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

Outline

Outline

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

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** $\langle s, a, s', r \rangle$ [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]

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

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**

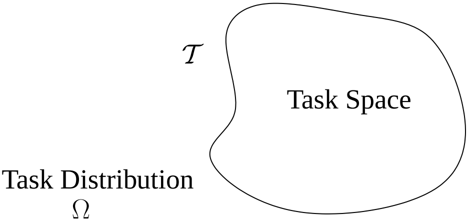
The Goal

- *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**

The Goal

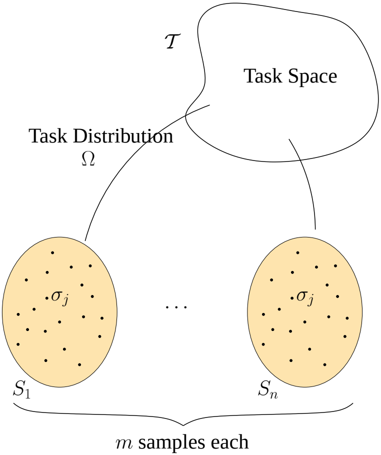
- *Fact*: In batch RL algorithms, the **set of samples** used to feed the learning algorithm influences the performance
- *Goal*: **Transfer samples** coming from other (source) tasks in order to **improve** the performance in a target task
- *Problem*: Avoid **negative transfer**

The Scenario

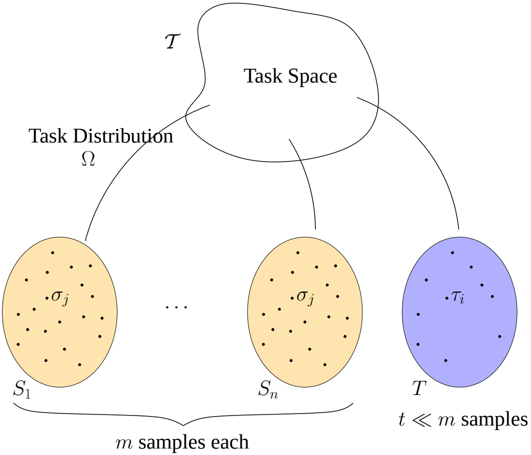


All the tasks share the same *state-action* space.

The Scenario



The Scenario



Outline

Task Compliance

- *Which tasks is it convenient to transfer from?*
- We compute the **avarage 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

$$
\begin{aligned}
P(S|\tau_i) &\propto P(\tau_i|S)P(S) \\
&= \mathcal{P}_S(s_i'|s_i,a_i)\mathcal{R}_S(r_i|s_i,a_i)P(S)
\end{aligned}
$$

Task Compliance

- *Which tasks is it convenient to transfer from?*
- We compute the **avarage 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

$$\begin{aligned} P(S|\tau_i) &\propto P(\tau_i|S)P(S) \\ &= \mathcal{P}_S(s'_i|s_i,a_i)\mathcal{R}_S(r_i|s_i,a_i)P(S) \end{aligned}$$

Task Compliance

- *Which tasks is it convenient to transfer from?*
- We compute the **avarage 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

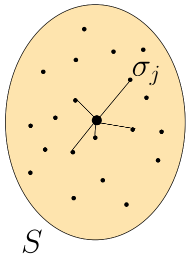
$$
\begin{aligned}
P(S|\tau_i) &\propto P(\tau_i|S)P(S) \\
&= \mathcal{P}_S(s'_i|s_i,a_i)\mathcal{R}_S(r_i|s_i,a_i)P(S)
\end{aligned}
$$

Continuous Model Approximation

- $P(\tau_i|S) = ?$
- We follow the kernel-based approximation proposed in [Jong & Stone, 2007]
- Given kernel function $\varphi(\cdot)$, $\sigma_j = \langle s_j, a_j, s'_j, r_j \rangle \in \widehat{S}$

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

with weights w_j computed according to distance in the state-action space

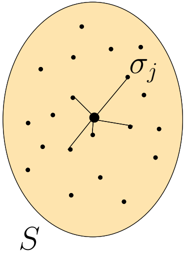


Continuous Model Approximation

- $P(\tau_i|S) =?$
- We follow the kernel-based approximation proposed in [Jong & Stone, 2007]
- Given kernel function $\varphi(\cdot)$, $\sigma_j = \langle s_j, a_j, s'_j, r_j \rangle \in \widehat{S}$

