Transfer of Samples in Batch Reinforcement Learning

A. Lazaric  M. Restelli  A. Bonarini

Department of Electronics and Information  Politecnico di Milano, Italy

International Conference on Machine Learning, 2008  Helsinki, Finland
1. Introduction
   - Transfer in Reinforcement Learning

2. Transfer of Samples in Batch Reinforcement Learning
   - The Scenario
   - The Implementation

3. Experimental Results
   - The Boat Problem
   - Results

4. Summary
   - Conclusions & Future Works
Outline

1 Introduction
   - Transfer in Reinforcement Learning

2 Transfer of Samples in Batch Reinforcement Learning
   - The Scenario
   - The Implementation

3 Experimental Results
   - The Boat Problem
   - Results

4 Summary
   - Conclusions & Future Works
**Assumption:** Different tasks are somehow **related**

**Goal:** Develop algorithms to **find and exploit** this relatedness in order to **improve** the learning performance

**How:** **Retain knowledge** from a set of tasks and **transfer** it to new different tasks
Transfer in Reinforcement Learning

- **Assumption**: Different tasks are somehow related
- **Goal**: Develop algorithms to **find and exploit** this relatedness in order to **improve** the learning performance
- **How**: Retain **knowledge** from a set of tasks and **transfer** it to new different tasks
**Transfer in Reinforcement Learning**

- **Assumption**: Different tasks are somehow **related**
- **Goal**: Develop algorithms to **find and exploit** this relatedness in order to **improve** the learning performance
- **How**: **Retain knowledge** from a set of tasks and **transfer** it to new different tasks
State of the Art

What can be transferred?

- **Solutions**: value functions [Taylor et al., 2005], policies [Torrey et al., 2006] [Taylor et al., 2007][Madden & Howley, 2004]
- **Structure**: options [Konidaris & Barto, 2007][Şimşek et al., 2005][Perkins & Precup, 1999], hierarchical decomposition [Mehta et al., 2005], MDP abstraction [Walsh et al., 2006]
- **Experience**: samples \( \langle s, a, s', r \rangle \) [Taylor et al., 2008]
State of the Art

What can be transferred?

- **Solutions:** value functions [Taylor et al., 2005], policies [Torrey et al., 2006] [Taylor et al., 2007] [Madden & Howley, 2004]

- **Structure:** options [Konidaris & Barto, 2007] [Şimşek et al., 2005] [Perkins & Precup, 1999], hierarchical decomposition [Mehta et al., 2005], MDP abstraction [Walsh et al., 2006]

- **Experience:** samples \((s, a, s', r)\) [Taylor et al., 2008]
**State of the Art**

*What can be transferred?*

- **Solutions**: value functions [Taylor et al., 2005], policies [Torrey et al., 2006] [Taylor et al., 2007] [Madden & Howley, 2004]

- **Structure**: options
  - [Konidaris & Barto, 2007] [Şimşek et al., 2005] [Perkins & Precup, 1999], hierarchical decomposition [Mehta et al., 2005], MDP abstraction [Walsh et al., 2006]

- **Experience**: samples \( \langle s, a, s', r \rangle \) [Taylor et al., 2008]
State of the Art

What can be transferred?

- **Solutions**: value functions [Taylor et al., 2005],
  policies [Torrey et al., 2006] [Taylor et al., 2007][Madden & Howley, 2004]

- **Structure**: options
  [Konidaris & Barto, 2007][Şimşek et al., 2005][Perkins & Precup, 1999],
  hierarchical decomposition [Mehta et al., 2005], MDP
  abstraction [Walsh et al., 2006]

- **Experience**: samples \( \langle s, a, s', r \rangle \) [Taylor et al., 2008]
Outline

1. Introduction
   - Transfer in Reinforcement Learning

2. Transfer of Samples in Batch Reinforcement Learning
   - The Scenario
   - The Implementation

3. Experimental Results
   - The Boat Problem
   - Results

4. Summary
   - Conclusions & Future Works
Fact: In batch RL algorithms, the set of samples used to feed the learning algorithm influences the performance.

Goal: Transfer samples coming from other (source) tasks in order to improve the performance in a target task.

Problem: Avoid negative transfer.
The Goal

- **Fact:** In batch RL algorithms, the **set of samples** used to feed the learning algorithm influences the performance.

