Partial Transfer Learning: Using Sub-Goals to Improve the Learning Rate

Barış Gökçe sozbilir@boun.edu.tr
Boğaziçi University, Department of Computer Engineering, Bebek, Istanbul, TURKEY

H. Levent Akın akin@boun.edu.tr
Boğaziçi University, Department of Computer Engineering, Bebek, Istanbul, TURKEY

Abstract

Although Reinforcement Learning (RL) is one of the most popular learning methods, it suffers from the curse of dimensionality. As the domains of the state and the action sets increase, the learning rate of the agent decreases dramatically and a solution for the problem may not be learned at all. In order to manage the effects of the curse of the dimensionality, researchers typically concentrate on methods that reduce the complexity of the problem. While some of them do this by using hiearchical representations, others try to transfer knowledge obtained during the learning process of simpler tasks. In a transfer, the source and the target tasks may have conflicting requirements as well as common requirements. While learning from scratch ignores the common requirements, transferring all knowledge ignores the conflicting requirements. In this study, we aim to improve the learning rate of an agent by transferring only the common requirements. The main contribution is modeling the problem in a hierarchical manner to transfer only the relevant knowledge in a given setting. The proposed method is tested on three robot navigation tasks in room-based environments and is shown to make a dramatic increase in the learning rate compared to previous approaches.

1. Introduction

Learning is one of the main components of the intelligence. Piaget (Piaget, 1950) has introduced the developmental structure of intelligence as the *Theory of Cognitive Development*. We can define the developmental structure of the learning process as learning new skills by making use of previously learned skills with simpler tasks. For example, a baby learns how to get up before learning how to walk. The learning process of a baby continues its development with the acquisition of the running skill. Therefore, an agent should learn some basic skills to be capable of learning more complex tasks.

Reinforcement Learning (RL) is a powerful learning methodology for an artificially intelligent agent. Although RL is remarkably effective on the problems where the agent cannot define the goodness of each action in each state, it suffers from the curse of dimensionality. In other words, as the domains of the state and the action sets increase, the learning rate of the agent decreases dramatically and eventually the solution for the problem cannot be learned. Researchers propose two approaches to address this problem; *Hierarchical Reinforcement Learning* (based on defining the original task as a combination of sub-goals which are easier to learn) and *Transfer Learning* (reducing the complexity by transferring the knowledge obtained as a result of previous experiments).

In this work, we mainly concentrate on the way of transferring only the necessary knowledge from source task to target task. In general, the source task and the target task may have both common and conflicting sub-goals. We propose to use the sub-goals while transferring the experiences acquired as a result of learning source tasks. Transferring the information about the sub-goals of simpler tasks reduces the complexity of a task to a simpler one. By that way, we can

Proceedings of the 30th International Conference on Machine Learning, Atlanta, Georgia, USA, 2013. JMLR: W&CP volume 28.

prevent the agent from getting misleading knowledge, while transferring the common goals in previous tasks (source task) to current tasks (target task) leading to a significant increase in the learning rate.

The organization of the rest of the paper is as follows. In the next section, we will give a brief discussion about the state of the art for *HRL* and *Transfer Learning*. The proposed algorithm is explained in Section *Partial Transfer Learning*. In section *Experimental Results*, performance of the proposed method is tested on a robot navigation problem. Future directions are listed in Section *Conclusion*.

2. Related Work

The hierarchical reinforcement learning (HRL) and transfer learning (TL) are popular approaches to improve the learning rate of the reinforcement learning. While the first approach makes temporal abstraction over actions, other one aims to retrieve useful information from the experiences gained in previous learning processes.

HRL approach represents the learning problem in a layered-manner and typically, the sub-goals in this setting are the nodes in a hierarchical structure. So, the bottleneck to construct the hierarchy is finding the sub-goal states. We can classify the main approaches for discovering the sub-goals autonomously as *graph-based* and *metric-based* methods.