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

with weights w_j computed according to distance in the state-action space

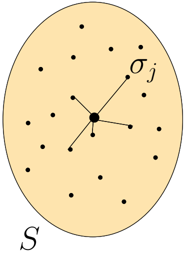


Continuous Model Approximation

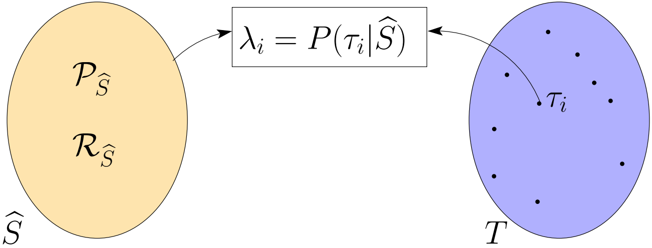
- $P(\tau_i|S) = ?$
- We follow the kernel-based approximation proposed in [Jong & Stone, 2007]
- Given kernel function $\varphi(\cdot)$, $\sigma_j = \langle s_j, a_j, s'_j, r_j \rangle \in \widehat{S}$

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

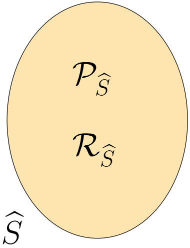
with weights w_j computed according to distance in the state-action space



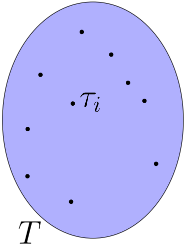
Task Compliance



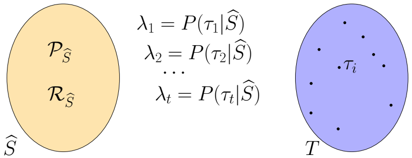
Task Compliance



$$\lambda_1 = P(\tau_1|\widehat{S})$$
$$\lambda_2 = P(\tau_2|\widehat{S})$$
$$\dots$$
$$\lambda_t = P(\tau_t|\widehat{S})$$



Task Compliance

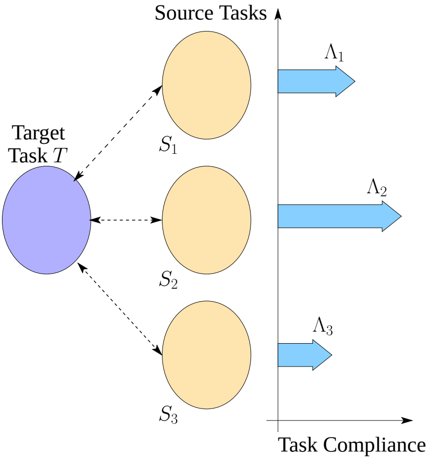


Definition

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

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

Task Compliance



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

Sample Relevance

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

Sample Relevance

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

Sample Relevance

Definition

The *relevance* of σ_j is defined as

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

Transfer σ_j whenever

- high probability to be generated by the target task (high λ_j)
- poor approximation (few samples) of $\widehat{T}$ near σ_j (high d_j)

Sample Relevance

Definition

The *relevance* of σ_j is defined as

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

Transfer σ_j whenever

- high probability to be generated by the target task (high λ_j)
- poor approximation (few samples) of $\widehat{T}$ near σ_j (high d_j)

Sample Relevance

Definition

The *relevance* of σ_j is defined as

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

Transfer σ_j whenever

- high probability to be generated by the target task (high λ_j)
- poor approximation (few samples) of $\widehat{T}$ near σ_j (high d_j)