- **Goal:** Transfer **samples** coming from other (source) tasks in order to **improve** the performance in a target task.

- **Problem:** Avoid negative transfer.
**The Goal**

- **Fact**: In batch RL algorithms, the *set of samples* used to feed the learning algorithm influences the performance.

- **Goal**: Transfer *samples* coming from other (source) tasks in order to *improve* the performance in a target task.

- **Problem**: Avoid *negative transfer*.
All the tasks share the same *state-action* space.
The Scenario

The Scenario

Task Space

Task Distribution

\[ \mathcal{T} \]

\[ \Omega \]

\[ \sigma_j \]

\[ S_1 \]

\[ S_n \]

\[ m \text{ samples each} \]
The Scenario

Task Space

Task Distribution $\Omega$

$S_1$, $S_n$ with $m$ samples each

$T$ with $t \ll m$ samples
Which tasks is it convenient to transfer from?

We compute the average probability of each source task $S$ to be the model from which the target samples $(\tau_i = \langle s_i, a_i, s'_i, r_i \rangle)$ are generated, that is its compliance to the target task

\[
P(S|\tau_i) \propto P(\tau_i|S) P(S) = P_S(s'_i|s_i, a_i) R_S(r_i|s_i, a_i) P(S)
\]
Task Compliance

- **Which tasks is it convenient to transfer from?**
- We compute the *average probability* of each source task $S$ to be the model from which the target samples $(\tau_i = \langle s_i, a_i, s'_i, r_i \rangle)$ are generated, that is its *compliance* to the target task

$$
P(S|\tau_i) \propto P(\tau_i|S) P(S) \\
= \mathcal{P}_S(s_i'|s_i, a_i) R_S(r_i|s_i, a_i) P(S)$$
Which tasks is it convenient to transfer from?

- We compute the average probability of each source task $S$ to be the model from which the target samples $(\tau_i = \langle s_i, a_i, s'_i, r_i \rangle)$ are generated, that is its compliance to the target task

\[
P(S|\tau_i) \propto P(\tau_i|S) P(S)
= P_S(s'_i|s_i, a_i)R_S(r_i|s_i, a_i)P(S)
\]
Continuous Model Approximation

- \( P(\tau_i|S) =? \)
- We follow the kernel-based approximation proposed in [Jong & Stone, 2007]
- Given kernel function \( \varphi(\cdot) \),
  \[
  \sigma_j = \langle s_j, a_j, s'_j, r_j \rangle \in \hat{S}
  \]
- \[
  \mathcal{P}_{\hat{S}}(s'_i|s_i, a_i) \propto \sum_{j=1}^{m} w_j \cdot \varphi \left( \frac{d(s'_i, s_i + (s'_j - s_j))}{\delta_{s'_i}} \right)
  \]
- with weights \( w_j \) computed according to distance in the state-action space
Continuous Model Approximation

- $P(\tau_i|S) = ?$

- We follow the kernel-based approximation proposed in [Jong & Stone, 2007]

- Given kernel function $\varphi(\cdot)$,

$$\sigma_j = \langle s_j, a_j, s'_j, r_j \rangle \in \hat{S}$$

$$P_\hat{S}(s'_i|s_i, a_i) \propto \sum_{j=1}^{m} w_j \cdot \varphi \left( \frac{d(s'_i, s_i + (s'_j - s_j))}{\delta_{s'_i}} \right)$$

with weights $w_j$ computed according to distance in the state-action space
Continuous Model Approximation

- \( P(\tau_i | S) = \)?
- We follow the kernel-based approximation proposed in [Jong & Stone, 2007]
- Given kernel function \( \varphi(\cdot) \),
  \[
  \sigma_j = \langle s_j, a_j, s'_j, r_j \rangle \in \hat{S}
  \]
  \[
  P_{\hat{S}}(s'_i | s_i, a_i) \propto \sum_{j=1}^{m} w_j \cdot \varphi \left( \frac{d(s'_i, s_i + (s'_j - s_j))}{\delta_{s'_i}} \right)
  \]
  with weights \( w_j \) computed according to distance in the state-action space
Task Compliance

\[ \lambda_i = P(\tau_i | \tilde{S}) \]

\[ P_{\tilde{S}} \]
\[ R_{\tilde{S}} \]

\[ \tilde{S} \]

\[ T \]

\[ \tau_i \]
Task Compliance

\[ \lambda_1 = P(\tau_1 | \hat{S}) \]
\[ \lambda_2 = P(\tau_2 | \hat{S}) \]
\[ \cdots \]
\[ \lambda_t = P(\tau_t | \hat{S}) \]
Task Compliance