In (Hengst, 2002), Hengst defines the taxi problem as a *Markov Decision Process* (MDP) and divides its state space into nested sub-MDP regions. In order to make this division, he models the problem as a directed graph where the nodes are the states and the edges are the transitions and it is decomposed into strongly connected components to represent the sub-MDPs. The transitions with non-stationary probability distribution are the sub-goals. Due to the time complexity of the algorithm, it is not feasible to apply in real-time. Kazemitabar and Beigy introduced a linear time complexity version of this algorithm in (Kazemitabar & Beigy, 2009a), and (Kazemitabar & Beigy, 2009b) by the adjacency list representation of the transition graph instead of the adjacency matrix.

In another graph-based approach (Menache et al., 2002), Menache defines the problem as a *max-flow/min-cut* problem as described in (Elias et al., 2002). The *Preflow/Push* algorithm (Goldberg & Tarjan, 1988) is used to solve the *max-flow/min-cut* problem. Although the algorithm finds the bottleneck states as the sub-goals, the complexity of the algorithm is $O(n^3)$.

In (Ö. Şimşek and A.P. Wolfe and A.G. Barto, 2005), the authors partition the local transition graph. First, the local state-transition graph is constructed by the recent experiences, (i.e. trajectories). While constructing the graph, the states in the trajectory are defined as the nodes, transitions are the edges and frequency of the corresponding transitions determines the weight of the edges. A cut point is determined as the transition from one region to another in one step that has a low but strictly positive probability and the end points of such transitions are the sub-goals. Since the graph is constructed from the local trajectories, it does not use the entire state space which in turn improves the performance of the method. The probability of transformation from region A to region B is measured by the *Normalized Cut (NCut)* metric, as in Equation 1, where *cutsize(A, B)* is the sum of the weights on edges that originate in A and end in B, and *vol(A)* is the sum of weights of all edges that originate in A.

$$NCut = \frac{cutsize(A,B)}{vol(A)} + \frac{cutsize(B,A)}{vol(B)} \tag{1}$$

The most widely used metric for discovering the sub-goals is the number of visits to a state, i.e. the frequency of the state (Stolle & Precup, 2002), (Kretchmar et al., 2003), (Şimşek & Barto, 2004), (Shi et al., 2007). Although the complexity of these algorithms is low, the performance is not as good as that of the graph based approaches.

In (Stolle & Precup, 2002), the authors discover sub-goals with a strategy based mainly on random sub-goal generation. This method can be classified as metric based because they eliminate sub-goals according to the number of visits during training.

In (Kretchmar et al., 2003), Kretchmar *et al* propose a sub-goal discovery approach by merging the frequency and the distance metrics. Initially, the successful T trajectories without cycles in the previous trials are stored. They calculate the frequency and distance measure of the state i by Equations 2 and 3. g is the goal state and l_t is the trajectory length. The metric to define the sub-goal is the multiplication of frequency and distance measures (i.e. $C_i = F_i \times D_i$).

$$F_i = \frac{\#\ of\ trajectories\ with\ state\ i}{T} \tag{2}$$

$$d_i = 2 \cdot \min_{t \in T} \min_{s \in \{s_0, g\}} \frac{|s - i|}{l_t} \tag{3}$$

$$D_i = e^{-1.0 \cdot (\frac{1-d_i}{a})^b} \tag{4}$$

The second approach to improve the performance of RL is transferring the experiences from previous tasks by shortening the exploration period. There are two tasks in the definition of a transfer learning method: the source task and the target task, and we can group transfer learning studies into two distinct classes, namely, in-domain and inter-domain transfer in terms of the state and action domains of the tasks.

In (Torrey et al., 2007), policies are stored as relational macros called as *Skills*. Skill transfer is achieved via inductive logic programming (ILP). SARSA with Support Vector Machines is the learning engine. In this work, agent transfers the knowledge obtained in the training process of 2 on 1 Breakaway task to learn the policies for 3 on 2 and 4 on 3 Breakaway tasks.

In (Barrett et al., 2010), Barrett *et al* apply transfer learning for RL on a Nao robot. The experiment is about hitting a ball to throw it with 45 degrees. In the experiment, the state consists of the positions and velocities of the joints and the actions are the increase or decrease in the joint velocity in 10 degrees per second. In the source task, the robot has control on two shoulder joints. Therefore, the learning process is fast but the success rate of the robot is low. When the robot learns the target task, it is able to control all four joints in the shoulder. The learning process starts with the Q-values of the source task.