Sample Relevance

Definition

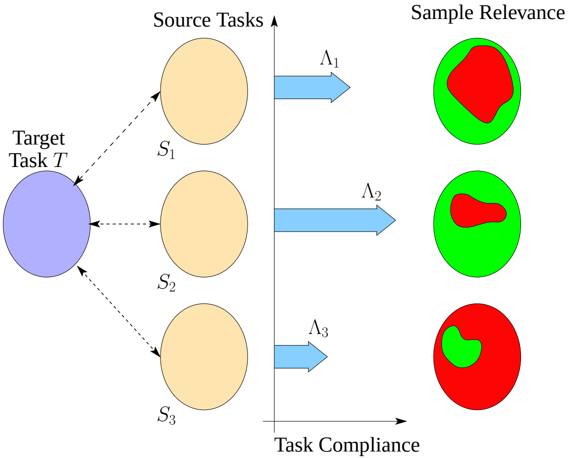
The *relevance* of σ_j is defined as

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

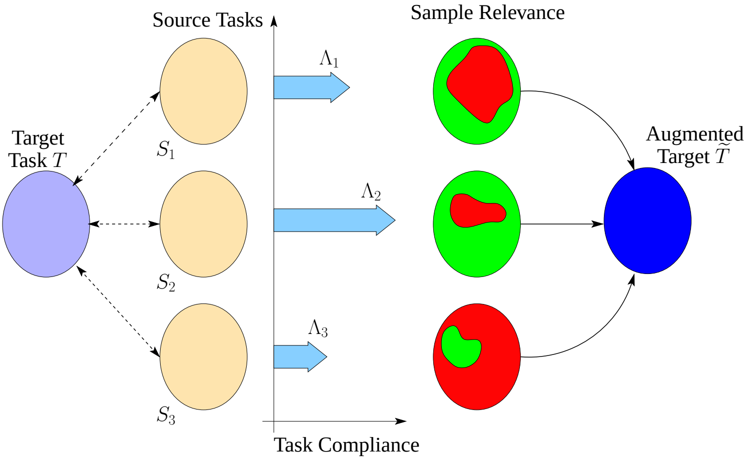
Transfer σ_j whenever

- high probability to be generated by the target task (high λ_j)
- poor approximation (few samples) of $\widehat{T}$ near σ_j (high d_j)

Sample Relevance



Transfer of Samples

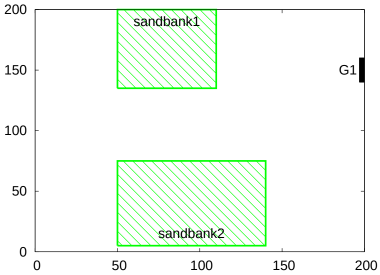


Outline

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

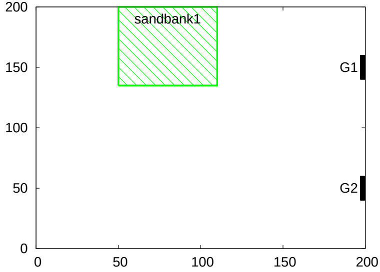
Target Task



The Boat Problem

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

Source Task S_1

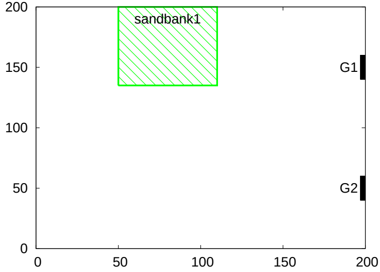


Additional goal, no *sandbank2*

The Boat Problem

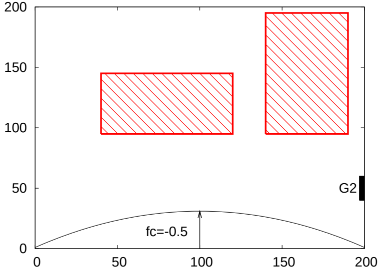
Hand-coded source tasks, see the paper for results with randomly generated tasks

