TransferRL: Survey

Norio Kosaka

December 2018

1 Profile of Paper

- Title: Transfer in Reinforcement Learning: a Framework and a Survey [1]
- Author: Alessandro Lazaric
- Submitted Date: On 10 Jan 2013

2 Content of Paper

- Introduction
- A Framework and a Taxonomy for Transfer in RL
  - Transfer Framework
  - Taxonomy
- Methods for Transfer from Source to Target with a Fixed State-Action Space
  - Problem Formulation
  - Representation Transfer
  - Parameter Transfer
- Methods for Transfer across Tasks with a Fixed State-Action Space
  - Problem Formulation
  - Instance Transfer
  - Representation Transfer
  - Parameter Transfer
- Methods for Transfer from Source to target Tasks with a Different State-Action Space
  - Problem Formulation

  - Instance Transfer
  - Representation Transfer
  - Parameter Transfer
- Conclusions and Open Questions

3 Abstract

Transfer in RL is a relatively novel research area that focuses on transferring the knowledge from a set of source tasks to a target task to efficiently adapt the learning agent to it. In this present paper, they aim at providing a formalisation of the general transfer problem. First of all, I would like to define the word of *Transfer* in learning procedure. A number of psychological studies (see e.g., Thorndike and Woodworth, 1901; Perkins et al, 1992) show that humans are able to learn a task better and faster by transferring the knowledge retained from solving similar tasks. Likewise, *Transfer* in ML aims at designing the process that analyse the existing knowledge collected from a set of source tasks and transfer it so as to *bias* the learning process on a target task towards a set of pre-defined *good* hypotheses.

4 Introduction

Transfer Learning in RL: Transfer algorithms automatically build prior knowledge from the knowledge collected in solving a set of similar source tasks (i.e., *training tasks*) and use it to bias the learning process on any new task (i.e., *testing task*). And it normally results that a dramatic reduction in the number of samples for training stage and a huge improvement in the accuracy of on the target tasks.

5 A Framework and a Taxonomy for Transfer in Reinforcement Learning

5.1 Transfer Framework

We start off this section with looking at the fundamental notations regarding MDP, which is $\langle S_M, A_M, T_M, R_M \rangle$ where the state space, the action space, the transition function and the reward function, respectively. The *space of tasks* involved in the transfer learning problems is denoted by $M = \{M\}$. Let Ω be a probability distribution over the space of tasks M, finally we have the comprehensive notation representing the *environment* $\epsilon = \langle M, \Omega \rangle$, which defines the setting of the transfer problem.

A standard learning algorithms take as input some form of knowledge of the task, denoted by K, and return a solution in a set of possible results, denoted by H. In this context, K consists of all the elements used in the learning algorithm, such as instances(e.g., samples), the *representation* of the problem(e.g., set of options, set of features), and also *parameters*.Notice that K includes *prior knowledge* provided by an expert. Hence, we can renovate the generic learning algorithm in the following way;

$$A_{learn} : K \to H$$

where a_{learn} represents the learning algorithm.

Given the previous definition, we can now define the general shape of the transfer learning algorithms with additional notations. Let L be the number of tasks drawn from M according to the distribution Ω used as source tasks and K_S^L be the knowledge collected from the L source tasks and finally K_t be the knowledge available from the target task. The transfer phase will look like;

$$A_{transfer} : K_S^L x K_t \to K_{transfer} \tag{1}$$

$$A_{learn} : K_{transfer} x K_t \to H \tag{2}$$

5.2 Taxonomy

In this section, we will look at the three different aspects of the transferring in RL problem.

5.2.1 The Settings

In the following we will distinguish among three different categories of transfer setting

- Transfer from source task to target task with fixed domain: most of the early literature focused on the problem in which the domain is fixed and only two tasks are involved: a source task and a target task.
- Transfer across tasks with fixed domain: In this setting, tasks share the same domain and the algorithm takes as input the knowledge collected from a set of source tasks and use it to improve the performance in the target task.
- Transfer across tasks with different domain: tasks have a different domain, that is the y might have different state-action space.

5.3 The Knowledge

In this section, we will classify the possible knowledge transfer approaches into three major categories.

- Instance Transfer: The simplest version off transfer learning algorithm is to collect the samples(episodes) from the source task and uses them to to build a solution for the target task at hand.

- Representation transfer: After learning on different tasks, transfer algorithms often perform an abstraction process which changes the representation of the task and of the solutions, such as reward functions.
- Parameter transfer: this type of algorithm has to change and adapt the parameters used in the source task and apply them to the target task

5.4 The Objectives

In this section, we will look at the method to evaluate approaches, and classify them into three main categories.