In (Taylor & Stone, 2007) Taylor proposes to use the learned policy for a domain as an advisor for learning other tasks in other domains. Rule transfer maps two different domains. Although it is mainly handled manually, an example is shown to learn the rule translation function between keep-away and ring-world tasks. He uses Radial Basis Functions as the action-value function. In (Taylor et al., 2007), the previous work (Taylor & Stone, 2007) is expanded by using three different function approximations: CMAC, RBF and ANN.

Fernández and Veloso (Fernández & Veloso, 2006) propose re-using the previously learned policies to improve the exploration stage of the learning process. They also construct a policy library. When a new task occurs, the agent chooses one of the previous policies probabilistically. At the end of each episode, the agent updates the weight of the selected policy using Equation 5. The main problem in this approach is determining the policy list to be reused.

$$W = \frac{1}{K}\sum_{k=0}^{K}\sum_{h=0}^{h}\gamma^{h} r_{k,h} \tag{5}$$

In (Ramon et al., 2007), the authors store the policy as a logical decision tree. For this reason, relations are the most appropriate representation for the environment and the task. In the experiments, they feed the system with relations one by one and the decision tree is constructed incrementally. When a change in the target occurs, the system adapts the tree using one of these four operations; *Splitting a leaf, Pruning a leaf, Revising an internal node, Pruning a subtree.* Although the proposed approach has a promising performance on the experiments described in the paper, the applicability of the proposed approach on real world robotic problems which are more complex than the problems in (Ramon et al., 2007) is dubious.

In (Mann & Choe, 2012), Mann and Choe define the positive transfer as the improvement on the sample complexity in the target task compared to that of base RL without loosing the convergence to the near-optimal policy. In order to guarantee the positive transfer, they defined *α-weak admissible heuristic (W:$S \times A \to R$)* as

$$V^{*}(s) - \alpha \leq Q*(S,\tilde{a}) \leq W(s,\tilde{a}) \leq \frac{1}{1-\gamma} \tag{6}$$

where α is the smallest positive value. They also assume the existence of intertask mapping (h: D $\to S_{src} \times A_{src}$ where $D \subseteq S_{trg} \times A_{trg}$), if the source and the target tasks have different state-action space. They propose the following function to initialize the action-value pairs:

$$W(s,a) = \begin{cases} min(\hat{Q}_{src}(h(s,a)) + \epsilon_{src}, \frac{1}{1-\gamma}) & \text{if (s,a)} \in \text{D} \\ \frac{1}{1-\gamma} & \text{otherwise} \end{cases} \tag{7}$$

where ϵ_{src} is the accuracy of the source task estimates ($\hat{Q}_{src}$). Finally, they tested proposed approach on an inverse kinematic problem to reach known points without complex mathematical calculations. While manipulator has control on two joints in the source task, three joints are enabled in the target task. In both tasks, the target index and joint locations form the state representation. The main weakness of the algorithm is the inability to recover when α value of admissible heuristic is not small. Additionally, proposed approach requires the intertask mapping function which is difficult to find.

Although these methods are promising, when the definition of the target problem has conflicts with the source task, it becomes necessary to define a mapping function between the states of the source and the target task. Finding such a function is generally nontrivial since it may mislead the agent in learning the policy for the target task.

3. Partial Transfer Learning

Transfer Learning is a powerful heuristic for initializing the Q-values in RL. On the other hand, it may have a side effect in misleading the agent when the target task requires conflicting decisions with the source task. In order to eliminate this side effect, the agent should know the common requirements of the source and target tasks. In this work, we propose using sub-goals for defining the commonality between tasks. In our proposed method, sub-goals are found automatically via an extended version of the frequency/distance (F/D) metric (Kretchmar et al., 2003). Since the learning procedure is incremental, we modify the approach by taking the previously found sub-goals into account. In order to cover the state space as much as possible, we extend the F/D metric to find sub-goals as separated as possible. The incremental sub-goal discovery approach used is explained in the following section in detail.

