Budgeted Knowledge Transfer for State-Wise Heterogeneous RL Agents

Farbod Farshidian, Zeinab Talebpour, and Majid Nili Ahmadabadi

Cognitive Robotics Lab., School of Electrical and Computer Engineering,
University College of Engineering, University of Tehran
{f.farshidian,z.talebpour,mnili}@ut.ac.ir

Abstract. In this paper we introduce a budgeted knowledge transfer algorithm for non-homogeneous reinforcement learning agents. Here the source and the target agents are completely identical except in their state representations. The algorithm uses functional space (Q-value space) as the transfer-learning media. In this method, the target agent's functional points (Q-values) are estimated in an automatically selected lower-dimension subspace in order to accelerate knowledge transfer. The target agent searches that subspace using an exploration policy and selects actions accordingly during the period of its knowledge transfer in order to facilitate gaining an appropriate estimate of its Q-table. We show both analytically and empirically that this method decreases the required learning budget for the target agent.

Keywords: Reinforcement Learning, Knowledge Transfer, Dimension Reduction, Exploration Policy.

1 Introduction

In many problems where a Reinforcement Learning (RL) agent is trying to learn the optimal solution given a limited budget (in terms of time or other factors such as number of failures), the main challenge is to reduce the number of learning trials. The agent may highly expedite its learning process if, similar to human-beings and even some animals, it can use the knowledge and experiences of other agents. This process is called knowledge transfer (KT) and takes place in different forms. Here we focus on direct KT where the source agent provides the target agent with its solution to the problem. By solution in RL agents we mean the learned Q-tables.

The main goal of KT is to make use of the knowledge acquired in a set of source tasks to improve the performance in a related target task which has not previously been experienced by the agent. Transfer learning is a challenging and relatively new area in the field of RL [1-2]. The main challenge of transfer learning is reducing the information about the relationship between the source and the target task which is given to the learner.

In the case of homogeneous agents, transfer learning can be directly applied between agents but when agents are heterogeneous this transfer of knowledge can only take place when a correspondence between the source and target agents is known in

T. Huang et al. (Eds.): ICONIP 2012, Part I, LNCS 7663, pp. 439–446, 2012.

terms of some common property. Difference in environmental dynamics, state or action spaces and reward functions can cause heterogeneity between agents [3]. In this article we focus on heterogeneous agents with different perceptions of the environment, this means that the agents perceive their environment with different features while other aspects of the world and the agents remain the same.

To overcome the heterogeneity problem, we need to find some common grounds between agents. This is referred to as the mapping problem in Torrey and Shavlik [2] which states that a mapping to translate the properties of the source agent to the target agent is needed in order to enable the target agent to use the knowledge acquired by the source agent. Many transfer approaches assume that human beings provide such information [4-6]. Finding several possible mappings among the properties of agents and allowing the target agent to experiment with them all, is another approach used by Soni and Singh [7], Mihalkova et al.]8], Taylor et al. [9], and Taylor et al. [10]. Soni and Singh [7] limit the mappings by considering object types and also avoiding separate evaluation of each mapping by the use of options. In the work of Mihalkova et al.]8] the search is limited to Markov logic networks, requiring that mapped predicates have matching arity and argument types. Taylor et al. [9] use a classification method to find the mapping between the tasks for state space and action space, assuming that some information on state variable grouping is provided by a human. In another work of Taylor et al. [10], they try to find the mapping between the state space and the action space through building a transition model based on samples from the target task and searching for a mapping between the tasks which minimize the prediction error criterion for the data from the source task. Another transfer learning approach is to define domains in such a way that the agents become homogeneous. Relational learning can be used to construct domains with this property [11-12].

In this work we are concerned with finding a method to increase the speed of learning in the target agent—given a limited budget—using the knowledge of an agent which has already learned the task. For this purpose we have chosen to work in a space in which proximity conveys having the same effect when applied to a task and therefore the difference in state representation is no longer a problem since despite this difference, points with similar functionalities will be positioned close together. We propose a method which finds an appropriate exploration policy for the target agent during its use of the transferred knowledge, based on space reduction of this space. This exploration policy can significantly reduce the learning trials needed to achieve a specified performance as shown in the experiments.

