HIERARCHICAL FUNCTIONAL CONCEPTS FOR KNOWLEDGE TRANSFER AMONG REINFORCEMENT LEARNING AGENTS

A. MOUSAVI, M. NILI AHMADABADI, H. VOSOUGHPOUR,
B. N. ARAABI AND N. ZAARE

Abstract. This article introduces the notions of functional space and concept as a way of knowledge representation and abstraction for Reinforcement Learning agents. These definitions are used as a tool of knowledge transfer among agents. The agents are assumed to be heterogeneous; they have different state spaces but share a same dynamic, reward and action space. In other words, the agents are assumed to have different representations of an environment while having similar actions. The learning framework is Q-learning. Each dimension of the functional space is the normalized expected value of an action. An unsupervised clustering approach is used to form the functional concepts as some fuzzy areas in the functional space. The functional concepts are abstracted further in a hierarchy using the clustering approach. The hierarchical concepts are employed for knowledge transfer among agents. Properties of the proposed approach are tested in a set of case studies. The results show that the approach is very effective in transfer learning among heterogeneous agents especially in the beginning episodes of the learning.

1. Introduction

Reinforcement learning (RL) agents are very popular because of their ability to learn in interaction with the environment in face of limited prior knowledge and minimal environmental feedback. Nevertheless, one of the main deficits of an RL agent is to be slow in learning. In addition, knowledge transfer among heterogeneous RL agents is a big challenge. Human beings -and some animals- apply RL-like methods as well, however they are fast learners with the capability of transferring their knowledge to each other in a number of ways. One may contribute these characteristics to humans' knowledge abstraction capabilities. Therefore, we target developing a knowledge abstraction approach to improve RL methods in terms of learning speed and transferability of gained knowledge. By knowledge abstraction we mean clustering the experiences to form granules of knowledge as independent from the observed situations as possible. We call these granules functional concepts.

According to Zentall's definition [26], concept is the agent's representation of its environment. So, the most important problems in concept learning methods are how to form that internal representation, and how to use it in decision making.

Received: December 2013; Revised: December 2014; Accepted: June 2015

Key words and phrases: Reinforcement Learning, Transfer Learning, Heterogeneous Agents, Hierarchical Concepts.

Knowledge transfer is almost a new and challenging area in the field of RL [17, 22]. The goal of transfer is to speed up the learning process of an agent in a target task by using the knowledge of a different agent that has already learned a related task. The main challenge of knowledge transfer is the heterogeneity of the target and source agents. Lazaric [8] classifies the transfer problems into three categories; goal, dynamic and domain transfer problems. A problem in which the agents share the same context (i.e., state and action spaces) and the same transition model, but have different reward functions, is a goal transfer problem. A problem in which tasks share the same context and the same reward function, but have different transition models, is a dynamics transfer problem. In the case of domain transfer, each agent may have different dynamics, goals and state-spaces. This is the most general and complex problem of transfer.

There is also another category of problems that is not mentioned in the Lazaric's classification. This category is referred to as representation transfer problems [19]; the agents share the same transition model, reward function and action spaces, but have different state spaces. This article focuses on presenting an unsupervised algorithm for solving this kind of problems. Difference between the representations of different agents may be due to the difference between their sensors or the way they control their attention in the environment.

Besides, an agent may have a poor performance with a representation or attention control in an environment and needs to change it. If experience is expensive then it is desirable to leverage the previous knowledge to improve the learning with the new representation or attention control.

In this paper, we propose a learning framework for unsupervised formation of abstract concepts and their hierarchy in RL agent's functional space. By using extracted concepts and the hierarchy formed on top of those concepts in the source agent, the target agent with a different perceptual space can speed up the convergence of its action-values and consequently accelerates its learning.

In the next section, some preliminary definitions are reviewed and discussed. The notions of functional space and functional concepts are discussed in sections 3 and 4, respectively. The relation of the approach and the transfer of knowledge in RL is discussed in section 6. Section 7 contains the simulations and results. The last section contains the conclusion and proposes some future works.

2. **Background**

In this section, some main definitions are reviewed and discussed. Firstly, we discuss the different levels of concept abstraction as proposed by Zentall [26]. Then the main learning framework which is Q-learning and the notion of knowledge transfer in RL are reviewed and discussed.

2.1. **Concept Abstraction.**

Several definitions and theories have been proposed to explain the humans' mental models of concepts and how those concepts are formed in our minds [1]. In addition, studies in cognitive psychology on evidences in humans' behavior show that the concept is not necessarily the thing which exists in the environment and is observed. For example the concept of an apple in the mind is more informative than what we percept by vision or other sensors. Moreover, the concept is not the word or sound used to express it. Sometimes the concept of an object or situation is clear for us, but we cannot find a word to express it. In addition, the concepts and the words or sounds used to express them are formed distinctly in the mind.

These evidences show that the concepts are modeled in the mind in a manner independent from perceptual or communicative spaces [4, 12]. More evidences about concept leaning in human infants show that early concepts in children are functional concepts.

Concepts by an artificial system may be formed in different spaces; including perceptual and functional spaces. By functional space we mean a representation space in which items with similar functionalities form distinct clusters. Abstraction in the perceptual space does not result in state-independent abstraction while abstraction in the functional space does.

Moreover, in the real environments, the perceptual space usually is continuous and complex and the concepts may be very scattered in the perceptual space; which is not the case in the functional space. In addition, functionality can be defined easily for RL agents.

Therefore, we form the abstract concepts in the functional space. The concepts are associative concepts [26]. It means that the stimuli within the classes of this type of concepts bear no obvious similarities, but rather cohere because of shared functional properties. In other words, the exemplars of a concepts do not necessarily resemble each other perceptually and all the information needed to categorize stimuli into concepts are out of the perceptual space.