To maximize the gain, we should find the sub-goals of the source and the target tasks with the maximum information to accomplish the target task. We define an initial belief function given in Equation 8 for the target task as a Gaussian where the mean is the goal state and the variance is the difference between the initial and the goal state. An example is given in Figure 1 where the target state is (40,5) and the initial state is (2,2). By using this function, we can find the spatially closest sub-goal to the goal state which is generally the most informative one.

$$IB(s) \sim N(S_g, |S_g - S_i|) \tag{8}$$

But, in some exceptional cases, the spatially closest sub-goals may mislead the agent because of the transition probabilities. As an example, we can consider a robot navigation task in the environment shown in Figure 2. In the figure, I represents the initial state, A, B and C represent the goal states and S_a and S_b represent the sub-goal states. After the task A and B, the agent learns to reach C by using S_a and S_b. Based on the initial belief function, the agent chooses S_b as the most informative sub-goal. However, in this case S_a is more informative than S_b. To recover the agent from the misleading close sub-goals, we define the performance of the agent as the goodness of the corresponding sub-goal. We calculate the performance via Equation 9. We calculate the *"value"* of each sub-goal as the product of these two functions. If the maximum value is greater than a threshold value, the corresponding sub-goal will be the most informative one. Otherwise, the agent concludes with no commonality between source and target task and it should learn to accomplish the target task from scratch. Therefore, training is applied for only for the non-common regions of the state space. Because the domain of state and actions are the same, we can classify the proposed method as in-domain transfer learning. The pseudocode for the algorithm is given in Figure 3.

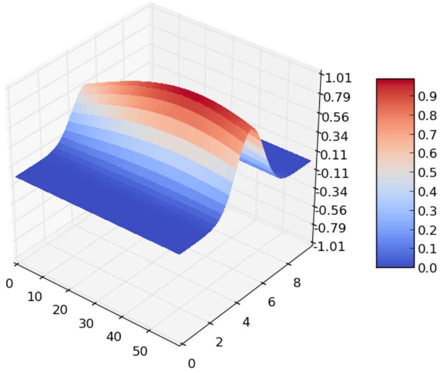


Figure 1. Example Initial Belief Function

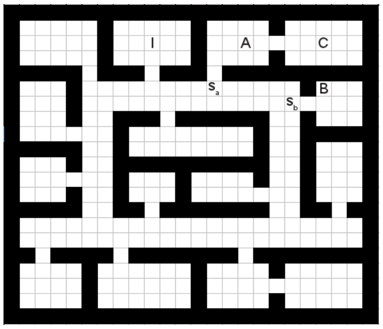


Figure 2. Example Case for Conflicting sub-goal

$$W(SG_i) = \frac{\|S_i - S_g\|}{number\ of\ steps\ to\ reach\ the\ goal} \tag{9}$$

3.1. Incremental Sub-Goal Discovery

The sub-goals give us an intuition about the important states in the state domain. Therefore, we can store the knowledge to reach the important states by these sub-goals. In order to cover the state space as much as possible, we should separate them. In this study, we have modified the approach for finding sub-goals autonomously with the frequency/distance (F/D) metric

```
Algorithm Partial Transfer Learning
  Reset Q-values
  for all Tasks do
    Find the most informative sub-goal by IB and weights
    if $IB \times Weight \geq threshold$ then
      Transfer the Q-values used to lead the agent to the common sub-goal state
    end if
    Learn the optimal policy for the task by Q-learning
    Find the sub-goal candidates by incremental sub-goal discovery method explained in the next section
  end for
```

Figure 3. Partial Transfer Learning Algorithm

(Kretchmar et al., 2003) by considering the incremental structure of the problem. When some sub-goals found in the earlier experiments are available, we extend the method to use the distance value to these states in the same manner as the start and the goal states. Therefore the *shortest distance* to the start, goal and other sub-goal states is used as the distance metric. Equation 3 is modified as in Equation 10 where SG represents the set of the previously discovered sub-goal states.

$$d_i = 2 \cdot \min_{t \in T} \min_{s \in \{s_0, g\} \cup SG} \frac{|s - i|}{l_t} \qquad (10)$$