2 Proposed Method

In this paper, we are supposed to solve a KT problem in which the source and the target tasks differ just in the state representing features. It is assumed that there is a one-to-one mapping between the states of the source and the target agents but this correspondence mapping between the states of these agents is unknown.

Our solution to this KT problem is to find state-correspondence mapping between source and target agents. The proposed algorithm has two main attributes. The first is the use of functional space which embodies some task-invariant knowledge and the other is the use of a particular exploration policy which reduces the number of trials needed to find correspondence in the functional space.

The Functional Space—for a specific task—is a space in which neighboring is defined as being equally rewarding when applied to the task. Each state of the task corresponds to a point in Functional Space called a *functional point* (FP). While abstraction in the perceptual space does not result in state-independent abstraction, abstraction in the FS does. In this paper the mapping function between states and FPs is chosen to be the Q-values of the states which only depend upon the action set of the agents, with no dependency upon the states representing features. Heterogeneity in the sense of states' features difference is of no significance in this space because each state is represented by a vector of its Q-values and is only dependant on the values of actions in that state. This is a common ground for agents with different state representing features. Here the FS is a continuous *n*-dimensional space where *n* is the size of the agent's set of actions and represented by a functional vector containing the Q-values of a state. As an example $Q(s_i)=\left[Q(s_i,a_1),Q(s_i,a_2),...,Q(s_i,a_n)\right]^T$ is a functional point to which s_i is mapped, and *i* belongs to the set $\{1,2,...,NumberOfStates\}$.

In this algorithm, we consider a spherical region around each Q-vales of the source agent as the neighboring area. If the target agent's Q-value located in this area during the updating process, it is considered that those states are in an exact correspondence. It is important to notice that the Q-values of the target agent are not available and should be estimated through interaction with the environment. In our algorithm, we have proposed an exploration policy in which instead of evaluating Q-values (*Q(s)*) in the original functional space, we estimate its projection onto a lower-dimension subspace. Updating Q(s) in this lower-dimension subspace rather than whole FS reduces the number of required samples which accelerates finding the correspondence between the agents' states.

In order to achieve a subspace of FS which still makes appropriate discrimination between the functional points (From now then we refer to the source agent's Q-values as functional points), we employed Linear Discriminant Analysis (LDA). LDA method is one of the common techniques used for reducing the dimensionality of data. This method selects directions in the original space which the largest amount of discrimination between classes is obtainable. As mentioned a spherical neighboring area is defined around each FPs, the radius of this area is a user defined parameter. After choosing the Neighborhood Radius (NR), the dimension of the reduced subspace in the LDA-obtained space is determined in a way that the neighborhood regions of different functional points do not overlap with each other. By utilizing LDA on the points in the functional space and computing the dimension of the reduced subspace, we obtain the projection of the FS in the lower-dimension space along with the transfer matrix of this subspace.

To evaluate *Q(s)* in the prescribed subspace of the functional space, we have designed an exploration policy which provides informative samples for updating Q(s) in the reduced FS exclusively.

In order to formulate this idea, we assume the Q-values to be random processes in an n-dimensional space where *n* is dimensionality of the FS (equal to the cardinality of the action set). This random process is updated according to a learning algorithm such as Monte Carlo method or Q-Learning method, *Q(s)* is a non-stationary random process in which as the number of updates increases the first moment of it approaches to the optimal value of *Q* while the higher-order central moments approach to zero thus the probability density of $Q_t(s,a)$ moves toward a Dirac delta function. Now we want to determine the exploration policy which causes only the probability density of the projected Q-values in the specified subspace to approach the Dirac delta function. For this reason we need a measure for approaching to the Dirac delta function and we have chosen the trace of covariance matrix as our measure for this purpose.