2.2. **Reinforcement Learning Agent.** An RL agent acts in a sequential manner to maximize a received reward signal from the environment [15]. The environment is typically formulated as a finite-state Markov decision process (MDP) defined as follows.

Definition 2.1. A Markov Decision Process (MDP) is a tuple $\langle S, A, T, r\rangle$, where S is the set of all states, A is the set of all actions, $T : S \times A \rightarrow p(S)$ is the state transition function and $r : S \times A \rightarrow R$ is the reward function. $p(S)$ is the set of probability distributions over the set of S.

At each time step, t, the agent senses the environment's state, $s_t \in S$, and performs an action, $a_t \in A$. As a consequence of its action, the agent receives a numerical reward, $r_{t+1} \in R$, and finds itself in a new state s_{t+1}. The objective of the agent is to learn a policy for acting, $\pi : S_t \rightarrow A_t$, in order to maximize its cumulative reward.

One popular RL technique is Q-learning which involves learning a Q-function [23]. The Q-function, $Q(s, a)$, estimates the discounted cumulative reward starting in state s and taking action a and following the current policy thereafter. When the number of states is finite and small, the Q-function is represented as a table. Given the optimal Q-function, the optimal policy is to take the action $\underset{a\in A}{argmax}\, Q(s_t, a)$. The Q-functions are recursively updated after each step according to the following equation:

$$Q(s_t, a_t) \longleftarrow (1-\alpha)Q(s_t, a_t) + \alpha(r_t + \gamma \underset{a\in A}{max}\, Q(s_{t+1}, a)) \tag{1}$$

where $\gamma \in [0,1]$ and $\alpha \in (0,1]$ are called discount factor and learning rate, respectively. Under certain conditions, Q-learning is guaranteed to converge to an accurate Q-function [24].

RL methods are divided into two distinct categories; methods that must follow the policy they are learning about, called *on-policy* methods, and those that can learn from behavior generated by a different policy, called *off-policy* methods [15]. Q-learning is an off-policy method in that it learns the optimal policy even when actions are selected randomly. It only requires that all actions be tried in all states, whereas on-policy methods require that actions be selected by a specific probabilities.

2.3. **Knowledge Transfer.**

Transfer learning is almost a new and challenging area in the field of RL [17, 22]. The goal of transfer in reinforcement learning is to speed up the learning process of an agent in a target task by using the knowledge of a different agent that has already learned a related task.

If two agents are homogeneous, then the transfer learning can be applied directly between agents. But, if the agents are heterogeneous, the knowledge from an agent can be transferred to another only if one can distinguish some common properties between the source and target agents. The heterogeneity between agents can be due to the difference in their environment's dynamic, reward or the state-action spaces [8]. In this article, the heterogeneity among agents is assumed to be due to the difference between their state spaces. In other words, it is assumed that the agents have identical dynamic, reward and action spaces but differ in the perceptual spaces. This situation is referred to as *representation transfer* in [19].

The main problem of knowledge transfer between heterogeneous agents is to find a common language between the agents. It, actually, requires a mapping to translate some properties of the source to the target agent. In [22], this problem is referred to as the mapping problem. Many transfer approaches assume that a human provides such information [16, 20, 25]. Sometimes, a domain can be constructed such that in that domain two agents become homogeneous. Relational learning is useful for creating such domains [3, 7, 13]. Another way of finding the common properties is to generate several possible mappings among the properties and allow the target agent to try them all. Taylor et. al. [18] perform an exhaustive search. Mihalkova et. al. [9] limit their search in Markov logic network, requiring that mapped predicates have matching arity and argument types. Soni and Singh [14] not only limit the candidate mappings by considering object types, but also avoid a separate evaluation of each mapping by using options in RL transfer.

In this article, we assume that the mapping between action spaces is given and is a one-to-one mapping, but the mapping between state spaces is unknown. As the functional concepts are defined in the space of actions, we use them as a common language between agents to transfer the knowledge.

3. **Functional Space**

Real world concepts are usually scattered irregularly and asymmetrically in an agent's perceptual space. It is due to the fact that concepts are mostly associative and two neighbor observations in the perceptual space do not necessarily have the

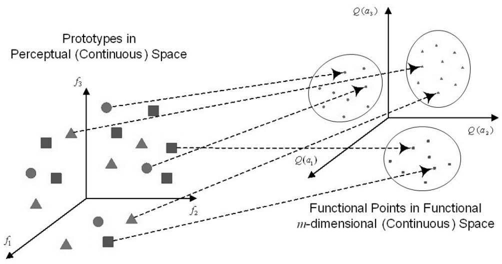
$Q(a_3)$
Prototypes in
Perceptual (Continuous) Space
f_3
$Q(a_2)$
$Q(a_1)$
f_2
f_1
Functional Points in Functional
m-dimensional (Continuous) Space

Figure 1. A Schematic View of Samples in the Perceptual Space and the Corresponding Functional Points in the Agent's Functional Space

same functionality. In addition, samples of a relational concept may form disjoint clusters in the perceptual space.

Difference in the perceptual space of different agents is also common as the agents control their attention and process their sensory information, differently. In addition, agents may even have different sensors; which is the case in robots.

As a result, having a representation space in which the samples of a concept -relational or associative- form a single cluster is highly desired; especially if agents with different perceptual spaces share that space.

One of the desired properties of this space is to be agent-centered rather than being environment-centered. By agent-centered we mean that the space is based on the values of the agent's expected reward. The reward is the result of the agent's interaction with the environment. A candidate for such abstract space could be the space of agent's best response to perceptual stimuli [10], agent's emotional space, etc. This space will be used to serve our main objective; forming a hierarchy of concepts and transferring knowledge among agents with different perceptual spaces.