- Learning speed improvement: it is about the reduction in the amount of the experience needed to learn the solution of the task at hand.
- Asymptotic improvement: The core concept is that the more accurate the approximation is, the better the generalisation at convergence. This objective is usually targeted by representation-transfer algorithms.
- Jumpstart improvement: This objective is about the efficient exploration and normally used by *parameter-transfer* algorithms

6 Methods for Transfer from Source to Target with a Fixed State-Action space

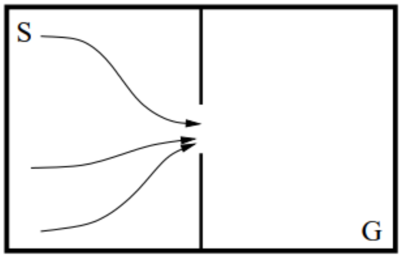
In this section, we will consider the simplest setting in which *transfer* occurs from one source task to a target task.

6.1 Problem Formulation

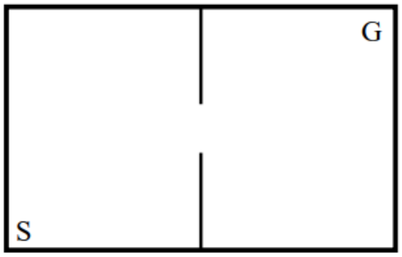
We will consider the setting where the environment consists of two MDPs, a *source task* $M_s = \langle S, A, T_s, R_s \rangle$ and a *target task* $M_t = \langle S, A, T_t, R_t \rangle$, sharing the same state-action space as follows. Hence, the environment is defined by the task space, $M = \{M_s, M_t\}$ and a task distribution Ω which simply returns M_s as the first task and M_t as second. In figure4, it describes the basic problem setting for our transfer learning environment in this present paper, and it is about the navigation problem where the agent should move from the regions marked with S to the goal region marked as G. Indeed, in this problem setting, the agent can easily find the similarity between the tasks of source and target. Then it is expected to apply the learning outcomes from the *source task* to the *target task*.

6.2 Representation Transfer

In representation transfer, it abstracts characteristic of the task from the source task. Then it changes the representation collected from the source task either



source task



target task

of the solution space H or of the MDP so as to speed-up the learning process in the target task. So, in this section, I would like to describe three different methods as follows;

- Option Discovery: is to change the representation of the MDP by adding *options* to the set of available actions.(Sutton et al., 1999 [7]) The core idea is to exploit the structure of the dynamics shared by the two tasks and to ignore the details in the specific source reward function. There are other approaches as well, [4] [5]
- Action Space Transfer: using random task perturbation, a set of task is artificially generated from one single source task and the actions which end up with being not optimal are replaced with the generated actions. Then with the set of renewed actions, learning agent begins working on the target task. The advantage of this approach is that the reduction of unnecessary actions enhance the learning speed.(Sherstov and Stone, 2005[6]).
- Feature Discovery: this is sort of an advanced version of *option discovery* in terms of exploring the similarity in characteristic between the source and target tasks. However, the biggest difference is that it extracts a set of features which defines the environment, instead of options and it is transferred to the target task from the one of source. Indeed, it does differ in terms of the purpose of the approach, for instance, *Feature discovery* aims at achieving better approximation of the target value function. One of the approaches is that Mahadevan and Maggioni (2007)[3] has proposed a method to generate *proto-value functions* using spectral analysis of the *Laplacian* of the estimated graph of the source MDP.

6.3 Parameter transfer

Previously stated approaches implicitly assumed the similarity of source and target tasks. Ferns et al (2004) and Phillips (2006) have examined the option-transfer approach to see if it can accommodate *partially similar* target task.

And they have come up with the metric to calculate the similarity of the environmental setting of the tasks, which is denoted as follows:

$$d(s) = max_{a \in A}(|R_s(s,a) - R_t(s,a)| + \gamma J(d)(T_s(\cdot|s,a), T_t(\cdot|s,a))) \qquad (3)$$

where $J(d)$ is the Kantorovish distance which measures the difference between the two transition distributions.

7 Methods for Transfer across Tasks with a Fixed State-Action Space

In previous section, we have considered the setting in which only one source task is available, in contrast, in this section we will review the main transfer approaches to the general setting when a set of source tasks is available. In this case, there are two things mattering.

1. How to merge knowledge coming from different sources
2. How to avoid the transfer from sources which differ too much from the target task

7.1 Problem Formulation

In this section, we consider the setting in which the environment is defined by a set of tasks and a distribution. In fact, we can still assume that all the tasks share the same state-action space. But the reward functions do differ among tasks.

7.2 Instance Transfer

The core idea is that the transfer of source samples might improve the learning on the target task. For running example, we would like to consider the proposal of Lazaric et al (2008) [2] which selectively transfers samples on the basis of the similarity between source and target tasks. As opposed to review earlier, this algorithm selects which source samples should be included in transfer batch based on a measure of similarity between the source and the target tasks. The number of source samples is assumed to be large enough to build an accurate kernel-based estimation of each source model. In the experiments reported in their research, this method is shown to successfully identify which sources are more relevant to transfer samples from and to avoid *negative transfer*.