The transfer matrix to the lower subspace is called *T* which is a $m \times n$ matrix where *m* is the dimensionality of the specified subspace. The covariance matrix of the projected Q-value ($\overline{Q_t}(s)$) can be written as Equation (1).

$$Cov\left[\overline{Q_t}(s)\right] = TCov\left[Q_t(s)\right]T^T \qquad (1)$$

Using Equation (1) beside the property of the matrix trace operator in which we have $trace(ABC) = trace(CAB)$ if the multiplication of CAB exists and also considering that the chosen policy affects the number of updates of a *(s,a$_i$)* pair, our problem is transferred to the optimization problem in Equation (2).

$$\underset{t_1,t_2,\ldots,t_n}{Min}\ trace\left(T^T TCov\left[Q_t(s)\right]\right) \qquad s.t.\ \sum_i t_i = t_{total} \qquad (2)$$

Let us assume the covariance matrix of the Q can be written in the form $Cov\left[Q_t(s)\right] = diag\left[c/f(t_1), c/f(t_2), \ldots, c/f(t_n)\right]$. Here $f(t_i)$ is an increasing function of the number of updates on *(s,a$_i$)* pair. This assumption holds for the Monte Carlo method but it is not proven to be completely true for other methods such as Q-Learning. In the end of this section we discuss this assumption in more detail. By this assumption and using Lagrange multiplier, we have Equation (3).

$$\begin{aligned} & t_i = F^{-1}(cA_i/\lambda), && for\ i = 1,2,\ldots,n \\ & \sum_{i=1}^{n} F^{-1}(cA_i/\lambda) = t_{total} && \end{aligned} \qquad (3)$$

Where $[A_i] = diag\left(T^T T\right)$ and $F(t) = \left(f(t)\right)^2 / \left(\frac{df(t)}{dt}\right)$. $F^{-1}(t)$ is the inverse function of $F(t)$. In the case of Monte Carlo learning *f(t) = t*. So we have Equation (4).

$$t_i = \frac{\sqrt{A_i}}{\sum_i \sqrt{A_i}} t_{total} \qquad for\ i = 1,2,\ldots,n \qquad (4)$$

As you can see *c* does not appear in the computation of t_i. In terms of probability of selection or exploration policy, we can write Equation (5).

$$\pi(a_i \mid s) = \frac{\sqrt{A_i}}{\sum_i \sqrt{A_i}} \qquad for\ i = 1,2,\ldots,n \qquad (5)$$

In our algorithm, we aim to find the optimal Q-value which corresponds to the optimal policy. Therefore we should utilize an off-policy learning method such as off-policy Monte-Carlo or Q-Learning. In the literature, it is discussed that the off-policy Monte Carlo has a slow convergence rate and other methods such as Q-learning are suggested. Based on this consideration, we use Q-Learning in our implementations as the learning method. So far we have derived the relationship between the specified subspace and the exploration policy for the Monte Carlo learning method. Deriving a formulation like Equation (5) for Q-Learning is a complicated task since $C\text{ov}[Q_t(s)]$ not only is not diagonal but also depends on other states' number of updates. Since the convergence rate of Q-Learning is usually faster than that of Monte Carlo method in many problems, we expect its covariance-matrix trace to be smaller than the Monte Carlo method's although, to the best of our knowledge, there is not any analytical proof for this claim. However, if we assume this statement to be correct, we can assert that by choosing the exploration policy according to Equation (5) we minimize the upper bound of our criterion for projected Q-value estimations.

As the learning process proceeds the states' Q-values approach to the vicinity of their corresponding FPs. Also since the designed exploration policy tends to update the Q-values only in the specified subspace of the FS, the states' Q-values will more rapidly converge in that subspace comparing to the Q-values in the orthogonal subspace. In our algorithm, a state will be attributed to a FP, if it is located in the NR-radius neighborhood of a FP for a number of successive updating iterations in the specified subspace. As soon as a state is attributed to a FP its value is altered to the value of that FP. This strategy of altering the Q-values to the value of the corresponding FPs will cause an immediate change in the Q-value in the orthogonal subspace which needs to be propagated through other state-action pairs. Therefore we apply an updating method which is very similar to that of the model-based reinforcement learning. Here we use a simple version of this method; however using more advanced methods such as prioritized sweeping [13] will be more promising in complicated tasks.

3 Evaluation and Results

For the purpose of evaluating the method and analyzing its main properties, we have conducted a series of tests in a multi-arm bandit benchmark which has many emerging applications including in clinical trials, resource allocation, and adaptive routing. The experimental set-up is simple; nevertheless it is selected to reveal the positive or indifferent aspects of this approach clearly and the task is chosen in such a way that its complexity wouldn't shadow the approach itself.