Definition

Given the target samples $\hat{T}$ and the source samples $\hat{S}$, the task compliance of $S$ is

$$\Lambda = \frac{1}{t} \sum_{i=1}^{t} \lambda_i P(S)$$
Task Compliance

Source Tasks

Target Task $T$

$\Lambda_1$

$S_1$

$\Lambda_2$

$S_2$

$\Lambda_3$

$S_3$
Sample Relevance

- Which samples are worth transferring?
- Also in highly compliant source tasks there may be regions where samples are much dissimilar from target samples
Sample Relevance

- *Which samples are worth transferring?*
- Also in highly compliant source tasks there may be regions where samples are much **dissimilar** from target samples.
Sample Relevance

- Given a **source** sample $\sigma_j \in \hat{S}$ and a model approximation of the target task $\hat{T}$

- Source sample compliance (normalized over all source samples): $\lambda_j = P(\sigma_j|\hat{T})$

- Unreliability of approximation $\hat{T}$ at $\sigma_j$: $d_j$
Sample Relevance

- Given a **source** sample \( \sigma_j \in \hat{S} \) and a model approximation of the target task \( \hat{T} \)
- Source sample compliance (normalized over all source samples): \( \lambda_j = P(\sigma_j|\hat{T}) \)
- Unreliability of approximation \( \hat{T} \) at \( \sigma_j \): \( d_j \)
Sample Relevance

- Given a **source** sample $\sigma_j \in \hat{S}$ and a model approximation of the target task $\hat{T}$
- Source sample compliance (normalized over all source samples): $\lambda_j = P(\sigma_j | \hat{T})$
- Unreliability of approximation $\hat{T}$ at $\sigma_j$: $d_j$
Sample Relevance

**Definition**

The *relevance* of $\sigma_j$ is defined as

$$\rho_j = \rho(\bar{\lambda}_j, d_j) = \exp \left( - \left( \frac{\bar{\lambda}_j - 1}{d_j} \right)^2 \right).$$

Transfer $\sigma_j$ whenever

- high probability to be generated by the target task (high $\lambda_j$)
- poor approximation (few samples) of $\hat{T}$ near $\sigma_j$ (high $d_j$)
Sample Relevance

Definition

The relevance of $\sigma_j$ is defined as

$$
\rho_j = \rho(\bar{\lambda}_j, d_j) = \exp \left( - \left( \frac{\bar{\lambda}_j - 1}{d_j} \right)^2 \right).
$$

Transfer $\sigma_j$ whenever

- high probability to be generated by the target task (high $\lambda_j$)
- poor approximation (few samples) of $\hat{T}$ near $\sigma_j$ (high $d_j$)
Sample Relevance

Definition

The relevance of $\sigma_j$ is defined as

$$\rho_j = \rho(\bar{\lambda}_j, d_j) = \exp \left( - \left( \frac{\bar{\lambda}_j - 1}{d_j} \right)^2 \right).$$

Transfer $\sigma_j$ whenever

- high probability to be generated by the target task (high $\lambda_j$)
- poor approximation (few samples) of $\hat{T}$ near $\sigma_j$ (high $d_j$)
Sample Relevance

Definition

The relevance of $\sigma_j$ is defined as

$$
\rho_j = \rho(\bar{\lambda}_j, d_j) = \exp \left( - \left( \frac{\bar{\lambda}_j - 1}{d_j} \right)^2 \right).
$$

Transfer $\sigma_j$ whenever

- high probability to be generated by the target task (high $\lambda_j$)
- poor approximation (few samples) of $\hat{T}$ near $\sigma_j$ (high $d_j$)
Sample Relevance