Figure 1 schematically shows the goal space; called functional space. The functional space (F) is a continuous m-dimensional space where m is the size of agent's action set. In this figure m is three. The i^{th} dimension of the functional space represents the normalized expected value of action a_i; namely $Q(a_i)$. Each perceptual sample is mapped into a point in the functional space and the points are clustered into disjoint and compact regions. Each cluster is called a functional concept and represents a set of perceptual samples that -despite being scattered in the perceptual space (P)- are similar with respect to their functionality (Q-values). In other words, the agent conceptualizes its environment by its action values in its functional space.

In a discrete perceptual space, each stimuli can be considered as a perceptual sample. In a continuous perceptual space, the samples can be generated by a discretization method. Due to the reality that stimuli corresponding to a concept are scattered in the perceptual space irregularly, the sampling radius is set to a small value; which yields a large number of samples. In other words, sampling is a mapping from the continuous perceptual space to the agent's large discrete state space (S):

$$Sampling : P \to S \tag{2}$$

Each point in the functional space F is called a *functional point* and is represented by a *functional vector*. Multiple states of S might be mapped to a single functional point. Because of being uncertain about the action values during learning, functional points are modeled as fuzzy points in the functional space [5]. For this purpose, each functional point is represented by a vector of action values (Q_s) representing the center of the fuzzy point and a vector of uncertainties (U_s) representing the amount of uncertainty (or the fuzziness) of action values in each dimension of the functional space:

$$\begin{gathered}\Psi : S \to \mathcal{F}\\ \Psi(s) = \langle Q_s, U_s \rangle\\ Q_s = \langle Q(s,a_1), \cdots, Q(s,a_m)\rangle\\ U_s = \langle U(s,a_1), \cdots, U(s,a_m)\rangle\end{gathered} \tag{3}$$

where $\mathcal{F}$ is the space of fuzzy points in F, Ψ is a mapping from the states in the perceptual space to $\mathcal{F}$, $\{a_1, a_2, \cdots, a_m\}$ is the agent's action set, m is the number of actions and s is a state in the perceptual space.

We define the Q-values of the functional points as the expected discounted reward and the centers of the functional points are continuously updated after every agent-environment interaction (equation 1).

The uncertainty of $\Psi(s)$ in dimension i is modeled as a function of action a_i's value in state s in addition to the frequency of being selected in that state ($C_{s,i}$). Variance of a_i's value is represented by $\overline{\sigma}^2_{s,i}$ and is used as a criterion for estimation of the level of uncertainty in the environment.

Assume s and s' are two states in the perceptual space. Then, we have:

$$\begin{gathered}C_{s,i} < C_{s',i}, \overline{\sigma}^2_{s,i} = \overline{\sigma}^2_{s',i} \Rightarrow U(s,a_i) > U(s',a_i),\\ \overline{\sigma}^2_{s,i} < \overline{\sigma}^2_{s',i}, C_{s,i} = C_{s',i} \Rightarrow U(s,a_i) < U(s',a_i).\end{gathered} \tag{4}$$

where C_{s,a_i} is the number of times that a_i is experienced in response to s. It suggests that uncertainty about $Q(s,a_i)$ should be proportional to its variance ($\overline{\sigma}^2_{s,i}$) and inversely proportional to C_{s,a_i}:

$$U(s,a_i) = \frac{\overline{\sigma}^2_{s,i}}{K \times C_{s,i} + 1} \tag{5}$$

where K is a constant and is specified manually to scale the uncertainty measures in a normal range. The uncertainty measure, $U(s,a_i)$, models a combination of uncertainty in the environment and lack of sufficient experience.

4. Functional Concepts

An estimation of a functional concept is formed by clustering the functional points. As we use the softmax action selection policy, the lenght of the vectors does not have any effect on the decision and just the orientation of them is important.

So, to cluster the functional points, firstly, we normalize the functional vectors using the following equation:

$$L = \sqrt{\sum_{k=1}^{m} Q^2(s, a_k)} \ , \ q(s, a_i) = \frac{Q(s, a_i)}{L} \ , \ u(s, a_i) = \frac{U(s, a_i)}{L} \tag{6}$$

By normalizing the functional vectors, we actually transfer all the points on the surface of a unit globe. It means that we ignore the lenght of the vectors and concentrate on the orientation of them to form the functional concepts.

When some states in the perceptual space map to a compact region in the functional space, we can assume that those states belong to a single concept from the view point of the agent's actions. Then, a functional concept FC can be represented by a compact region in the functional space as follows:

$$FC = \{x = \Psi(s) | \|x - C_{FC}\| \leq \varepsilon, \varepsilon > 0\} \tag{7}$$

where $\|.\|$ is the Euclidean norm in the functional space and C_{FC} stands for representative functional point of concept FC. Therefore, C_{FC} is a functional point in the functional space:

$$C_{FC} = \langle Q_{FC}, U_{FC} \rangle \tag{8}$$

Q_{FC} and U_{FC} are the center and uncertainty vector of FC.

In this paper, the functional clusters are formed by using Basic Sequential Algorithmic Scheme (BSAS) [21] on the normalized functional points. In order to cluster, firstly, we need a distance measure. The distance between two functional points is a function of their q and u vectors. The following distance measure is used to cluster functional points:

$$d(f, f') = \sqrt{\sum_{i=1}^{m} d_i(f, f')^2} \tag{9}$$

where m is the size of agent's action set and $d_i(f, f')$ represents the distance between f and f' functional points in the i^{th} dimension. $d_i(f, f')$ is computed as:

$$d_i(f, f') = \frac{|q(s, a_i) - q(s', a_i)|}{(u(s, a_i) + 1)(u(s', a_i) + 1)} \tag{10}$$

where f and f' are the functional points of states s and s' respectively.