There is a set of 10 slot machines called bandits with 10 arms per each machine. Every arm when pulled yields a pay-off from an unknown but fixed distribution. The goal of this problem is to maximize the expected cumulative discounted pay-off.

In our experiments we consider every bandit to have arms conforming to a Gaussian distribution with variance 5 and mean values which are extracted from a normal distribution with specific variance in Table 1 and randomly chosen means on the bisector of a 4 by 4 hypercube centered at the origin. These bandits form the state space and their arms form the action space of our RL problem. In all experiments the mean values of the distributions have been randomly chosen. Every experiment is performed 1000 times where the mean values for the bandit arms changing according to the specified distribution and the average results are reported. The following table contains the value of other parameters used in the experiments.

Table 1. Experimental setting parameters

Experiment	Variance	Discount Factor (γ)	Subspace Dimension
1	0.25	0	User-defined
2	0.09	0.9	automatically

3.1 Experiment 1

This experiment is designed to reveal the effect of dimension reduction on the convergence speed of the algorithm. In this setting an agent using normal Q-learning method needs a much more precise estimation of the Q-values, in order to achieve the optimal performance since most FPs are locates near the bisector.

The expected accumulated reward during an episode where agents make greedy decisions based on the Q-tables they have obtained so far is the metric used for evaluation of the performance of this algorithm. Then, the average number of episodes needed to reach a Q-table which causes 90% and 95% of the optimal performance is obtained and shown in Table 2 for different subspace reductions. As Table 2 shows that by reducing the dimension of the exploration subspace the performance of the KT algorithm is increased constantly except dimension 1. These results confirm that reducing the exploration subspace is quite advantageous.

Table 2. Number of episodes needed to reach the given criteria

Reduced Dimension	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Normal Learning**
90%	126	136	158	185	217	245	257	302	314	326	319
95%	268	213	233	262	316	353	393	436	471	506	675

3.2 Experiment 2

Fig. 1.a depicts the ratio of accumulated reward to the maximum possible accumulated reward for every episode; we refer to this as the normalized expected accumulated reward in the figure. The results are accomplished by averaging over different arrangement of bandits. As mentioned earlier, the accumulated reward in each episode is the expected accumulated reward during an episode where agents make greedy decisions based on their latest estimation of the Q-tables. This is similar to the situation where agent has a finite budget for learning and at the end of this budget the agent is asked to act according to a greedy policy based on its estimation of the Q-values so far. Here the agent using normal Q-learning needs 794 episodes to reach 90% of the optimal solution whereas the proposed method with the average reduced dimensionality of 8.1 needs only 708 episodes which is making an improvement of about 11%. Also for reaching 95% of the optimal solution, the normal learner needs

1608 episodes while the proposed algorithm needs only 1103 episodes which is 31.0% improvement. This figure also shows an evident improvement in the asymptotic behavior of learning trend when using the proposed method.

In Fig. 1.b, the percentile of error in finding the correspondence between the source and target agents' states during the KT period is shown. This graph is yielded by averaging between 1000 aforementioned bandit sets. In this figure, we see that around episode 2300 the mapping error is converged to zero which is equivalent to the time when the KT learner reaches the maximum possible accumulative reward in figure 1.a. Also we see that the target agent can find 50% of state-wise mapping up to episode 1000. This is also around time when knowledge-transfer learner boosts its performance comparing to the normal learner in figure 1.a.

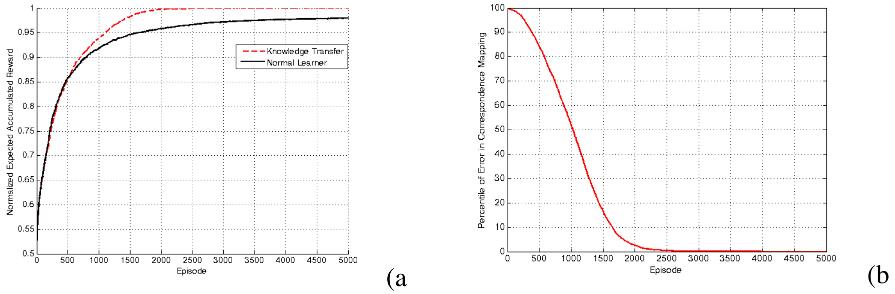

Fig. 1. a) The normalized accumulated reward for each episode. Here dash curve depicts the performance of the KT method and the bold curve depicts the performance of the Normal Reinforcement Learning method. **b)** Percentile of error in finding correspondence between the source and target agents during KT, averaged from 1000 different bandit sets

