rightarrow Pass2_{3v2}$
- $Pass3_{4v3} \rightarrow Pass2_{3v2}$



Value Function Transfer Flexibility

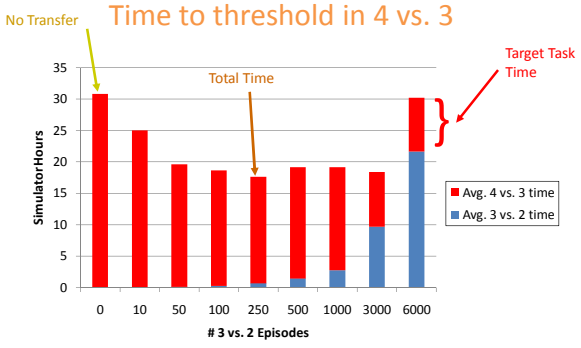
- Different Function Approximators
- Different Actuators
- Different Keepaway Tasks
- Partial Mappings



Transfer Functional	# 3 vs. 2 Episodes	Avg. 4 vs. 3 Time
none	0	30.84
Full	100	17.71
Full	3000	9.12
Partial	100	18.90
Partial	3000	12.00

Value Function Transfer:

Time to threshold in 4 vs. 3



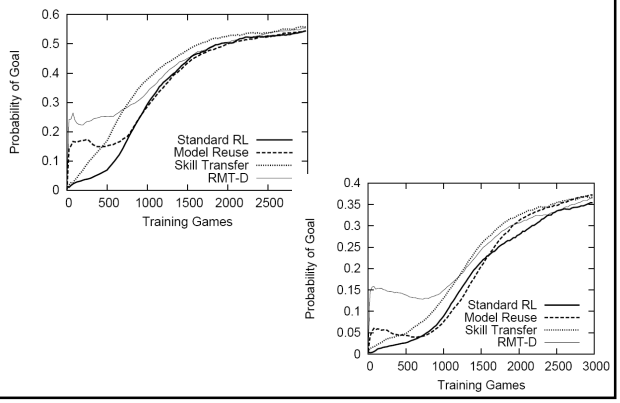
- *The idea*: learn and transfer strategies
- *Task difference*: state features and actions
- *Transferred knowledge*: options
- *Learning algorithm*: sarsa
- *Metric*: learning speed

Strategy Transfer

"Relational macros for transfer in reinforcement learning," (Torrey et al., 2007)

1. Structure Learning
   - ID sequences of successful actions
   - Compose into Finite State Machine
2. Ruleset Learning
   - Learn when action in a strategy should be taken
   - Aleph
3. Remap strategies via human
4. DemonstrateStrategy
5. Sarsa(λ) with SVM function approximation

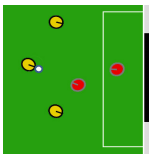
Torrey et al., 2007: Results



Torrey et al., 2007: Example

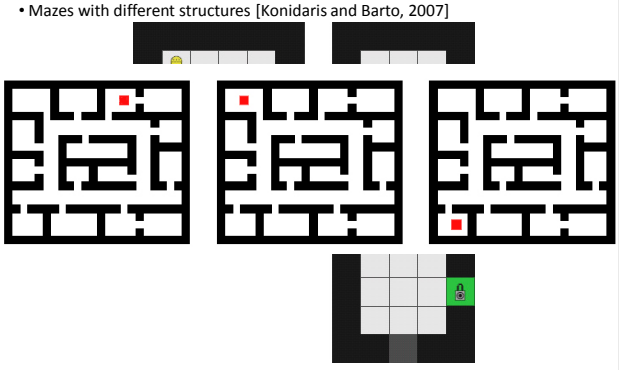
- Relational Macro Transfer via Demonstration (RMT-D)
- Macro learned in 2-on-1 Breakaway

- Demonstrated in 3-on-2 Breakaway



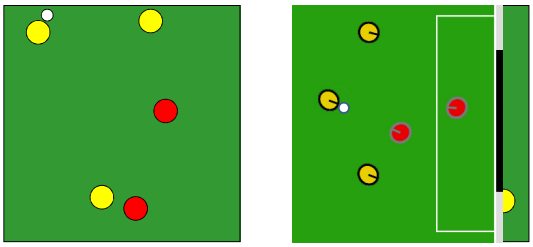
Cross Domain Transfer?

