퍼지 및 지능시스템학회 논문지 2003, Vol. 13, No. 6, pp. 635-642

강화학습의 학습 가속을 위한 함수 근사 방법

Function Approximation for accelerating learning speed in Reinforcement Learning

이영아 · 정태충
YoungAh Lee, TaeChoong Chung

경희대학교 컴퓨터공학과

요 약

강화학습은 제어, 스케쥴링 등 많은 응용분야에서 성공적인 학습 결과를 얻었다. 기본적인 강화학습 알고리즘인 Q-Learning, $TD(\lambda)$, SARSA 등의 학습 속도의 개선과 기억장소 등의 문제를 해결하기 위해서, 여러 함수 근사방법(function approximation methods)이 연구되었다. 대부분의 함수 근사 방법들은 가정을 통하여 강화학습의 일부 특성을 제거하고 사전지식과 사전처리가 필요하다. 예로 Fuzzy Q-Learning은 퍼지 변수를 정의하기 위한 사전 처리가 필요하고, 국소 최소 자승법은 훈련 예제집합을 이용한다. 본 논문에서는 온-라인 퍼지 클러스터링을 이용한 함수 근사 방법인 Fuzzy Q-Map을 제안한다. Fuzzy Q-Map은 사전 지식이 최소한으로 주어진 환경에서, 온라인으로 주어지는 상태를 거리에 따른 소속도(membership degree)를 이용하여 분류하고 행동을 예측한다. Fuzzy Q-Map과 다른 함수 근사 방법인 CMAC와 LWR을 마운틴 카 문제에 적용하여 실험 한 결과, Fuzzy Q-Map은 훈련예제를 사용하지 않는 CMAC보다는 빠르게 최고 예측율에 도달하였고, 훈련 예제를 사용한 LWR보다는 낮은 예측율을 보였다.

Abstract

Reinforcement learning got successful results in a lot of applications such as control and scheduling. Various function approximation methods have been studied in order to improve the learning speed and to solve the shortage of storage in the standard reinforcement learning algorithm of Q-Learning. Most function approximation methods remove some special quality of reinforcement learning and need prior knowledge and preprocessing. Fuzzy Q-Learning needs preprocessing to define fuzzy variables and Local Weighted Regression uses training examples. In this paper, we propose a function approximation method, Fuzzy Q-Map that is based on on-line fuzzy clustering. Fuzzy Q-Map classifies a query state and predicts a suitable action according to the membership degree. We applied the Fuzzy Q-Map, CMAC and LWR to the mountain car problem. Fuzzy Q-Map reached the optimal prediction rate faster than CMAC and the lower prediction rate was seen than LWR that uses training example.

Key words : Q-learning, 퍼지 클러스터링, 소속도, CMAC, LWR

1. 소 개

교사학습(supervised learning)은 상태공간의 일부분에서 또는 전역적인 최적의 제어 전략이 알려진 상태에서 훈련 예제에 함축되어있는 사상(mapping)을 오프라인으로 학습한다. 반면 강화학습(reinforcement learning)은 사전지식이 없어도 학습이 가능하고, 환경(environment)과의 반복적인 시도와 오류(trial and error)를 통해서 최적에 가까운 가치 함수(value function)를 학습한다. 강화학습은 속도는 느리지만 훈련예제와 사전지식이 충분히 주어지지 않은 경우, 온라인 학습이 가능한 방법이다.

환경의 각 상황을 표현하는 상태(state)는 학습에 영향을 미치는 요소들(features)로 이루어진다. 상태는 이산 상태(discrete state)와 연속 상태(continuous state)로 나눌 수 있는데, 강화학습의 기본적인 알고리즘인 Q-Learning, SARSA등은 이산 상태 공간에서 학습한다. 하지만 제어 전략(control policy)을 학습하는 많은 강화학습 문제들은 연속 상태와 연속 행동을 갖고 있어서 기본 강화학습 알고리즘을 그대로 이용할 수 없다. 이 문제를 해결하기 위하여 연속상태와 연속 행동을 이산화(discretization)하는데, 의미 있는 최적의 전략들을 표현할 수 있도록 이산화 하는 것이 중요하다. 모든 의미 있는 상태-행동 쌍의 전략들을 얻기 위해서, 연속적인 상태공간을 세밀하게 나눈다면 상태 공간의 크기가 거대해진다. 이 경우, 기본 강화학습 알고리즘은 모든 상태-행동 쌍을 저장하고 반복적으로 경험해야 하므로 매우 긴 학습 시간과 많은 저장 공간이 필요하다.

