Graph Laplacian Based Transfer Learning in Reinforcement Learning (Short Paper)

Yi-Ting Tsao
Department of Computer Science
National Tsing-Hua University
HsinChu, Taiwan
yiting.tsao@gmail.com

Ke-Ting Xiao
Department of Computer Science
National Tsing-Hua University
HsinChu, Taiwan
peter.xiau@gmail.com

Von-Wun Soo
Department of Computer Science
National Tsing-Hua University
HsinChu, Taiwan
soo@cs.nthu.edu.tw

ABSTRACT

The aim of transfer learning is to accelerate learning in related domains. In reinforcement learning, many different features such as a value function and a policy can be transferred from a source domain to a related target domain. Many researches focused on transfer using hand-coded translation functions that are designed by the experts a priori. However, it is not only very costly but also problem dependent. We propose to apply the Graph Laplacian that is based on the spectral graph theory to decompose the value functions of both a source domain and a target domain into a sum of the basis functions respectively. The transfer learning can be carried out by transferring weights on the basis functions of a source domain to a target domain. We investigate two types of domain transfer, scaling and topological. The results demonstrated that the transferred policy is a better prior policy to reduce the learning time.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning – *knowledge acquisition, parameter learning.*

General Terms

Experimentation, Theory.

Keywords

reinforcement learning, transfer learning, graph Laplacian

1. INTRODUCTION

One of the disadvantages in reinforcement learning (RL) [1] is that two different domains with different initial states and goal states must be learned separately to acquire an optimal policy for each domain. It would waste time to simply learn twice in two different domains even if they might share some similar subtasks. Transfer learning is an approach to improve the performance of cross domains by avoiding redundant learning.

In a reinforcement learning problem, the value function provides a guideline for action selection in a given state that is known as a policy. Many transfer methods that transfer different features from a source domain to a target domain have been proposed [2, 3, 4]. One work is a rule transfer method that acquires some rules that approximate the policy in a source domain and translates into ones that can be used as a policy for a target domain [2]. Thus an agent may apply the translated policy that is acquired by hand-coded translation functions and revise a partial policy in a target domain. However, designing general translation functions becomes a problem. Another work based on case-based reasoning uses a similar idea but it acquires rules using a decision-tree method [3]. The other work is to transfer the policy from a source domain to a target domain directly [4] but it also requires hand-coded translation functions. Proto-value functions derived from spectral graph theory, harmonic analysis, and Riemannian manifold can be used to represent a set of the basis functions to approximate a value function [5, 6, 7]. A novel transfer method has been proposed to reuse a set of the basis functions from a source domain and just to learn the weights of the set of the basis functions to compose a value function for a target domain. This method can transfer domain features without hand-coded translation functions but it needs some exploring trials for a target domain to acquire the combination weights.

The aim of the transfer learning is to use the knowledge learned from a source domain to accelerate learning in a related target domain. In this paper, we propose a transfer method to obtain a better prior policy from a source domain to reduce the learning time in a similar target domain without hand-coded translation functions by spectral graph theory.

2. BACKGROUND

Most reinforcement learning researches are based on Markov Decision Processes (MDP) and a value function to guide an agent's actions in solving a domain. However, a value function can be too rigid to apply to a domain such that to transfer it directly to another domain is hard. Finding a set of suitable basis functions to express the value function helps the transfer. In this paper, the development is based on a discrete MDP and the spectral graph theory.

2.1 Markov Decision Process

A discrete Markov Decision Process M which is defined by a 4-tuple $(S, A, P^a_{ss'}, R^a_{ss'})$ where S is a finite set of states, A is a finite

Cite as: Graph Laplacian Based Transfer Learning in Reinforcement Learning (Short Paper), Yi-Ting Tsao, Ke-Ting Xiao, Von-Wun Soo, *Proc. of 7th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2008)*, Padgham, Parkes, Müller and Parsons (eds.), May, 12-16., 2008, Estoril, Portugal, pp. 1349-1352.