Using this distance measure, we expect that the centers of functional clusters to be more affected by the functional points with lower uncertainties. It is compatible with the fact that these points are more reliable for concept representation than the points with higher uncertainties. In addition, the measure satisfies the following desired properties:

(1) Symmetry

$$d_i(f, f') = d_i(f', f).$$

(2) Zero distance conditions

$$d_i(f, f') = 0 \Leftrightarrow q(s, a_i) = q(s', a_i).$$

(3) The effect of points' center distances

$$|q(s,a_i) - q(s',a_i)| < |q(s,a_i) - q(s'',a_i)|, u(s',a_i) = u(s'',a_i) \\ \Rightarrow d_i(f,f') < d_i(f,f'').$$

(4) The effect of points' uncertainties

$$|q(s,a_i) - q(s',a_i)| = |q(s,a_i) - q(s'',a_i)|, u(s',a_i) < u(s'',a_i) \\ \Rightarrow d_i(f,f') > d_i(f,f'').$$

Now we define representative point of functional concept FC; namely $C_{FC} = \langle Q_{FC}, U_{FC} \rangle$. The center of mass of each cluster is defined as the cluster center. The mass of each functional point in each dimension is in inverse proportion to its uncertainty in that dimension. The following formula is used to compute the center of cluster FC in the i^{th} dimension:

$$Q_{FC}(a_i) = \frac{\sum\limits_{\forall s:\Psi(s)\in FC} u(s,a_i)^{-1} \times q(s,a_i)}{\sum\limits_{\forall s:\Psi(s)\in FC} u(s,a_i)^{-1}} \tag{11}$$

It shows that, the cluster center is closer to the points with lower uncertainties.

Uncertainty measure for a cluster can be defined as an inter-cluster distance [11]. We suggest the following measure in which the uncertainty of a cluster in the i^{th} dimension is computed as the average of its members' distance to the cluster center in that dimension:

$$U_{FC}(a_i) = \frac{\sum\limits_{f\in FC} d_i(f, Q_{FC})}{\|FC\|} \tag{12}$$

where $\|FC\|$ denotes the number of the functional concept's members.

5. **Abstract Hierarchy**

A meaningful hierarchy can help the agent to have more abstract representation of its environment. Such a hierarchy can be made based on either "has-a" or "is-a" relation among the concepts. Describing the concepts in terms of action values in the agent's functional space facilitates establishing "is-a" relation among the concepts and to form an abstract hierarchy on them. We will discuss about application of the hierarchy more later.

To achieve this goal, similarity between the concepts in the functional space is used to form more abstract concepts which represent higher levels of the hierarchy. These concepts are more general and cover more samples in the perceptual space. As uncertainty about the action-values is high during the early stages of learning, the concepts may be formed in the higher levels of the hierarchy first.

The similarity between two functional concepts FC_1 and FC_2 is inversely proportional to the distance of their representative points; namely C_{FC_1} and C_{FC_2}. Two concepts are similar if the distance between them is smaller than a threshold:

$$d(C_{FC_1}, C_{FC_2}) \leq \varepsilon, \tag{13}$$

where $d(.,.)$ is the distance function defined in equation 9. If the functional concepts FC_1 and FC_2 are similar, then they can be treated as a members of a higher level cluster as a more abstract functional concept (FC_3). In other words, FC_3 is a higher

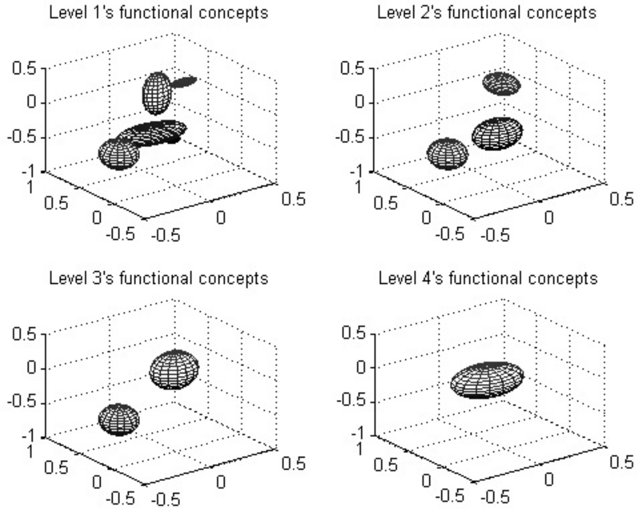

FIGURE 2. The Functional Concepts in the Different Levels of the Hierarchy

level concept in the abstract concept hierarchy which is formed by the similarity between FC_1 and FC_2. This higher level concept forms "is-a" relation with all of the concepts in the lower level that have formed it and inherits its action value from the most compact one.

To obtain the higher level concepts, we treat the current functional concepts as some functional points and cluster them to obtain the new level's functional concepts. We use the BSAS using the defined distance measure and a new threshold which is expected to be higher than the threshold used for the previous level. We also weight the functional concepts with the number of their members for clustering and forming the new level of the hierarchy. This process is repeated for every new level of the hierarchy until there are no new clusters.

Example 5.1. Consider five functional points as $\{f_1, f_2, f_3, f_4, f_5\}$ where:

$$q_{f_1} = \langle 0.1, 0.3, -0.5\rangle \ , \ u_{f_1} = \langle 0.05, 0.1, 0.07\rangle$$
$$q_{f_2} = \langle -0.6, -0.3, -0.1\rangle \ , \ u_{f_2} = \langle 0.12, 0.19, 0.2\rangle$$
$$q_{f_3} = \langle 0.3, 0.5, 0.1\rangle \ , \ u_{f_3} = \langle 0.1, 0.05, 0.05\rangle$$
$$q_{f_4} = \langle -0.1, 0.2, -0.3\rangle \ , \ u_{f_4} = \langle 0.25, 0.2, 0.1\rangle$$
$$q_{f_5} = \langle 0.2, 0.7, -0.1\rangle \ , \ u_{f_5} = \langle 0.1, 0.1, 0.3\rangle$$

These functional points are clustered in an abstract hierarchy. The threshold for different levels are:

$$\varepsilon_1 = 0.3 \ , \ \varepsilon_2 = 0.5 \ , \ \varepsilon_3 = 0.7$$

where ε_i is the threshold for clustering of level i's concepts. The functional concepts of the different levels of hierarchy is shown in Figure 2.