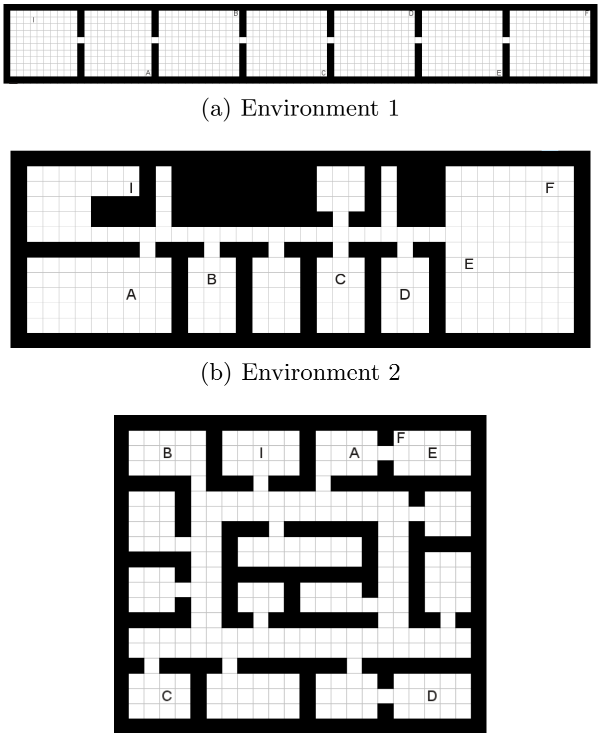
The final version of the sub-goal discovery algorithm is given in Figure 4.

4. Experimental Results

We test the algorithm on a robot navigation problem with three different world configurations shown in Figure 5. In the figures, *"I"* represents the initial state and *"A, B, C, D, E, and F"* are the goal states for different tasks. The agent learns the tasks in alphabetical order. First configuration helps us to visualize the incremental structure of the learning process. The agent starts learning to reach closer locations and uses the knowledge acquired to learn those simpler tasks to improve the learning rate of complex tasks. The second configuration is one of the floors of our department building. In this configuration, there are dead-ends and rooms whose doors open to a common corridor. The complexity of the tasks is similar to that of the previous configuration. The last world has the same configuration with the environment used in *probabilistic policy re-use* approach (Fernández & Veloso, 2006).

```
Algorithm Incremental Sub-Goal Discovery
  Reset all freq, dist
  Set the current state to the start state
  for all successful trajectories do
    repeat
      Increment freq value of the current state.
      Calculate the minimum distance (minDist) of the current state with Equation 10
      Set dist value of the current state to min(dist, minDist)
      Set the current state to the next state along the trajectory
    until current state is the goal state
  end for
  for all States do
    Set freqDist value corresponding state to the freq × dist
  end for
```