set of actions, $P_{ss'}^{a}$ and $R_{ss'}^{a}$ represent the probability and reward of transiting to state *s'* when taking action *a* on state *s* respectively [1]. A function which determines the action that an agent should take at any state that the agent could reach is called a policy π. A policy is a mapping from a state to a unique action. The value function V^{π} represents the value by using policy function *π* and the optimal policy *π** is defined as a unique optimal value function *V** that can maximize the expected reward starting at a given state *s* with discount factor γ. The optimal value function *V**(*s*) is defined as follows.

$$V^*(s) = \max_a \sum_{s'} P_{ss'}^{a}(R_{ss'}^{a} + \gamma V^*(s'))$$

The value function is represented in tabular form with one output for each input tuple. However, the state space in the real world is often so huge that to memorize the value table is impossible. We can approximate the value function in terms of a linear combination of a set of the basis functions as:

$$V^{\pi} = \alpha_1 V_1^{B} + \dots + \alpha_n V_n^{B}$$

where each V_i^B is a basis function. Approximating by the basis functions saves a lot of memory. However, different sets of the basis functions may affect the function approximation. Therefore, for an agent to have good performance, selecting good basis functions to make good value approximation plays an important role.

2.2 Spectral Graph Theory

A Fourier analysis is to decompose a function in terms of a sum of trigonometric functions with different frequencies that can be combined together to represent the original function. Each frequency of trigonometric functions is inversely proportional to its importance in representing characteristics of the function. Therefore, if two functions are similar, their trigonometric functions tend to be similar at low frequencies and differ at high frequencies.

A graph Laplacian can be defined as the combinatorial Laplacian or the normalized Laplacian [8]. The combinatorial Laplacian *L* of the undirected unweighted graph *G* is defined as *L* = *D* - *A* where *A* is the adjacency matrix and *D* is a diagonal matrix whose entries are the row sums of *A*. In problem solving, the states are represented as the vertices and the edges represent the connection (undirected) or transitions (directed) between the states so that one state can reach another. Let *u* and *v* represent two states in a graph and d_v represents the degree of *v*, a graph Laplacian *L*(*u*, *v*) is defined as follows:

$$L(u,v) = \begin{cases} d_v & \text{if } u = v \\ -1 & \text{if } u \text{ and } v \text{ are adjacent} \\ 0 & \text{otherwise} \end{cases}$$

Let *f* denote a function mapping each vertex *u* of the graph into a real number. The combinatorial Laplacian *L* acts on a function *f* as

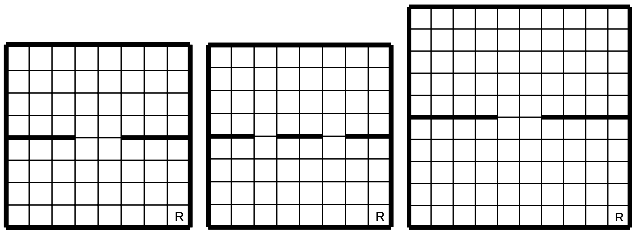
$$Lf(u) = \sum_{u \sim v} (f(u) - f(v))$$

where *u* and *v* are adjacent vertices. Functions that solve *Lf* = 0 are called harmonic functions [9]. It turns out that to find the harmonic functions is equivalent to finding the eigenvectors (or eigenfunctions) of *Lf* = *λf*, where *f* is the eigenfunction and *λ* is the associated eigenvalue. A smaller eigenvalue implies a smoother eigenfunction. Furthermore, we can extend the idea in general with normalized graph Laplacian[8]. In our cases, the normalized graph Laplacian has the better consequences than the combinatorial Laplacian.

The spectral analysis of the graph Laplacian operator provides an orthonormal set of the basis functions that can approximate any square-integrable functions on a graph [8]. These basis functions which are called as proto-value functions in [5, 6, 7] construct a global smooth approximation of a function on the graph. In other words, the function can be decomposed into a sum of the basis functions [10]. Besides, the notion of the spectral analysis on graph Laplacian is similar to the Fourier analysis. The basis functions of a graph Laplacian corresponding to the smaller eigenvalues represent more valuable features and are thus more important. It implies that if two graphs are similar, their features tend to be similar at low-order basis functions and different at high-order basis functions.