- Series of mazes with different goals [Fernandez and Veloso, 2006]
- Mazes with different structures [Konidaris and Barto, 2007]

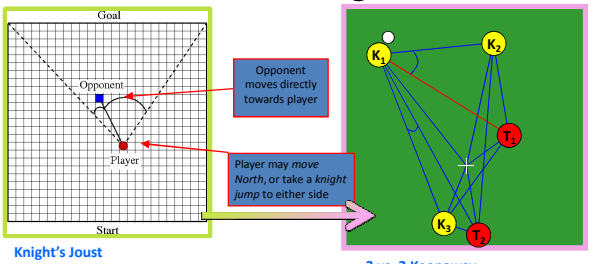


Related Transfer Work

- Series of mazes with different goals [Fernandez and Veloso, 2006]
- Mazes with different structures [Konidaris and Barto, 2007]
- Keepaway with different numbers of players [Taylor and Stone, 2005]
- Keepaway to Breakaway [Maclin et al, 2005]



Source Task: Knight's Joust



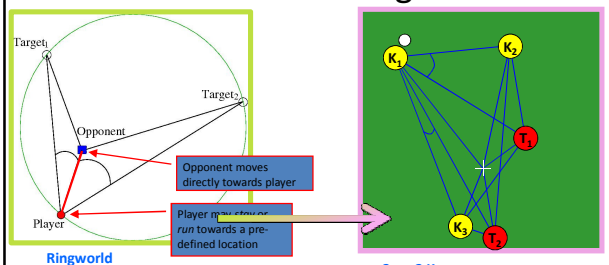
Knight's Joust
Goal: Travel from start to goal line

2 agents
3 actions
3 state variables
Fully Observable
Discrete State Space (Q-table with ~600 s,a pairs)
Deterministic Actions

3 vs. 2 Keepaway
Goal: Maintain possession of ball

5 agents
3 actions
13 state variables
Partially Observable
Continuous State Space
Stochastic Actions

Source Task: Ringworld



Ringworld
Goal: avoid being tagged

2 agents
3 actions
7 state variables
Fully Observable
Discrete State Space (Q-table with ~8,100 s,a pairs)
Stochastic Actions

3 vs. 2 Keepaway
Goal: Maintain possession of ball

5 agents
3 actions
13 state variables
Partially Observable
Continuous State Space
Stochastic Actions

- *The idea*: learn and transfer learning-independent rules
- *Task difference*: state features and actions
- *Transferred knowledge*: rules
- *Learning algorithm*: any / sarsa
- *Metric*: learning speed

Rule Transfer Overview

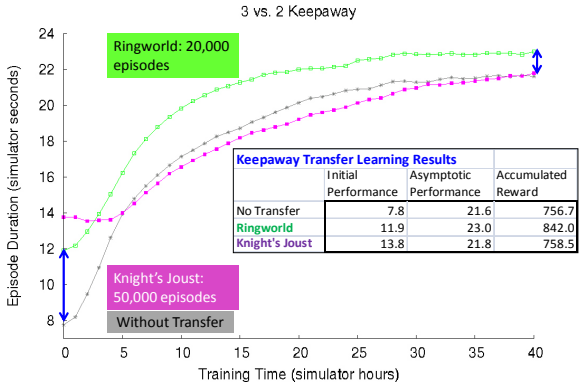
1. Learn a policy ($\pi : S \rightarrow A$) in the source task
   - TD, Policy Search, Model-Based, etc.
2. Learn a decision list, D_{source}, summarizing π
3. Translate (D_{source}) → D_{target} (applies to target task)
   - State variables and actions can differ in two tasks
4. Use D_{target} to learn a policy in target task

Allows for different learning methods and function approximators in source and target tasks

Section Outline

- Introduction to RL
- The dimensions of transfer
  - task relatedness
  - transferred knowledge
  - learning algorithms
- Transfer between tasks with same state-action variables
  - From one source task to one target task
  - From many source tasks to one target task
  - Multitask learning: Learning a distribution of tasks
- Transfer between tasks with different state-action variables
  - No explicit mapping
  - Mapping state variables and actions between tasks
  - Learning the inter-task mapping

"Cross-domain Transfer for Reinforcement Learning," Taylor and Stone 2007



Keepaway Transfer Learning Results

	Initial Performance	Asymptotic Performance	Accumulated Reward
No Transfer	7.8	21.6	756.7
Ringworld	11.9	23.0	842.0
Knight's Joust	13.8	21.8	758.5

Learning Task Relationships

- Sometimes task relationships are unknown
- Necessary for Autonomous Transfer
- But finding similarities (analogies) can be very hard!
- Key ideas:
  1. Data Driven
     - Agents may generate data (experience) in both tasks
     - Leverage existing machine learning techniques
  2. Domain Analysis
     - Find similarities in problem structure