Figure 4. Sub-Goal Discovery Algorithm



(a) Environment 1

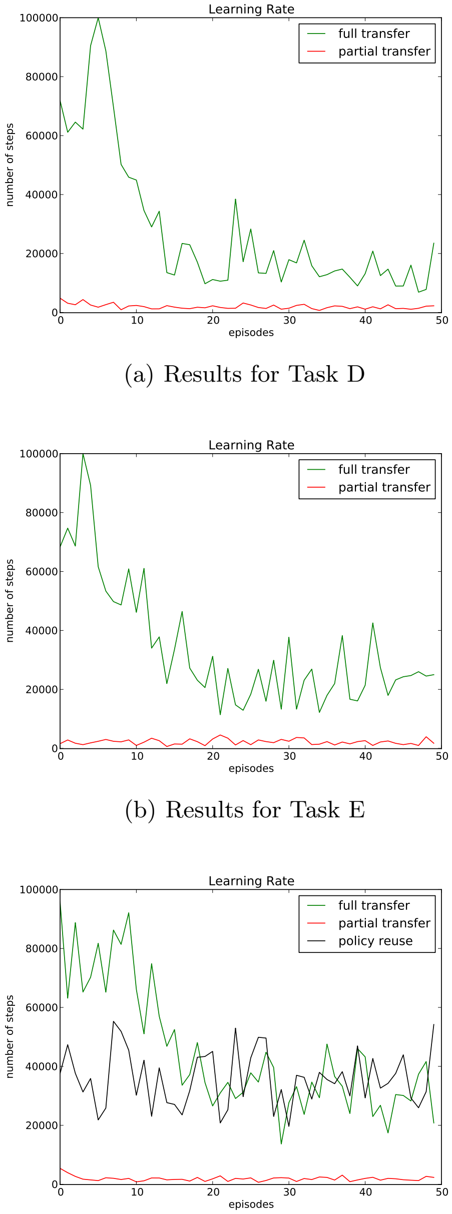
(b) Environment 2

(c) Environment 3

Figure 5. Experiment Environments

We quantize the environment into grid cells and the agent has four different actions to move into one of the four neighboring cells; north, west, south, east. If it encounters a wall, the robot stays in the same grid cell. The state representation is based on the location of the robot. We conduct the experiments with three different approaches; policy re-use, full transfer, and partial transfer. Policy re-use approach requires to have a policy library. For that purpose, the agent learns first five tasks by pure q-learning and uses policy re-use approach just for the last task as explained in (Fernández & Veloso, 2006). Because all of the approaches are probabilistic, each approach is repeated in each environment 5 times. In each experiment, the agent receives no reward except when in the goal location. When the agent reaches to the goal location, it receives a reward of +1.

We use the number of the primitive steps as the metric of the comparison to verify the improvements of the proposed approach on the learning rate. The performances of each approach are shown in Figure 6, 7 and 8. The results in Figure 6 show us the dramatic improvement on the learning rate if the complexity of the tasks are increasing. Even learning complex tasks becomes easier than the initial learning process of simpler tasks. In Figure 7, it is shown that the dead-ends have no side-effect on the improvement. We can see the performance of the proposed approach if the complexity of the tasks is not increasing in Figure 8. If the complexity does not increase, the requirement for learning nearly vanishes. Figure 8(a) shows a peak at the beginning because *task D* is far away from the previous tasks. Therefore agent needs to learn much. But even in this case it just takes a few episodes which is much faster than the other methods. We can also see the importance of the weights in calculating the *"value"* of a sub-goal in Figure 8(b). The sub-goal to reach D is the closest one for the task E. On the other hand, it over estimates the informative locations because of the locations of the doors to reach E. The agent starts to use sub-goal of task D, but it learns that the sub-goal of task A is more informative in just a few episodes. Because task E and task F are very close, both full transfer and partial transfer learning require almost no learning at all. But, it takes more time for the policy re-use because of the conflicting advises of the old policies.