4 Conclusion and Discussion

This paper tackles the problem of KT between two agents performing the same task with different state representation. Our proposed method uses functional space as the means of facilitation for transferring knowledge. Rather than estimating the target agents' functional points, we tried to estimate the functional points in a lower-dimension subspace. Following this notion, we suggested a method for determining the exploration policy. This policy tries to decrease the covariance of functional points' error on the specified subspace faster. The proposed algorithm demonstrated superior performance in terms of learning speed with limited learning budget.

One critical issue in our proposed algorithm is the absence of a double checking procedure in the last step of the algorithm when the target agent's states are mapped to the source agent's states. Any mistake in this step will not be compensated for. A possible remedy for this issue is that, the target agent initializes its Q-table using the proposed method and then continues learning: see the *optional* step in Algorithm 1 Automatic adjustment of the algorithms' parameters along with testing the method on more sophisticated tasks is the next steps of this research.

References

1. Taylor, M.E., Stone, P.: Transfer learning for reinforcement learning domains: A survey. The Journal of Machine Learning Research 10, 1633–1685 (2009)
2. Torrey, L., Shavlik, J.: Transfer learning. In: Handbook of Research on Machine Learning Applications, vol. 3, pp. 17–35. IGI Global (2009)
3. Lazaric: Knowledge transfer in reinforcement learning. PhD thesis, PhD thesis, Politecnico di Milano (2008)
4. Tanaka, F., Yamamura, M.: Multitask reinforcement learning on the distribution ofMDPs. In: Proceedings. 2003 IEEE International Symposium on Computational Intelligence in Robotics and Automation, vol. 3, pp. 1108–1113 (2003)
5. Taylor, M.E., Stone, P., Liu, Y.: Value functions for RL-based behavior transfer: Acomparative study. In: Proceedings of the National Conference on Artificial Intelligence, vol. 20, p. 880 (2005)
6. Wilson, A., Fern, A., Ray, S., Tadepalli, P.: Multi-task reinforcement learning: a hierarchical bayesian approach. In: Proceedings of the 24th International Conference on Machine learning, pp. 1015–1022 (2007)
7. Soni, V., Singh, S.: Using homeomorphisms to transfer options across continuous reinforcement learning domains. In: Proceedings of the National Conference on Artificial Inligence, vol. 21, p. 494 (2006)
8. Mihalkova, L., Huynh, T., Mooney, R.J.: Mapping and revising Markov logic networksfor transfer learning. In: Proceedings of the National Conference on Artificial Intelligence, vol. 22, p. 608 (2007)
9. Taylor, M.E., Whiteson, S., Stone, P.: Transfer via inter-task mappings in policy search reinforcement learning. In: Proceedings of the 6th International Joint Conference on Autonomous Agents and Multi-agent Systems, p. 37 (2007)
10. Taylor, M.E., Jong, N.K., Stone, P.: Transferring Instances for Model-Based Reinforcement Learning. In: Daelemans, W., Goethals, B., Morik, K. (eds.) ECML PKDD 2008, Part II. LNCS (LNAI), vol. 5212, pp. 488–505. Springer, Heidelberg (2008)
11. Driessens, K., Ramon, J., Croonenborghs, T.: Transfer learning for reinforcement learning through goal and policy parameterization. In: Proceedings of the ICML Workshop on Structural Knowledge Transfer for Machine Learning (Online Proceedings), p. 14 (2006)
12. Pan, S.J., Kwok, J.T., Yang, Q.: Transfer learning via dimensionality reduction. In: Proceedings of the 23rd National Conference on Artificial Intelligence, vol. 2, pp. 677–682 (2008)
13. Moore, A.W., Atkeson, C.G.: Prioritized sweeping: Reinforcement learning with lessdata and less time. Machine Learning 13(1), 103–130 (1993)