3. THE TRANSFER METHOD

In [6], the authors distinguished three transfer types: task transfer, topological domain transfer, and scaling domain transfer as shown in Figure 1. The domain transfer problem means only the topology of the state space changes and rewards do not change. In this paper, we focus on both topological and scaling domain transfer.



(a) 8 x 8 grids source (b) 8 x 8 grids topological target (c) 10 x 10 grids scaling target

Figure 1. The example of topological and scaling domain transfer.

The transfer algorithm is described in Figure 2. The first step is to collect the topological knowledge of both domains retrieving basis functions respectively. The second step is to compute the corresponding basis functions of the graph Laplacian. The third step is to compute the coefficients of the basis functions approximating the real value function in the source domain. The fourth step is to approximate the target value function in terms of the target basis functions and the weights that are obtained from the source domain. The last step is to acquire the target policy through the approximated target value function.

The reason why the transfer algorithm works is that the basis functions of both domains with the same order play the same important role for the value functions at each domain respectively. Therefore, we transfer the obtained weights from a source domain to a target domain. If two domains are similar, the basis functions tend to be similar. It does not imply similar numeric value but similar structure as shown in Figure 3. On the one hand, a small change of the domain cannot affect the global smooth structure so the low-order basis functions for the target domain tend to be the same as the corresponding basis functions for the source domain. On the other hand, the high-order basis functions for the target domain are affected by a small change of the domain so the target policy can be obtained from the target low-order basis functions that are similar to the source low-order basis functions and the high-order basis functions that are modified by a small change.

```
1. Perform random walk of $M$ trials, each with maximum
   $N$ steps on source domain and target domain and build
   the undirected graphs $G_S$, $G_T$ respectively.
2. Construct the normalized Laplacian on $G_S$, $G_T$ and
   solve the Laplacian to obtain the basis functions $V_S^B$,
   $V_T^B$. Sort them by eigenvalue in ascending order.
3. Approximate the source value function $V_S^*$ using $V_S^B$
   by the least-square error fit method to obtain the
   weight $w_i$ corresponding to the source basis
   function $V_{Si}^B$.
4. Transfer the weight $w_i$ from the source basis function
   $V_{Si}^B$ to the corresponding target basis function $V_{Ti}^B$.
       $V_T = \sum_i w_i * V_{Ti}^B$
5. Convert the approximation target value function to the
   target policy.
```

Figure 2. Pseudo-code of the transfer algorithm.

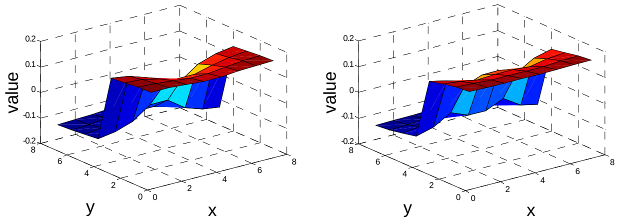


Figure 3. The similar structure of the basis functions of Figure 1(a) and 1(b).

4. EXPERIMENTS

First of all, we illustrate the basis functions of the graph Laplacian with different size of domains with the same topology. The upper two graphs and lower two graphs in Figure 4 and 5 show some low-order and high-order basis functions from graph Laplacian respectively. We note the two upper graphs in Figure 4 and 5 that represent the smoothest k basis functions of different domains respectively tend to be very similar while the lower graphs are not.

We design the experiments on scaling and topological domain transfer and evaluate the performance of an agent in the domains using different policies: random, transferred and optimal respectively. The agent is an active agent with ε-greedy behavior [1]. In other words, the agent has probability ε to act at random. A random policy selects an action at random, a transferred policy is obtained from the transfer method, and an optimal policy selects an action based on the optimal value function obtained by the value iteration method. The results are shown in Figure 6 and 8. The x-axis and the y-axis represent the number of states and the number of steps reaching the reward respectively. The diamond, square, and triangle lines represent the random, transferred and optimal policies respectively. Each point in the line represents the average number of steps reaching the reward state over all possible initial states.