(a) Results for Task D

(b) Results for Task E

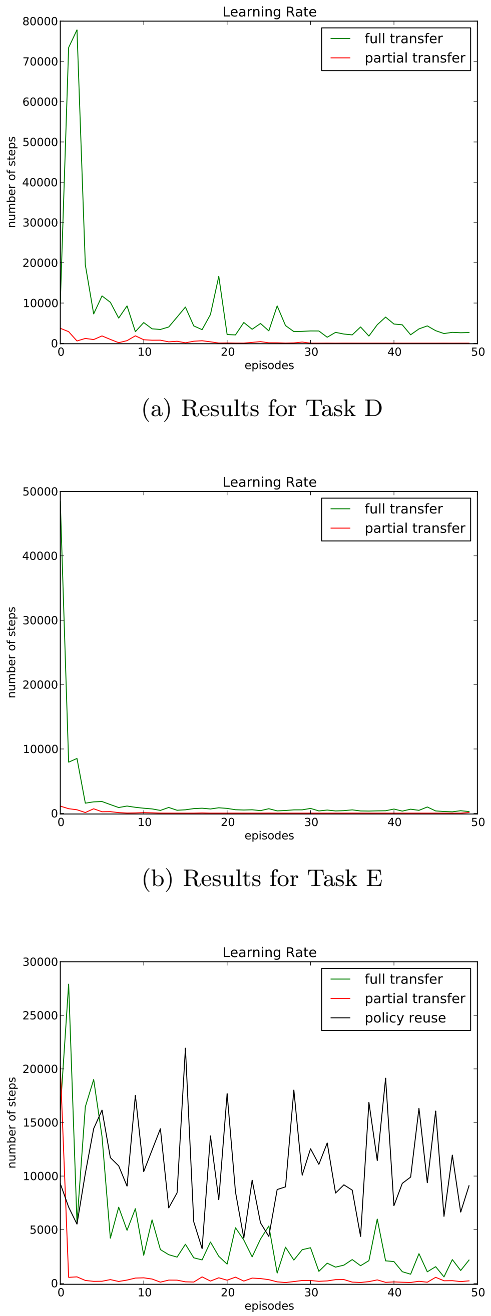
(c) Results for Task F

Figure 6. Results in World 1

5. Conclusion

In this work, we mainly address the curse of dimensionality problem and applicability of the RL in real world robotic problems. We propose to use the developmental structure of the learning process. Since the source and target tasks may have conflicting components as well as common ones, we re-use partial policies as the exploration advises instead of the whole policy. Subgoals determine the part of the policy to be reused. In this way, we aim to transfer exactly the common parts of the tasks. Additionally, we separate sub-goals as much as possible to cover the state space.

The performance of the proposed method is tested with a robot navigation problem in three different room based environments. As a result of these experiments, we can conclude that the information transfer

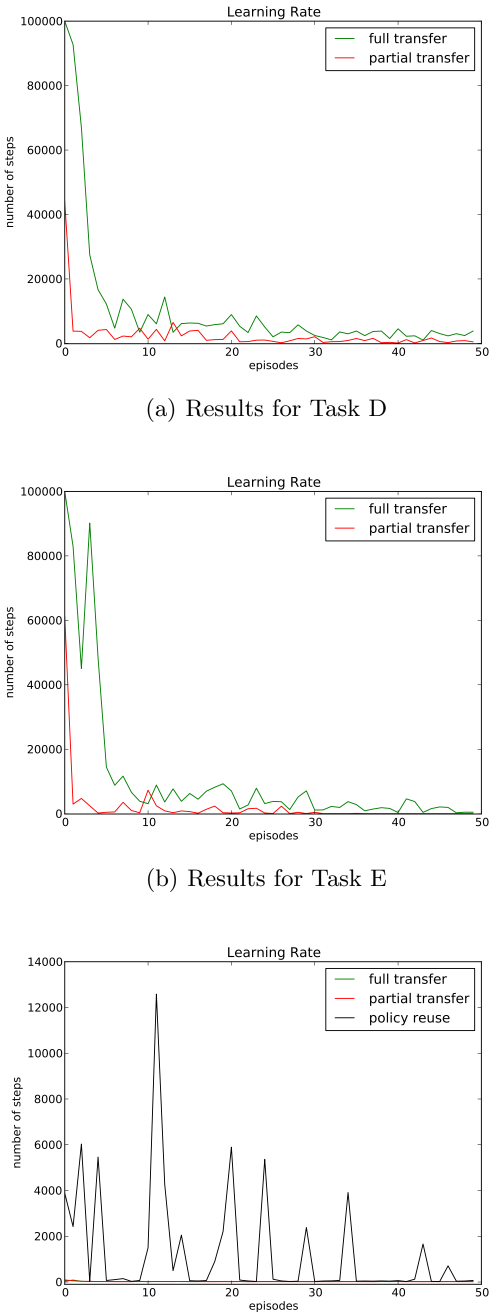


(a) Results for Task D

(b) Results for Task E

(c) Results for Task F

Figure 7. Results in World 2



(a) Results for Task D

(b) Results for Task E

(c) Results for Task F

Figure 8. Results in World 3

via sub-goals improves the learning rate of the agent significantly. Advises of the partial policies lead the agent to a closer state to the target state and the search continues from that state. This improvement significantly reduces the number of actions necessary and makes RL applicable in real world robotic problems.

As a future work, we are planning to conduct the experiments in a real world setting. Since the map of our building is the second environment in our experiments, we are expecting similar results with a real mobile robot, depending on the primitive action accuracy. Additionally, we will extend the proposed approach for inter-domain transfer.