Source Task S_1



Additional goal, no *sandbank2*

Source Task S_2



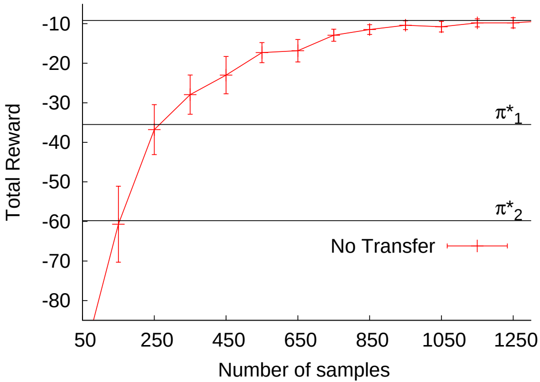
Different goal, sandbanks and current

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

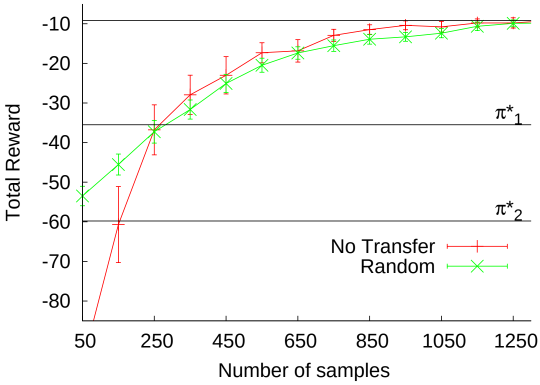
Transfer from S_1 and S_2 to T

FQI with Extra Randomized Trees



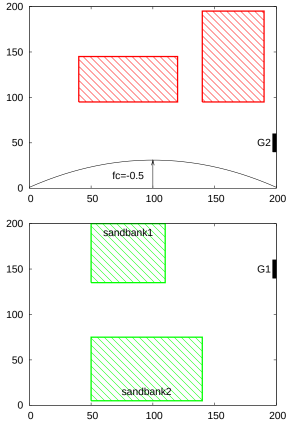
Transfer from S_1 and S_2 to T

Transfer of samples at random



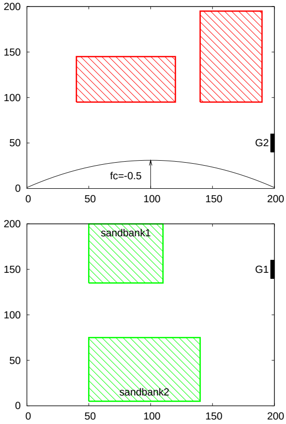
Transfer from S_1 and S_2 to T

- Most of the samples in $\widehat{S}_2$ are completely different from samples in $\widehat{T}$
- Normalized compliance $\overline{\Lambda}_1 = 0.93 \pm 0.09$, $\overline{\Lambda}_2 = 0.07 \pm 0.06$



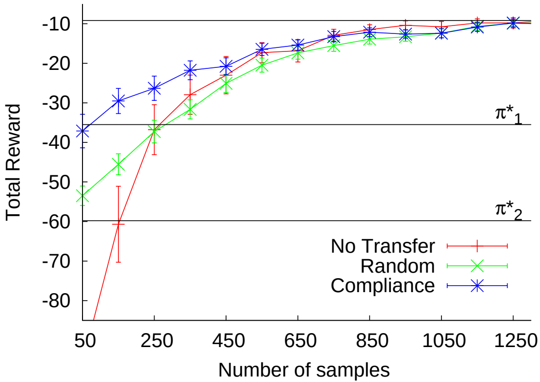
Transfer from S_1 and S_2 to T

- Most of the samples in $\widehat{S}_2$ are completely different from samples in $\widehat{T}$
- Normalized compliance $\overline{\Lambda}_1 = 0.93 \pm 0.09$, $\overline{\Lambda}_2 = 0.07 \pm 0.06$



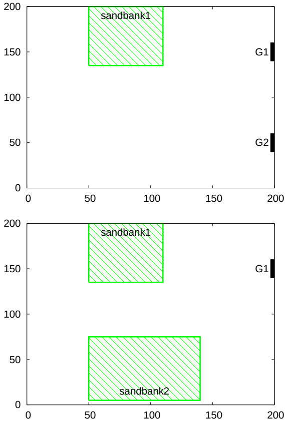
Transfer from S_1 and S_2 to T

Transfer of samples proportionally to task compliance



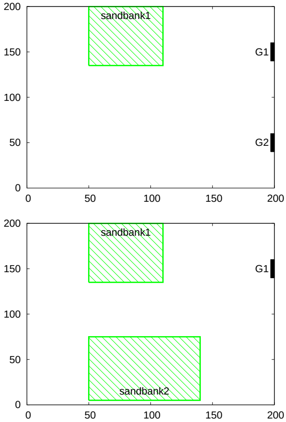
Transfer from S_1 and S_2 to T

- 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

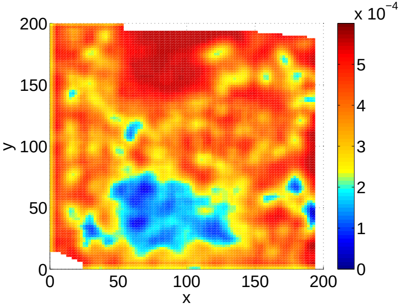
- Not all the samples from S_1 are worth transferring
- Avoid transferring samples in the region of *sandbank2* and G_2



Transfer from S_1 and S_2 to T

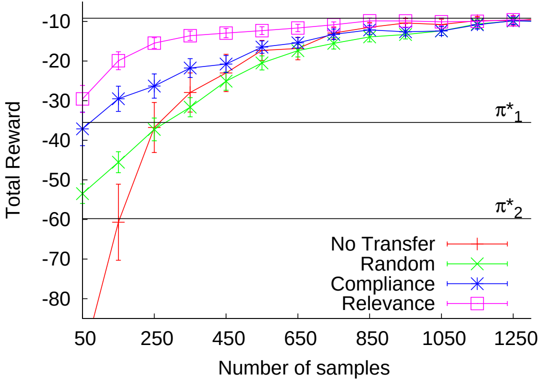
- Not all the samples from S_1 are worth transferring
- Avoid transferring samples in the region of *sandbank2* and G_2