135

MASTER Overview

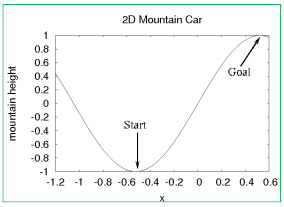
"Transferring instances for Model-Based Reinforcement Learning", Taylor et al., 2008

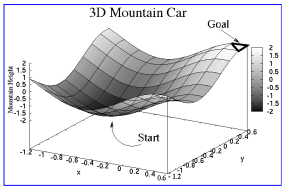
Modeling Approximate State Transitions by Exploiting Regression

- Goals:
  - Learn inter-task mapping between tasks
  - Minimize data complexity
  - No background knowledge needed
- Overview:
  1. Record data in source task
  2. Record small amount of data in target task
  3. Analyze data off-line to determine best mapping
  4. Use mapping in target task

Observations

- Pros:
  - Very little target task data needed (sample complexity)
  - Analysis for discovering mappings is off-line
- Cons:
  - Exponential in # of state variables and actions





138

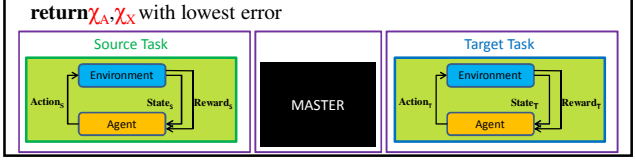
MASTER Algorithm

Record observed (s_{source}, a_{source}, s'_{source}) tuples in source task
Record small number of (s_{target}, a_{target}, s'_{target}) tuples in target task

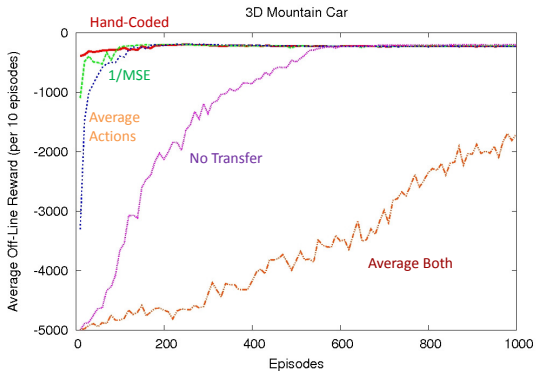
Learn one-step transition model, $T(S_T, A_T)$, for the target task:
$M(s_{target}, a_{target}) \rightarrow s'_{target}$

for every possible action mapping χ_A
 for every possible state variable mapping χ_X
  Transform recorded source task tuples
  Calculate the error of the transformed source task tuples on the target task model: $\sum(M(s_{transformed}, a_{transformed}) - s'_{transformed})^2$

return χ_A,χ_X with lowest error



Transfer in 3D Mountain Car



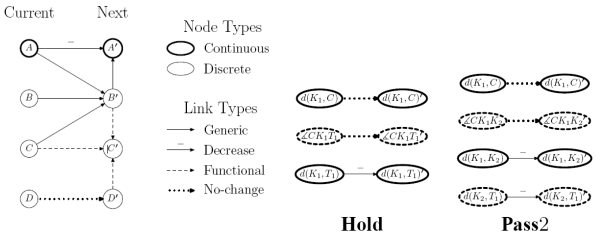
139

Other Data-Driven Methods

- Classification
  - Taylor et al., 2007
- Implicit search via multiple experts
  - Talvitie and Singh, 2007
- Homomorphisms
  - Soni and Singh, 2006
  - Sorg and Singh, 2009 at AAMAS-09

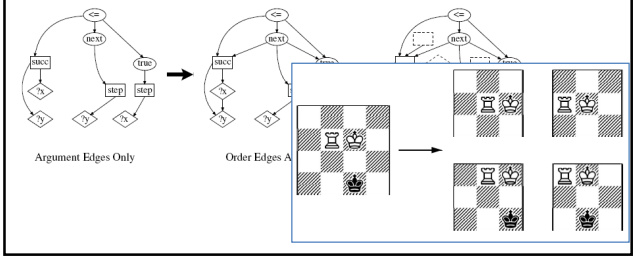
Domain Analysis Approach 2

- Qualitative Dynamic Bayes Networks (QDBN)
- "Value-function-based transfer for reinforcement learning using structure mapping," Liu and Stone, 2006
- Use variant of the Structure Mapping Engine



Domain Analysis Approach 1

- In GGP, complete model is given
- "Graph-based domain mapping for transfer learning in general games," Kuhlmann and Stone, 2007
  - Produces rule-graph for source task(s)
  - In target task, find isomorphic source state



Mapping Learning: Conclusions

- Method depends on knowledge available
- Never guaranteed to work
- Useful in the general case?