References

Barrett, S., Taylor, M.E., and Stone, P. Transfer learning for reinforcement learning on a physical robot. In *Ninth International Conference on Autonomous Agents and Multiagent Systems-Adaptive Learning Agents Workshop (AAMAS-ALA)*, 2010.

Şimşek, Ö. and Barto, A.G. Using relative novelty to identify useful temporal abstractions in reinforcement learning. In *Machine Learning-International Workshop then Conference-*, volume 21, pp. 751. Citeseer, 2004.

Elias, P., Feinstein, A., and Shannon, C. A note on the maximum flow through a network. *Information Theory, IRE Transactions on*, 2(4):117–119, 2002. ISSN 0096-1000.

Fernández, F. and Veloso, M. Probabilistic policy reuse in a reinforcement learning agent. In *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, pp. 720–727. ACM, 2006.

Goldberg, A.V. and Tarjan, R.E. A new approach to the maximum-flow problem. *Journal of the ACM (JACM)*, 35(4):921–940, 1988. ISSN 0004-5411.

Hengst, B. Discovering hierarchy in reinforcement learning with HEXQ. In *Machine Learning-International Workshop then Conference-*, pp. 243–250. Citeseer, 2002.

Kazemitabar, Seyed and Beigy, Hamid. Using strongly connected components as a basis for autonomous skill acquisition in reinforcement learning. In Yu, Wen, He, Haibo, and Zhang, Nian (eds.), *Advances in Neural Networks – ISNN 2009*, volume 5551 of *Lecture Notes in Computer Science*, pp. 794–803. Springer Berlin / Heidelberg, 2009a.

Kazemitabar, Seyed and Beigy, Hamid. Automatic discovery of subgoals in reinforcement learning using strongly connected components. In Köppen, Mario, Kasabov, Nikola, and Coghill, George (eds.), *Advances in Neuro-Information Processing*, volume 5506 of *Lecture Notes in Computer Science*, pp. 829–834. Springer Berlin / Heidelberg, 2009b.

Kretchmar, R.M., Feil, T., and Bansal, R. Improved automatic discovery of subgoals for options in hierarchical reinforcement learning. *Journal of Computer Science and Technology*, 3(2):9–14, 2003.

Mann, T.A. and Choe, Y. Directed exploration in reinforcement learning with transferred knowledge. In *Workshop on Reinforcement Learning*, pp. 59, 2012.

Menache, I., Mannor, S., and Shimkin, N. Q-cutdynamic discovery of sub-goals in reinforcement learning. *Machine Learning: ECML 2002*, pp. 187–195, 2002.

Ö. Şimşek and A.P. Wolfe and A.G. Barto. Identifying useful subgoals in reinforcement learning by local graph partitioning. In *Proceedings of the 22nd international conference on Machine learning*, pp. 816–823. ACM, 2005. ISBN 1595931805.

Piaget, Jean. *The psychology of intelligence*. London: Routledge & Kegan Paul, 1950.

Ramon, J., Driessens, K., and Croonenborghs, T. Transfer learning in reinforcement learning problems through partial policy recycling. *Machine Learning: ECML 2007*, pp. 699–707, 2007.

Shi, C., Huang, R., and Shi, Z. Automatic Discovery of Subgoals in Reinforcement Learning Using Unique-Dreiction Value. In *Cognitive Informatics, 6th IEEE International Conference on*, pp. 480–486. IEEE, 2007.

Stolle, M. and Precup, D. Learning options in reinforcement learning. *Abstraction, Reformulation, and Approximation*, pp. 212–223, 2002.

Taylor, M.E. and Stone, P. Cross-domain transfer for reinforcement learning. In *Proceedings of the 24th international conference on Machine learning*, pp. 879–886. ACM, 2007.

Taylor, M.E., Stone, P., and Liu, Y. Transfer learning via inter-task mappings for temporal difference learning. *Journal of Machine Learning Research*, 8 (1):2125–2167, 2007.

Torrey, L., Shavlik, J., Walker, T., and Maclin, R. Relational macros for transfer in reinforcement learning. In *Proceedings of the 17th international conference on Inductive logic programming*, pp. 254–268. Springer-Verlag, 2007. ISBN 3540784683.