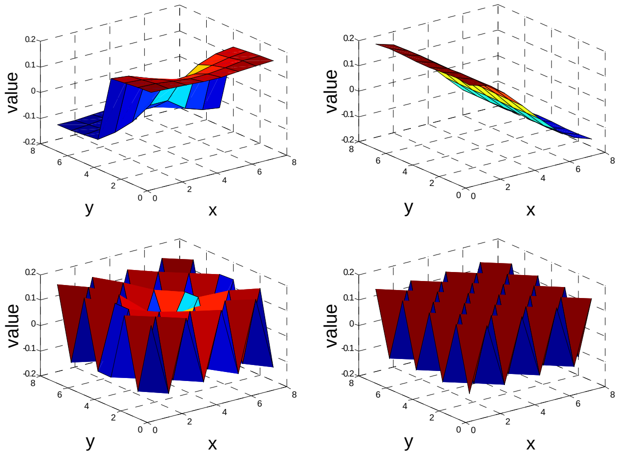


Figure 4. The basis functions of Figure 1(a).

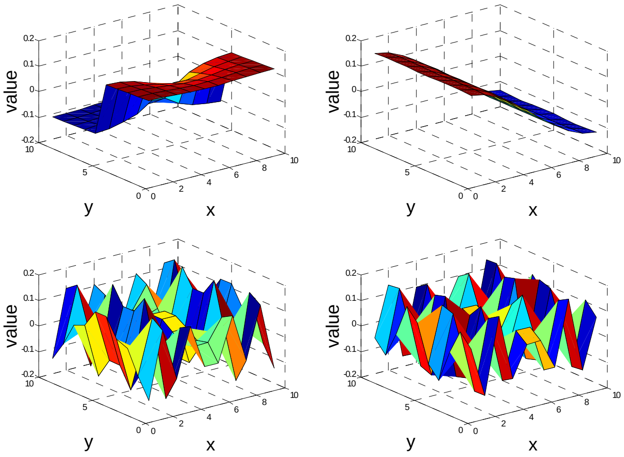
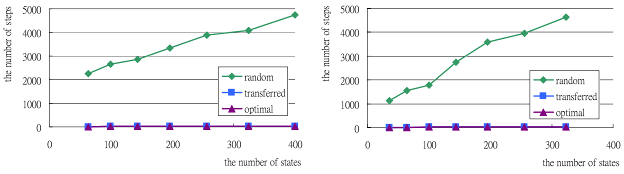


Figure 5. The basis functions of Figure 1(c).

4.1 Scaling Domain Transfer

These experiments investigate the effects of the scaling domain transfer. We separate the scaling domain transfer into two cases: up-scaling and down-scaling. The topology of each case is the same as shown in Figure 1(a). In up-scaling case, we choose the 6x6 grids world as a source domain and 8x8, 10x10, 12x12, 14x14, 16x16, 18x18, and 20x20 grids as target domains. In down-scaling case, we choose the 20x20 grids world as a source domain and 6x6, 8x8, 10x10, 12x12, 14x14, 16x16, and 18x18 grids as target domains. The results show that regardless of the size in a target domain, the transferred policy still performs very close to the optimal one as shown in Figure 6.



(a) up-scaling case (b) down-scaling case

Figure 6. The results of scaling domain transfer.

4.2 Topological Domain Transfer

These experiments investigate the effects of the topological domain transfer. The topology in the source domain is shown in Figure 1(a) and we design three different topological cases as target domains. Figure 7(a) represents a case that splits the door into two separating doors, Figure 7(b) represents a case that splits the door into two separating doors farther, and Figure 7(c) represents a case that increases the size of a door.

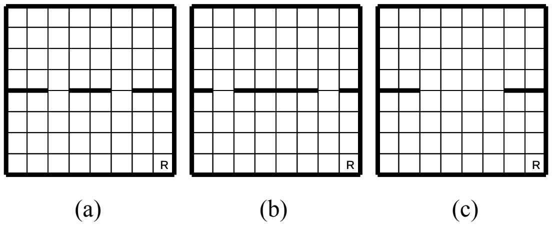


Figure 7. The topological transfer targets.