6. Functional Concepts and Knowledge Transfer

One of the main advantages of forming an abstract hierarchy on the functional concepts of an agent is that another agent with different perceptual space can use it to speed up its learning in the same environment. This is due to the fact that functional concepts are independent from the state representation. So, in this section, the notion of functional concept is used to propose an algorithm for transfer learning among heterogeneous agents which share a same action space, environment's dynamic and reward function, but have different state spaces.

Suppose that the first agent (source agent) has finished its learning and its functional concepts and the hierarchy are available for the second agent (target agent) which is at its initial steps of learning. Let the target agent be in its current state which is represented by a functional fuzzy point in the functional space. As the action spaces of two agents are the same, one can compare the similarity of functional point of the current state of target agent with the functional concepts of the source agent. This similarity will give some valuable information for choosing the best action for the current state.

The similarity of a functional point and a functional concept is defined using a distance function:

$$d(FC, f) = \sqrt{\sum_{i=1}^{m} d_i(FC, f)^2} \tag{14}$$

where f is the functional point of state s and FC is a functional concept and:

$$d_i(FC, f) = \frac{|Q_{FC}(a_i) - q_f(a_i)|}{(U_{FC}(a_i) + 1)(u_f(a_i) + 1)}. \tag{15}$$

All functional concepts near to the state's functional point f are candidate concepts and can be used to estimate the real action values for the observed state. It means "f is-a FC" or equivalently "s is-a FC". The value of each action is estimated from the candidate concept with lowest uncertainty in that action value. The details of the knowledge transfer algorithm are as follows:

(1) Form functional concepts and the hierarchy on them using the final Q-values of the source agent. Let H be the number of the levels of hierarchy.
(2) Initialize the Q-values of the target agent with the q-values of the highest level functional concept of the hierarchy.
(3) Initialize the U-values of the target agent with a large positive value.
(4) For every step i of learning of the target agent do:
    (a) Let $M = \underset{j}{max} K(s_t, a_j)$, where s_t is the current state of the agent and $K(s_t, a_j)$ is the number of visits of the pair (s_t, a_j) by the agent.
    (b) If $i < M$ then:
        (i) Normalize the Q-values of the current state (s_t) and form the functional point for the current state (FC).
        (ii) Find the nearest functional concept of level $M - i$ of the hierarchy to the functional point FC, denoted by FH.

```
            (iii) Use the vector of $FH$ and the softmax action selection policy to
                  make a decision and take an action.
      (c) Else
            (i) Use the classical softmax action selection policy to make a deci-
                sion and take an action.
      (d) End of if
      (e) Update the Q and U values of the target agent.
  (5) End of for
```

As indicated in the algorithm, the knowledge transfer algorithm is only applied to the first H steps of the learning where H is the number of the levels of the hierarchy. Based on the number of maximum number of visits of the current state with different actions, the functional concepts of a specific level of the hierarchy is used to make a decision. To make a decision, the functional point of the current state is compared with the functional concepts of the selected level of the hierarchy and the nearest functional concept is used to make a decision. The rest of decision making process is a softmax action selection policy, such as Boltzmann's method [5].

The point is that the functional concepts are just used for decision making and does not change the Q-values of the target agent. As the Q-learning is an off-policy method, the proposed transfer learning method will have the guarantee of convergence.

7. Case Studies

In this section, three simulations are performed in order to analyse the main properties of the proposed method. Here we assume that agents have the same functional but different perceptual spaces. All the simulations are performed twice; firstly, without knowledge transfer and then using the knowledge transfer algorithm. In the simulations without knowledge transfer the Q-values of the target agent is initialized with zero. The results of the simulations are compared to investigate the effectiveness of the proposed algorithm.

7.1. Grid World Problem.

To facilitate analysing the approach, a simple grid world problem is considered first. The grid world problem is one of the classical testbeds in the field of artificial intelligence. The goal of the learning agent is to reach the target place along shortest path without hitting the obstacles; see Figure 3. The agent has three actions which makes the visualization of the functional space and concepts possible. At each step, the agent can choose one of its three actions; moving forward, turning 90° to left or turning 90° to right. The reward function is:

$$\text{Reward}=\begin{cases} -1 & \text{moving without hitting the obstacles or turning} \\ -10 & \text{hitting the obstacles} \\ 100 & \text{reaching the goal} \end{cases}$$

Each episode of learning starts from a random location and finishes when the agent arrives at the goal place. The learning is repeated for 500 episodes and the whole learning process is repeated for 500 epochs. The action selection policy is softmax and the learning parameters are as follows; the learning rate (α) is 0.1,

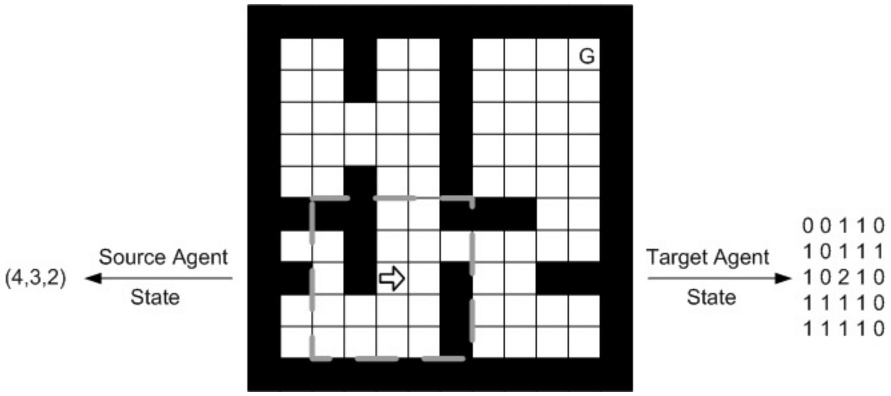

Figure 3. The Two-dimensional Grid Environment with Some Obstacles. The Arrow Shows the Agent'S Location and Heading and the Goal is at the Top-right Corner

the discount factor (γ) is set to 0.9, and the temperature (τ) decreases by the exponential function ($\tau = e^{-n} + 1$) where n is the number of iterations.