연속적인 상태공간에서 효율적으로 학습하기 위해서 강화학습은 유사한 상태들을 일반화 시키는 함수근사(function approximation) 방법을 이용한다[1,2]. 많은 연구에서 함수근사 방법으로서 전 방향 또는 역방향 신경망(Feedforward or Backpropagation Neural Network)[3], 자기 형상화 지도(Self-Organizing features Map: SOM), CMAC(cerebella

접수일자 : 2003년 9월 16일
완료일자 : 2003년 12월 11일

model articulation controller)[4,5,6]와, Q-Learning을 퍼지 환경으로 확장하는 연구[3,7,8,9,10,11]가 있었다.

강화학습은 보상 함수(reward function)를 이용하여 학습 에이전트가 가장 최근에 선택한 행동이 얼마만큼의 가치가 있는지를 평가한다. 보상함수가 결과로 내는 값은 선택한 행동에 의하여 도달한 다음 상태에 따라서 다르게 지정될 수 있다. 다음 상태가 목표 상태(goal state)와 같이 가치가 있는 경우에는 보상값(reward)이 지정되고, 다음상태가 치명적인 상태이거나 목표가 아닌 경우에는 적절한 벌금(penalty)이 주어진다. Q-Learning과 같은 기본 알고리즘은 사전지식이 없는 상태에서 학습을 시작하므로, 학습 에이전트가 보상값을 받는 목표상태(goal state)에 도달할 때까지 임의로 행동을 선택해야 한다. 학습 속도의 개선을 위해서, 강화학습 알고리즘인 HEDGER[6,12]과 JQL[6]은 사전처리로서 교수 예제(teaching instances)를 학습하거나 사용자의 조언을 이용해서 경험할 궤적(trajectory)을 선택하기도 한다[13]. HEDGER와 JQL은 모두 국소 최소 자승법(LWR, linearly Weighted Regression)[6]을 기초로 한다. LWR은 훈련 예제를 행렬로 저장하고, 각 질의(query)에 대한 예측은 거리에 따른 가중치가 큰 예제들로부터 구한다. 선형 회귀(linear regression)는 전역적인 함수를 구하는 반면 LWR은 질의를 해결하는 국소적인 함수를 구하는 것이다. 사전지식을 갖고 사전처리를 수행하는 알고리즘들은 다른 강화학습 시스템과 비교했을 때 최적에 수렴하는 속도가 빠르고, 테스트 집합 'test set)에 대해 예측율이 높아진다. 하지만 상태공간에서 교수 예제와 적절한 훈련 예제의 집합을 만드는 작업은 쉽지 않다.

본 논문에서는 퍼지 클러스터링을 이용한 함수 근사방법인 Fuzzy Q-Map을 소개한다. Fuzzy Q-Map은 거대한 상태공간의 문제를 해결하고, 훈련 예제의 도움 없이 최소한의 도메인에 대한 지식이 주어지는 문제에서 1-step Q-Learning의 학습 속도 개선을 시도하였다. Fuzzy Q-Map은 소속도(membership degree)를 이용해서 질의 상태(query)를 분류하고, 적합한 행동을 선택한다. 소속도는 질의 상태와 각 클러스터 중심사이의 거리를 정규화(normalization)한 것이다. 질의 상태는 소속도에 따라 여러 클러스터에 소속될 수 있고, LWR과 같이 거리가 가까운 클러스터들의 전략이 예측에 많은 영향을 준다. Fuzzy Q-Map에서는 승자 클러스터(winner) 중심과 Q값은 TD 에러(Temporal difference error)와 소속도를 이용해서 갱신하였다. Fuzzy Q-Map을 마운틴 카 문제에 적용하고, Smart의 논문에서 제시한 실험결과를 참조하여 비교한 결과, CMAC의 결과보다는 최적에 수렴하는 속도가 빠르고, 훈련 예제를 이용한 LWR보다는 낮은 예측율을 보였다.