Relevance of samples in S_1

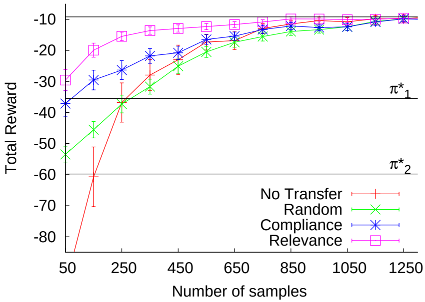


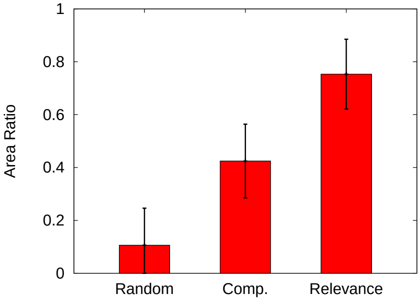
Transfer from S_1 and S_2 to T

Transfer of samples proportionally to task compliance and sample relevance



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

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

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

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

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

Conclusions

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])

Conclusions

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])

Conclusions

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])

Conclusions

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

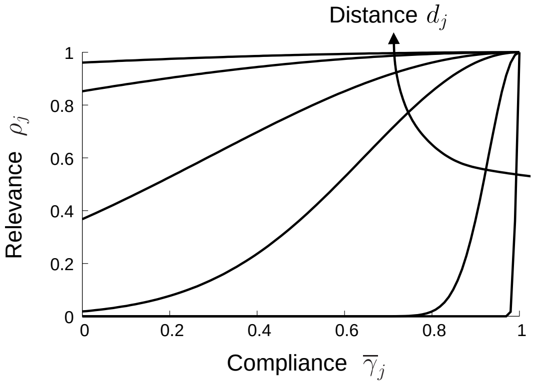
Definition

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

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

where $\overline{\lambda}_j$ is the compliance normalized over all the samples in $\widehat{S}$.

Sample Relevance



Bibliography I

Taylor, M. E., Stone, P., & Liu, Y. (2007).
Transfer learning via inter-task mappings for temporal difference learning.
Journal of Machine Learning Research, *8*, 2125–2167.

Konidaris, G., & Barto, A. G. (2007).
Building portable options: Skill transfer in reinforcement learning.
Proceedings of IJCAI (pp. 895–900).

Mehta, N., Natarajan, S., Tadepalli, P., & Fern, A. (2005).
Transfer in variable-reward hierarchical reinforcement learning.
NIPS Workshop on Inductive Transfer.

Şimşek, O., Wolfe, A. P., & Barto, A. G. (2005).
Identifying useful subgoals in reinforcement learning by local graph partitioning.
Proceedings of ICML (pp. 816–823).

Ferns, N., Panangaden, P., & Precup, D. (2004).
Metrics for finite markov decision processes.
Proceedings of UAI (pp. 162–169).

Jong, N. K., & Stone, P. (2007).
Model-based function approximation for reinforcement learning.
Proceedings of AAMAS (pp. 1–8).

Bibliography II

Taylor, M. E., Jong, N. K., & Stone, P. (2008).
Transferring instances for model-based reinforcement learning.
AAMAS 2008 Workshop on Adaptive Learning Agents and Multi-Agent Systems.

Munos R., Antos A., Szepesvari C. (2007).
Value-iteration based fitted policy iteration: Learning with a single trajectory.
In *IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning*, 2007.

Thomas J. Walsh, Lihong Li, and Michael L. Littman. (2006).
Transferring state abstractions between mdps.
In *ICML Workshop on Structural Knowledge Transfer for Machine Learning*, 2006.

Matthew E. Taylor, Peter Stone, and Yaxin Liu. (2005).
Value functions for RL-based behavior transfer: A comparative study.
In *Proceedings of the Twentieth National Conference on Artificial Intelligence*, July 2005.

Lisa Torrey, Jude W. Shavlik, Trevor Walker, and Richard Maclin. (2006).
Skill acquisition via transfer learning and advice taking.
In *ECML*, pages 425–436, 2006.

Michael G. Madden and Tom Howley. (2004).
Transfer of experience between reinforcement learning environments with progressive difficulty.
Artif. Intell. Rev., 21(3-4):375–398, 2004.

Bibliography III

T. J. Perkins and D. Precup. (1999).
Using options for knowledge transfer in reinforcement learning.
Technical report, University of Massachusetts, Amherst, MA, USA, 1999.

R. Sutton, A. Koop and D. Silver. (2007).
On the role of tracking in stationary environments.
In Proceeding of the ICML07.