Here we have two agents; namely the source and the target agents differing in state representation. The source agent learns first and then its knowledge is used by the target agent to speed up its learning. The source agent's state is its configuration, i.e., a tuple where its first and second elements are its horizontal and vertical locations, and the third element encodes its direction; see Figure 3.

When the source agent's learning is finished, its Q vectors are normalized and then clustered to form the functional concepts (Figure 4). The BSAS is used to cluster the Q vectors. The thresholds for clustering of different levels of hierarchy are: $\varepsilon_1 = 0.01$, $\varepsilon_2 = 0.02$, $\varepsilon_3 = 0.05$, $\varepsilon_4 = 0.1$, $\varepsilon_5 = 0.2$, $\varepsilon_6 = 0.3$, $\varepsilon_7 = 0.5$, $\varepsilon_8 = 0.7$ and $\varepsilon_9 = 0.9$. Some functional concepts of the hierarchy are shown in Figure 5. The number of functional concepts in different levels of the hierarchy are 274, 219, 159, 88, 36, 16, 12, 5, 3 and 1, respectively.

To evaluate the proposed approach for transfer learning, a target agent with different type of sensors is considered. The state of the target agent is represented by an agent-centered 5×5 binary matrix. Each entry of the matrix indicates existence or absence of obstacle in the corresponding grid with one exception; the entry $(3, 3)$ at the center of the matrix indicates the direction of the agent (see Figure 3). With this state representation the grid world problem will be an MDP for the target agent as well.

The functional points, the normalized values and the representative points of the clusters are shown in Figure 4. The functional concepts are clustered in nine levels of a hierarchy. These concepts are transferred to the target agent and that agent employs the proposed approach for learning and decision making. The average rewards per episodes are depicted in Figure 6. This figure and Table 7.1 shows the efficiency of the approach for knowledge transfer between two agents with different perceptual spaces in terms of the learning speed. As the results show the target

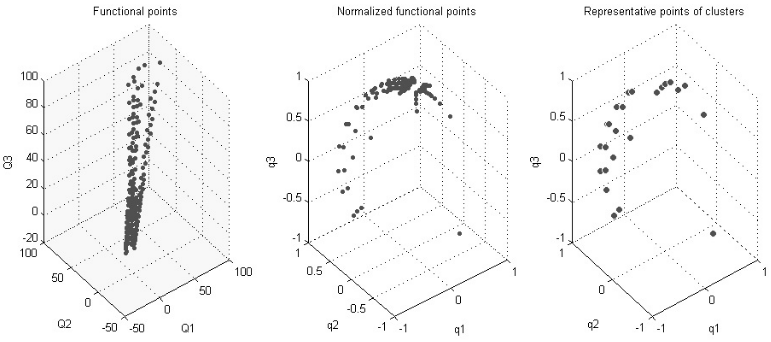

Figure 4. Functional Points, Normalized Values and the Representative Points of the Clusters

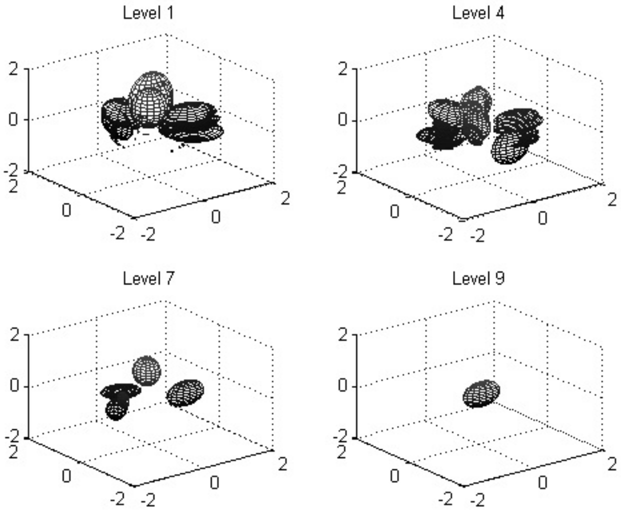

Figure 5. Functional Concepts of Different Levels of the Hierarchy

agent has learned the rewarding policy much faster than when it learns from scratch; which is the goal of knowledge transfer.

7.2. **Taxi Problem.** This problem is adopted from Diettrich's work [2]. The agent controls a taxi on a grid with some obstacles as shown in Figure 7. There are four locations on the grid that are possible passenger pick-up/drop-off locations. In each episode, the passenger must be dropped off at a specific drop-off location. Here, the environment is non-deterministic as well; with the probability of 0.1, the action may have a different result. At each step, the agent can choose one of its six actions; moving upward, downward, left, right, pick up passenger or drop off passenger. The

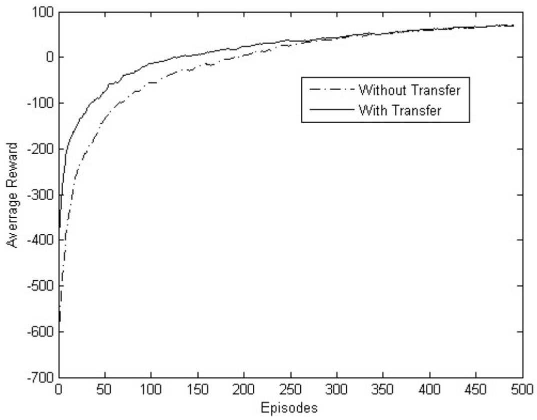

Figure 6. Comparing the Average Rewards of the Target Agent with and without Knowledge Transfer for the Grid World Problem