본 논문의 구성은 다음과 같다. 2장에서 강화학습의 기본 알고리즘인 Q-Learning과 $Q(\lambda)$를 기술하고, 여러 함수근사 방법을 설명한다. 3장에서는 본 논문에서 제안한 Fuzzy Q-Map을 설명한다. 4장에서는 실험 결과를 보인다. 5장에서는 향후 연구 과제를 제시한다.

2. 관련연구

2.1 강화학습 알고리즘

2.1.1 Q-Learning

Watkins가 제안한 Q-Learning[4]은 학습 에이전트와 환경과의 상호작용으로부터 상태-행동 쌍들(state-action pairs)의 Q함수(Q-function)를 학습한다. Q값 $Q(s,a)$는 상태 s에서 선택한 행동 a가 목표(goal)에 도달하는데 기여한 정도를 나타낸다. Q 값은 처음에는 랜덤하게 주어지고, 온라인으로 환경을 반복해서 경험할 때 점차적으로 최적의 값에 가까워진다. 근사값은 다음 식 (1)에 의해서 갱신된다. 식 (1)을 이용한 Q-Learning을 1-step Q-Learning이라 한다.

$$Q(s_t,a_t) \leftarrow Q(s_t,a_t) + \alpha(r_{t+1} + \gamma \max_{a'} Q(s_{t+1},a') - Q(s_t,a_t)) \quad (1)$$

연속적인 상태공간인 경우, 식 (1)에서 보여주듯이 현재 경험한 상태-행동 쌍의 Q값만을 갱신하므로 1-step Q-Learning의 학습속도는 느리고, lookup table의 크기는 매우 거대해진다.

2.1.2 $Q(\lambda)$

Watkins의 Q-Learning 알고리즘은 상태에 대한 가치 함수(value function)를 반복적으로 학습한다. 학습을 시작할 때, 각 상태의 함수 값을 랜덤하게 정하고, 갱신 식 (1)에 따라 가장 최근에 경험한 상태의 값이 갱신된다.

Q-Learning의 일반적인 형태인 $Q(\lambda)$[4]는 가장 최근에 방문한 상태뿐만 아니라 모든 상태의 값을 적합도(eligibility) $e_t(s_t)$에 따라 다음과 같이 갱신한다.

$$Q_{t+1}(s,a) = Q_t(s,a) + \alpha \delta_t e_t(s,a), \quad (2)$$

$$\delta_t = r_{t+1} + \gamma \max_{a'} Q(s_{t+1},a') - Q(s_t,a_t) \quad (3)$$

각 상태의 적합도는 다음 식 (4)에 따라 시간 단위별로 갱신된다.

$$e_t(s) = I_{ss_t} \cdot I_{aa_t} + \begin{cases} \gamma\lambda e_{t-1}(s,a) & \text{if } Q_{t-1}(s_t,a_t) = \max_a Q_{t-1}(s_t,a), \\ 0 & otherwise \end{cases} \quad (4)$$

적합도의 갱신은 두 단계로 이루어진다. 먼저, Q 값과 상관없이 탐험을 위해 행동을 선택했다면 0으로, 아니면 $\gamma\lambda$에 의해서 감소한다. γ는 할인율(discount factor)이고, λ는 감소율(decay parameter)이다. 그 다음 단계로, 현재 상태-행동 쌍인가에 따라서 1만큼 증가한다. I_{xy}는 식별 함수(identity-indicator function)로서 x와 y의 값이 같으면 1, 아니면 0이다.

2.1.3 FQL(Fuzzy Q-Learning) 알고리즘

FQL(Fuzzy Q-Learning)[9,10,11]은 Glorennce가 Watkins의 Q-Learning을 바탕으로 개발한 학습방법이다. FQL은 Q 함수를 근사화하기 위하여 FIS(Fuzzy Inference System)를 이용하는데, 퍼지 규칙(fuzzy rule)의 결론부를 강화학습의 보상값을 이용하여 조정하고, 조건부는 사전 정보에 의하여 상태공간을 퍼지 입력변수로 코딩한다. 각 규칙은 행동들의 경쟁을 위하여 각 행동의 Q값을 관리한다. 규칙의 형태는 다음과 같다. 규칙 i에서 가능한 행동 $a[i,j]$들이 J개이고 각각 Q 값 $q[i,j]$를 기억한다. S_i는 퍼지 레이블(fuzzy label)이다.