The results demonstrate that if both domains are similar enough, the transferred policy may perform very close to the optimal one as shown in Figure 8(a). However, when the source and target domains are not similar enough as in the case of Figure 8(c) in which the larger size domains are more affected than the smaller ones. Besides, when the number of states is small, the effect of a small change in the domain is large, but when the number of states is large enough, the effect of a small change in the different size domains is similar as shown in Figure 8(b).

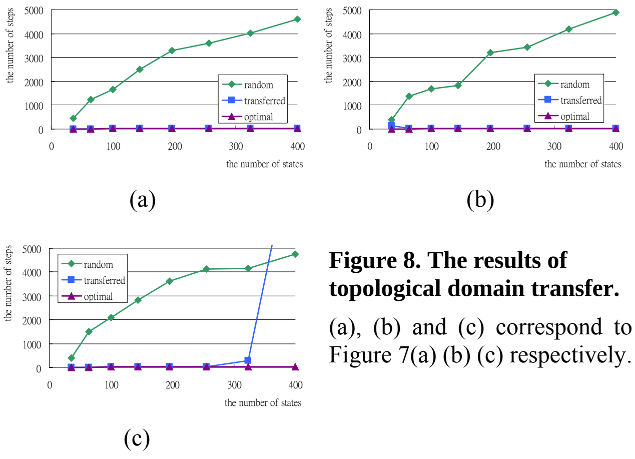


Figure 8. The results of topological domain transfer.

(a), (b) and (c) correspond to Figure 7(a) (b) (c) respectively.

5. CONCLUSIONS

The theoretical analysis of the transfer method is based on the spectral analysis on graph Laplacian. The low-order basis functions of the graph Laplacian represent major features of a value function while the high-order ones represent miner features. If the low-order basis functions of the source and target domains are similar, the transfer method performs well. In other words, similar domains tend to keep similar distributions in low-order basis functions so we can transfer the weights of the source domain to the target domain and acquire a good approximate policy for the target domain. In this paper, we have proposed a domain transfer method based on the topology of the state space to support the transfer for reinforcement learning. Our experimental results show that if two domains are similar topologically, the policy learned from transfer learning can be very close to the optimal one. However, how to determine if a topological similarity is enough to apply the transfer learning to ensure its error bound be close to the optimality still needs more theoretical analysis. This work only considers the state space topology of the problem but not the rewards. We should revise the domain transfer method by considering how to map a state in a source domain to the corresponding one in a target domain that considers the rewards in future work.

6. ACKNOWLEDGMENTS

This work is supported by the National Science Council of Taiwan under grant number NSC 96-2628-E-007-044-MY3.

7. REFERENCES

[1] Richard S. Sutton and Andrew G. Barto. Reinforcement learning: an introduction. MIT Press, 1998.

[2] Matthew E. Taylor and Peter Stone. Cross-domain transfer for reinforcement learning. In Proceedings of the Twenty-fourth International Conference on Machine Learning, 2007.

[3] Andreas von Hessling and Ashok K. Goel. Abstracting reusable cases from reinforcement learning. In Proceedings of the Sixth International Conference on Case-Based Reasoning Workshop, 2005.

[4] Mattew E. Taylor, Shimon Whiteson, and Peter Stone. Transfer via inter-task mappings in policy search reinforcement learning. In Proceedings of the Sixth International Conference on Autonomous Agents and Multiagent Systems, 2007.

[5] Sridhar Mahadevan. Proto-value functions: developmental reinforcement learning. In Proceedings of the Twenty-second International Conference on Machine Learning, 2005.

[6] Ferguson Kimberly and Sridhar Mahadevan. Proto-transfer learning in Markov decision processes using spectral methods. In Proceedings of the Twenty-third International Conference on Machine Learning Workshop on Structural Knowledge Transfer for Machine Learning, 2006.

[7] Sridhar Mahadevan and Mauro Maggioni. Proto-value functions: a Laplacian Framework for learning representation and control in Markov decision processes. Technical Report, 2006.

[8] Fan R. K. Chung. Spectral graph theory. American Mathematical Society, 1997.

[9] Sheldon Axler, Paul Bourdon, and Ramey Wade. Harmonic function theory. Springer, 2001.

[10] Mikhail Belkin and Partha Niyogi. Semi-supervised learning on Riemannian manifolds. Machine Learning, 2004.