	Without K.T.	With K.T.
Episodes to reach 40% of max. reward	12	3
Episodes to reach 50% of max. reward	18	5
Episodes to reach 60% of max. reward	32	10
Episodes to reach 70% of max. reward	52	26
Episodes to reach 80% of max. reward	93	53
Episodes to reach 90% of max. reward	195	133

Table 1. Comparing the Learning Speeds of the Target Agent with and without Knowledge Transfer for the Grid World Problem

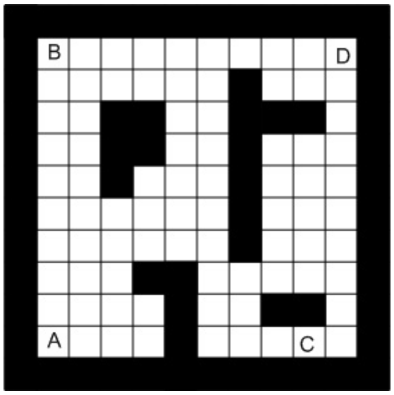

Figure 7. Two-dimensional Grid for Taxi Problem. The Locations for Passengers' Pick up and Drop off are Indicated by A, B, C and D

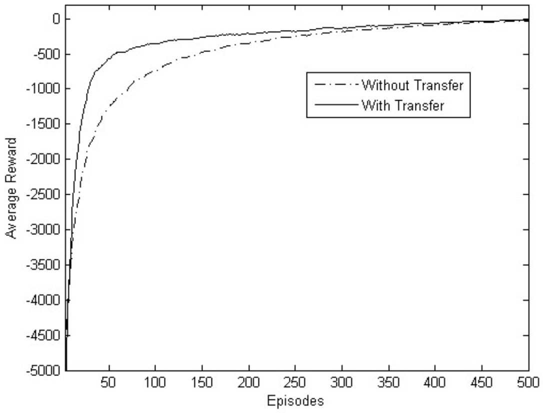

Figure 8. Comparing the Average Reward of the Target Agent with and without Knowledge Transfer for the Taxi Problem

reward function is as follows:

$$\text{Reward}=\begin{cases} -5 & \text{hitting the obstacles} \\ 100 & \text{successful pick up or drop off} \\ -10 & \text{drop off or pick up at wrong location} \\ -1 & \text{otherwise} \end{cases}$$

Each episode of learning finishes when the passenger is picked up and dropped off at the specific location. The learning is repeated for 500 episodes and the whole learning is initialized again for 500 epochs. The action selection policy is softmax and the learning parameters are as follows; the learning rate (α) is 0.1 and the discount factor (γ) is set to 0.9.

The state of the source agent is a tuple consisting three numbers; the first two values are the location indicating its row and column numbers, the second and fourth values are the passenger and destination locations, respectively. Again, we assume that the target agent has different type of sensors. The state of target agent is represented by an agent-centered 5 × 5 binary matrix where each entry of the matrix indicates existence or absence of obstacle in the corresponding grid.

For $\varepsilon_1 = 0.02$, $\varepsilon_2 = 0.03$, $\varepsilon_3 = 0.04$, $\varepsilon_4 = 0.06$, $\varepsilon_5 = 0.08$, $\varepsilon_6 = 0.1$, $\varepsilon_7 = 0.15$, $\varepsilon_8 = 0.2$, $\varepsilon_9 = 0.25$, $\varepsilon_{10} = 0.35$, $\varepsilon_{11} = 0.5$, $\varepsilon_{12} = 0.7$, $\varepsilon_{13} = 0.8$ and $\varepsilon_{14} = 0.9$, the number of functional concepts in different levels of the hierarchy are 886, 743, 542, 346, 219, 162, 85, 45, 28, 13, 7, 4, 2 and 1, respectively. The number of all functional points is 1296. The comparison of average reward with and without knowledge transfer is shown in Figure 8, and the number of episodes to reach a certain level of maximum reward is shown in Table 7.2.

The result clearly shows that the algorithm could efficiently increase the average reward, specially, at the beginning episodes of the learning. It also decreases the number of episodes to reach the maximum reward, which means that the algorithm has a higher convergence speed while using knowledge transfer.

	Without K.T.	With K.T.
Episodes to reach 40% of max. reward	13	10
Episodes to reach 50% of max. reward	19	13
Episodes to reach 60% of max. reward	26	16
Episodes to reach 70% of max. reward	40	21
Episodes to reach 80% of max. reward	71	29
Episodes to reach 90% of max. reward	148	57

Table 2. Comparing the Learning Speeds of the Target Agent with and without Knowledge Transfer for the Taxi Problem

8. Discussion and Conclusion

The notion of transfer learning is almost a new and challenging area in the field of Reinforcement Learning. This research focused on abstracting and representation of knowledge in a new space called functional space. It was discussed that in the functional space, the concepts that are related to the functionality of the agent have suitable and compact representations. In addition, it was targeted that, if the heterogeneity of the agents is due to the difference in their state spaces and the agents have the same action space, then the knowledge can be transferred among them using the functional space.

Each point in the functional space is an action-value vector. A functional concept in the functional space can be understood as a neighborhood area of a vector. In this article, this area was expressed by fuzzy values. In fact, extremely distant points of the same concept in the perceptual space may map to neighboring points in the functional space. A clustering approach was proposed for extracting the concepts and forming an abstract hierarchy based on fuzzy values. The functional concepts and the hierarchy were used as a knowledge transfer tool among heterogeneous agents with different state spaces.

The simulations showed that the representation of concepts in the functional space can provide a proper tool of transfer learning. The results showed significant improvement in the learning average reward especially in the beginning episodes of the learning when using the knowledge transfer algorithm.

The next step of this research is to code ambiguity in the functional space in addition to search for a set of distance measures that codes similarity of functional points better.