Source Tasks

Target Task $T$

$S_1$

$S_2$

$S_3$

Sample Relevance

Sample Relevance

Task Compliance

$\Lambda_1$

$\Lambda_2$

$\Lambda_3$
Transfer of Samples

The Scenario

The Implementation

The Scenario

The Implementation

Transfer of Samples

Source Tasks

Sample Relevance

Target Task $T$

$S_1$

$S_2$

$S_3$

Task Compliance

$\Lambda_1$

$\Lambda_2$

$\Lambda_3$

Augmented Target $\tilde{T}$

Lazaric, Restelli, Bonarini

Transfer of Samples in Batch Reinforcement Learning
The Boat Problem

- State: position $x, y$
- Action: rudder angle
- Reward: positive in the goal zone, negative out of boundaries and in the sand banks, zero elsewhere
- Dynamics: non-linear stochastic

Target Task

- sandbank1
- sandbank2
- G1
The Boat Problem

Hand-coded source tasks, see the paper for results with randomly generated tasks

Source Task $S_1$

Additional goal, no sandbank2
Hand-coded source tasks, see the paper for results with randomly generated tasks

Source Task $S_1$

Additional goal, no $sandbank2$

Source Task $S_2$

Different goal, sandbanks and current
Outline

1. Introduction
   - Transfer in Reinforcement Learning

2. Transfer of Samples in Batch Reinforcement Learning
   - The Scenario
   - The Implementation

3. Experimental Results
   - The Boat Problem
   - Results

4. Summary
   - Conclusions & Future Works
Transfer from $S_1$ and $S_2$ to $T$

**FQI with Extra Randomized Trees**

![Graph](image)

- $\pi^*_1$
- $\pi^*_2$

No Transfer

**Total Reward**

- $-10$
- $-20$
- $-30$
- $-40$
- $-50$
- $-60$
- $-70$
- $-80$

**Number of samples**

- 50
- 250
- 450
- 650
- 850
- 1050
- 1250
Transfer from $S_1$ and $S_2$ to $T$

Transfer of samples at random

Total Reward

Number of samples

No Transfer
Random

$\pi^*_1$

$\pi^*_2$
Transfer from $S_1$ and $S_2$ to $T$

- Most of the samples in $\hat{S}_2$ are completely different from samples in $\hat{T}$
- Normalized compliance
  $\bar{\Lambda}_1 = 0.93 \pm 0.09$,  
  $\bar{\Lambda}_2 = 0.07 \pm 0.06$
Transfer from $S_1$ and $S_2$ to $T$

- Most of the samples in $\hat{S}_2$ are completely different from samples in $\hat{T}$
- Normalized compliance
  $\bar{\Lambda}_1 = 0.93 \pm 0.09,$
  $\bar{\Lambda}_2 = 0.07 \pm 0.06$
Transfer from $S_1$ and $S_2$ to $T$

Transfer of samples proportionally to task compliance

![Graph showing the transfer of samples proportionally to task compliance. The graph plots Total Reward against Number of samples. The x-axis ranges from 50 to 1250 with intervals at 250, and the y-axis ranges from -80 to 0 with intervals at -20. The graph includes lines for No Transfer, Random, and Compliance, with markers and error bars. The lines are labeled as $\pi^*_1$ and $\pi^*_2$.](image)
Transfer from $S_1$ and $S_2$ to $T$

- Not all the samples from $S_1$ are worth transferring
- Avoid transferring samples in the region of sandbank2 and $G_2$
Transfer from $S_1$ and $S_2$ to $T$

- Not all the samples from $S_1$ are worth transferring
- Avoid transferring samples in the region of sandbank2 and $G_2$
Transfer from $S_1$ and $S_2$ to $T$

- Not all the samples from $S_1$ are worth transferring
- Avoid transferring samples in the region of $sandbank2$ and $G_2$
Transfer from $S_1$ and $S_2$ to $T$

*Transfer of samples proportionally to task compliance and sample relevance*

![Graph showing total reward vs. number of samples with different transfer methods](image-url)
Transfer from $S_1$ and $S_2$ to $T$

\[ r = \frac{\text{area of curve w/ transfer} - \text{area of curve w/o transfer}}{\text{area of curve w/o transfer}} \]
Outline

1. Introduction
   - Transfer in Reinforcement Learning

2. Transfer of Samples in Batch Reinforcement Learning
   - The Scenario
   - The Implementation

3. Experimental Results
   - The Boat Problem
   - Results

4. Summary
   - Conclusions & Future Works
Conclusions

Pros:

- **No need to solve the source tasks**
- More effective than transferring policies
- Works in any transfer scenario and with any batch RL algorithm
- Performance improvement even when few target samples available
Pros:
- No need to solve the source tasks
- More effective than transferring policies
- Works in any transfer scenario and with any batch RL algorithm
- Performance improvement even when few target samples available
Pros:

- No need to solve the source tasks
- More effective than transferring policies
- Works in any transfer scenario and with any batch RL algorithm
- Performance improvement even when few target samples available
Conclusions

Pros:

- No need to solve the source tasks
- More effective than transferring policies
- Works in any transfer scenario and with any batch RL algorithm
- Performance improvement even when few target samples available
Cons/Future works:

- *How compliance and relevance are related to performance loss?* (Define the MDP obtained by compliance/relevance transfer, measure its distance from the target MDP and bound the loss)

- *Tasks must share exactly the same state-action space* (inter-task mapping by [Taylor et al., 2007])

- *Other measures of task similarity* (e.g., [Ferns et al., 2004])

- *What about continuously changing tasks?* (Tracking changes by reusing samples [Sutton et al., 2007])
Cons/Future works:

- **How compliance and relevance are related to performance loss?** (Define the MDP obtained by compliance/relevance transfer, measure its distance from the target MDP and bound the loss)

- **Tasks must share exactly the same state-action space** (inter-task mapping by [Taylor et al., 2007])

- **Other measures of task similarity** (e.g., [Ferns et al., 2004])

- **What about continuously changing tasks?** (Tracking changes by reusing samples [Sutton et al., 2007])
Cons/Future works:

- **How compliance and relevance are related to performance loss?** (Define the MDP obtained by compliance/relevance transfer, measure its distance from the target MDP and bound the loss)

- **Tasks must share exactly the same state-action space** (inter-task mapping by [Taylor et al., 2007])

- **Other measures of task similarity** (e.g., [Ferns et al., 2004])

- **What about continuously changing tasks?** (Tracking changes by reusing samples [Sutton et al., 2007])
Cons/Future works:

- **How compliance and relevance are related to performance loss?** (Define the MDP obtained by compliance/relevance transfer, measure its distance from the target MDP and bound the loss)

- **Tasks must share exactly the same state-action space** (inter-task mapping by [Taylor et al., 2007])

- **Other measures of task similarity** (e.g., [Ferns et al., 2004])

- **What about continuously changing tasks?** (Tracking changes by reusing samples [Sutton et al., 2007])
Preliminary version of the software available at:
http://home.dei.polimi.it/lazaric/?Software

Thank you!

Any question?
Sample Relevance

**Definition**

Given a source sample $\sigma_j \in \hat{S}$, its compliance $\lambda_j$ and its average distance $d_j$ from target samples, the relevance of $\sigma_j$ is defined as

$$\rho_j = \rho(\bar{\lambda}_j, d_j) = \exp \left( - \left( \frac{\bar{\lambda}_j - 1}{d_j} \right)^2 \right),$$

where $\bar{\lambda}_j$ is the compliance normalized over all the samples in $\hat{S}$. 
Sample Relevance

![Graph showing sample relevance and distance](image)

- **Distance** $d_j$
- **Compliance** $\overline{\gamma}_j$
- **Relevance** $\rho_j$

---

*Lazaric, Restelli, Bonarini*
Bibliography I


Bibliography II

Transferring instances for model-based reinforcement learning.
*AAMAS 2008 Workshop on Adaptive Learning Agents and Multi-Agent Systems.*

Value-iteration based fitted policy iteration: Learning with a single trajectory.
*In IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning, 2007.*

Transferring state abstractions between mdps.
*In ICML Workshop on Structural Knowledge Transfer for Machine Learning, 2006.*

Value functions for RL-based behavior transfer: A comparative study.

Skill acquisition via transfer learning and advice taking.

Transfer of experience between reinforcement learning environments with progressive difficulty.
Bibliography III

Using options for knowledge transfer in reinforcement learning.
Technical report, University of Massachusetts, Amherst, MA, USA, 1999.

On the role of tracking in stationary environments.
In Proceeding of the ICML07.