7.3 Representation Transfer

Unlike in the source-to-target transfer, the objective of representation transfer in this setting is to infer from the source tasks general characteristics that are preserved across the tasks and that can be effectively exploited to improve the average learning performance with respect to single-task learning.

- Option Transfer: . Bernstein (1999) introduces a method to create a new option which is added to the set of available actions and which is built as a combination of the optimal policies learned on the L source tasks and it is then reused to speed-up learning process on the target task.
- Feature transfer to speed-up learning: it is to identify a function space which is likely to contain functions able either to speed-up the learning or to accurately approximate the optimal value functions. One particular method proposed earlier is to identify sub tasks and features according to the analysis of the dynamics of the source tasks, and solve them to find the optimal value-function.
- Feature transfer to improve asymptotic performance: While previously stated methods above are assuming the discrete MDPs, now we would like to consider the continuous state-action space in which function approximation is necessary. The objective is to learn the parameters which defines a feature space such that only a small set of features are used to approximate the value functions of the tasks.

7.4 Parameter Transfer

In contrast to representation-transfer approaches, all parameter-transfer algorithms explicitly define a distribution on the task space and try to estimate the true distribution in order to build a prior over the space of hypotheses so as to improve the initial performance and reduce the number of samples needed to solve any task in the distribution.

- The hierarchical Bayesian Model: Using the representation of HBM, we can formulate the problem in the equation and solve it using *Bayes Rule*.
- Inference for transfer: There is a simpler approach, which only uses the statistics about the action values. The mean and variance of the action values over different tasks are computed and then used to initialise the Q-table for new tasks.

8 Methods for Transfer from Source to Target Tasks with a Different State-Action Spaces

All the previous settings consider the case where all the tasks share the same *domain*. In the most general transfer setting, however, the tasks may also differ in terms of number or range of the state-action variables.

8.1 Problem Formulation

In this section, we consider the case where the source task and target task have different MDPs respectively. And the approaches here can be summarised in three categories.

- A hand-coded mapping
- learn a mapping from source to target
- extract from source task MDP some *abstract knowledge* that can be reused to solve the target task

And in this section, we would like to assume the *mountain car domain* game on our mind.

8.2 Instance Transfer

Taylor et al (2008a) study the transfer of samples by considering the case where there exists only one source task and no specific method for the selection of the samples is implemented.

8.3 Representation Transfer

A number of transfer algorithms deal with the problem, which is that options learned on the source task are defined as a mapping from specific state to action space and they cannot be used in a target task with different state-action variables, by considering abstract options that can be reused in different tasks. For instance, Ravindran and Barto (2003); Soni and Singh (2006) use the homomorphism framework to map tasks to a common abstract level.

8.4 Parameter Transfer

Most of the algorithms in this approach are based on the hand-coded mappings and investigate how the transfer of different sources of knowledge influence the performance on the target tasks.

9 Conclusions and Open Questions

Although it has been long time to investigate the junction of *transfer learning* and reinforcement learning, there are still a lot of things to be clear in the future. And in this section, I would like to briefly list the open questions for further work.

- Theoretical analysis of transfer algorithms: although experimental results have already showed the effectiveness of transfer learning in RL, recent research has explored the improvement of the performance in the multi-task learning the supervised learning paradigm under certain circumstances.
- Transfer learning for exploration: study of optimal exploration, trade-off between exploitation and exploration.

- Concept drift and continual learning: One of the fundamental assumptions of transfer learning is that a clear distinction between the tasks in M is possible. However, some recent researches has explored the case where the tasks vary over the time, such as non-stationarity of the environment.

References

[1] Alessandro Lazaric. "Transfer in reinforcement learning: a framework and a survey". In: *Reinforcement Learning*. Springer, 2012, pp. 143–173.

[2] Alessandro Lazaric, Marcello Restelli, and Andrea Bonarini. "Transfer of samples in batch reinforcement learning". In: *Proceedings of the 25th international conference on Machine learning*. ACM. 2008, pp. 544–551.

[3] Sridhar Mahadevan and Mauro Maggioni. "Proto-value functions: A Laplacian framework for learning representation and control in Markov decision processes". In: *Journal of Machine Learning Research* 8.Oct (2007), pp. 2169–2231.

[4] Amy McGovern and Andrew G Barto. "Automatic discovery of subgoals in reinforcement learning using diverse density". In: (2001).

[5] Ishai Menache, Shie Mannor, and Nahum Shimkin. "Q-cut—dynamic discovery of sub-goals in reinforcement learning". In: *European Conference on Machine Learning*. Springer. 2002, pp. 295–306.

[6] Alexander A Sherstov and Peter Stone. "Improving action selection in MDP's via knowledge transfer". In: *AAAI*. Vol. 5. 2005, pp. 1024–1029.

[7] Richard S Sutton, Doina Precup, and Satinder Singh. "Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning". In: *Artificial intelligence* 112.1-2 (1999), pp. 181–211.