References

[1] J. S. Bruner, *Actual minds, possible words*, Harvard University Press, 1987.

[2] T. Dietterich, *Hierarchical reinforcement learning with the MAXQ value function decomposition*, Journal of Artificial Intelligent Research, **13** (2000), 227-303.

[3] K. Driessens, J. Ramon and T. Croonenborghs, *Transfer learning for reinforcement learning through goal and policy parametrization*, In ICML Workshop on Structural Knowledge Transfer for Machine Learning, (2006), 1-4.

[4] W. Fritz, *Intelligent systems and their societies*, In Webpage: http://www. intelligentsystems.com.ar/intsyst/index.htm, January (1997).

[5] G. L. Klir, *Uncertainty and information: foundations of generalized information theory*, John Wiley, Hoboken, NJ (2005).

[6] G. L. Klir, B. Yuan, *Fuzzy sets and fuzzy logic: theory and applications*, Prentice Hall, 1995.

[7] G. Konidaris, A. Barto, *Autonomous shaping: Knowledge transfer in reinforcement learning*, In Proceedings of the 23rd international conference on Machine learning, 2006, 489-496.

[8] A. Lazaric, *Knowledge Transfer in Reinforcement Learning*, PhD thesis, Politecnico di Milano, 2008.

[9] L. Mihalkova, T. Huynh and R. Mooney, *Mapping and revising Markov Logic Networks for transfer learning*, In Proceedings of AAAI Conference on Artificial Intelligence, (2007), 608-614.

[10] H. Mobahi, M. Nili Ahmadabadi, and B. Nadjar Araabi, *A biologically inspired method for conceptual imitation using reinforcement learning*, Applied Artificial Intelligence, **21** (2007), 155-183.

[11] R. A. Mollineda, F. J. Ferri and E. Vidal, *A cluster-based merging strategy for nearest prototype classifiers*, In Proceedings of 15th International Conference on Pattern Recognition (ICPR'00), **2** (2000), 755-758.

[12] G. L. Murphy, *The big book of concepts*, MIT Press, 2004.

[13] S. Pan, J. Kwok and Q. Yang, *Transfer learning via dimensionality reduction*, In Proceedings of 23^{rd} AAAI Conference on Artificial Intelligence, (2008), 677-682.

[14] V. Soni, S. Singh, *Using hormomorphisms to transfer options across continuous reinforcement learning domains*, In Proceedings of 21^{st} AAAI Conference on Artificial Intelligence, (2006), 494-499.

[15] R. S. Sutton, A. G. Barto, *Reinforcement Learning: An Introduction*, MIT Press, Cambridge, MA (1998).

[16] F. Tanaka, M. Yamamura, *Multitask reinforcement learning on the distribution of MDPs*, Transactions of the Institute of Electrical Engineers of Japan, **123(5)** (2003), 1004-1011.

[17] M. Taylor, P. Stone, *Transfer learning for reinforcement learning domains: a survey*, Journal of Machine Learning Research, **10** (2009), 1633-1685.

[18] M. Taylor, G. Kuhlmann and P. Stone, *Autonomous transfer for reinforcement learning*, In Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent systems, **1** (2008), 283-290.

[19] M. Taylor, P. Stone, *Representation Transfer for Reinforcement Learning*, In Proceedings of AAAI Fall Symposium on Computational Approaches to Representation Change during Learning and Development, Arlington, Virginia, (2007), 78-85.

[20] M. Taylor, P. Stone and Y. Liu, *Value function for RL-based behavior transfer: A comparative study*, In Proceedings of the AAAI-05 Conference on Artificial Intelligence, (2005), 880-885.

[21] A. Thedoridis and K. Koutroumbas, *Pattern Recognition*, Elsevier Academic Press, Second Edition, 2003.

[22] L. Torrey and J. Shavlik, *Transfer learning*, In Soria, E., Martin, J., Magdalena, R., Martinez, M., and Serrano, A., editors, Handbook of Research on Machine Learning Applications, IGI Global, 2009, 242-264.

[23] C. J. Watkins, *Learning from Delayed Rewards*, Ph.D. thesis, Cambridge University, 1989.

[24] C. J. Watkins and P. Dayan, *Q-learning*, Machine Learning, **8** (1992), 279-292.

[25] A. Wilson, A. Fern, S. Ray and P. Tadepalli, *Multitask reinforcement learning: A hierarchical Bayesian approach*, In Proceedings of the 24^{th} International Conference on Machine Learning, (2007), 1015-1022.

[26] M. Zentall, M. Galizio and T. S. Critchfied, *Categorization, concept learning and behavior analysis: an introduction*, The Exprimental Analysis of Behavior, **3** (2002), 237-248.

A. Mousavi*, Control and Intelligent Processing Center of Excellence, School of Electrical and Computer Engineering, University of Tehran, Tehran, Iran
E-mail address: am.mousavi@ece.ut.ac.ir

M. Nili Ahmadabadi, Control and Intelligent Processing Center of Excellence, School of Electrical and Computer Engineering, University of Tehran, Tehran, Iran and School of Cognitive Science, Institute for Research in Fundamental Sciences (IPM), Tehran, Iran
E-mail address: mnili@ut.ac.ir

H. Vosoughpour, Control and Intelligent Processing Center of Excellence, School of Electrical and Computer Engineering, University of Tehran, Tehran, Iran
E-mail address: hamide@gmail.com

B. N. Araabi, Control and Intelligent Processing Center of Excellence, School of Electrical and Computer Engineering, University of Tehran, Tehran, Iran and School of Cognitive Science, Institute for Research in Fundamental Sciences (IPM), Tehran, Iran
E-mail address: araabi@ut.ac.ir

N. Zaare, Control and Intelligent Processing Center of Excellence, School of Electrical and Computer Engineering, University of Tehran, Tehran, Iran
E-mail address: zare_narjes2010@yahoo.com

*Corresponding author