if x is S_i then $a[i,1]$ with $q[i,1]$
or $a[i,2]$ with $q[i,2]$
.....
or $a[i,J]$ with $q[i,J]$

함수 $x \rightarrow a_i(x)$은 입력 상태 x의 규칙 i에 대한 진리값으로 [0,1]사이의 값을 갖는다. FIS의 N개의 규칙이 추론하는 행동 $a(x)$은 다음과 같다.

$$a(x)=\frac{\sum_{i=1}^{N} a_i(x)\times a_i}{\sum_{i=1}^{N} a_i(x)} \tag{5}$$

추론된 행동 a의 Q값은 다음 식 (6)과 같다. i^+은 EEP(Exploration/Exploitation Policy)에 의하여 규칙 i에서 선택된 행동이고, $q[i,i^+]$은 규칙 i에서 선택된 행동 i^+의 Q값을 표시한다.

$$Q(x,a)=\frac{\sum_{i=1}^{N} a_i(x)\times q[i,i^+]}{\sum_{i=1}^{N} a_i(x)} \tag{6}$$

상태 x에 대한 평가값은 다음 식 (7) 같다. i^*은 규칙 i에서 최대의 Q값을 갖는 행동이다.

$$V(x)=\frac{\sum_{i=1}^{N} a_i(x)\times q[i,i^*]}{\sum_{i=1}^{N} a_i(x)} \tag{7}$$

Q값은 식 (1),(5),(6),(7)를 이용하여 갱신된다. 식(8)에서 $\triangle Q$는 $\triangle Q=r+\gamma V(y)-Q(x,a)$로서 에러를 의미하며, ε은 학습율이다.

$$\triangle q[i,i^+]=\varepsilon\times\triangle Q\frac{a_i(x)}{\sum_{i=1}^{N} a_i(x)} \tag{8}$$

FQL에서는 학습 속도를 높이기 위해서 행동의 적합도(eligibility) $e[i,j]$을 다음과 같이 정의하여 식 (8)를 수정하였다.

$$e[i,j]=\begin{cases}\lambda\gamma e[i,j]+\dfrac{a_i(x)}{\sum_{i=1}^{N} a_i(x)} & \text{if} \quad j=i^+\\ \lambda\gamma e[i,j] & otherwise\end{cases} \tag{9}$$

$$\triangle q[i,i^+]=\varepsilon\times\triangle Q\times e[i,j] \tag{10}$$

FQL은 상태공간을 퍼지 입력변수로 코딩하기 위한 사전 정보와 사전 처리가 필요하다.

2.2 함수 근사 방법(function approximations)

2.2.1 신경망(Neural Nets)

신경망은 많은 강화학습 문제에서 성공적으로 함수 근사를 했지만, 블랙박스이기 때문에 예측한 결과에 대한 설명이 불가능하다. 또한 신경망은 간섭(interference)문제를 가지고 있다. 상태공간의 일부분에서 학습을 한 후 다른 지역으로 옮겨가서 학습을 계속했을 때, 학습한 함수(value function)가 새로운 지역에 효과적이지 않다면 함수는 새로운 훈련 데이터에 맞게 바뀌어야 한다. 이런 현상은 전에 학습한 지식이 파괴되므로 파괴적인 간섭(destructive interference)[6]이라고 한다.

2.2.2 CMAC(Cerebellar Model Articulation Controller)

Sutton의 CMAC[4,5,6]는 sparse coarse-coded memory의 한 방법으로 여러 강화학습 응용문제에 광범위하게 사용된다. CMAC는 상태공간이 부분적으로 중첩되는 여러 개의 타일링(tiling)으로 구성되어 있다. '상태-행동' 형태의 입력이 들어오면 그 상태-행동 쌍을 포함하는 타일들(tiles)을 활성화시키고, 활성화된 타일의 값의 합이 해당 상태-행동 쌍의 Q값이 된다. CMAC의 구조와 기본적인 연산은 다음과 같다.

타일 구조: CMAC는 다음과 같이 N_T개의 타일링 집합으로 구성된다.

$$T_i,\ i=1,\ldots.N_T$$

각 타일링 T_i은 상태공간을 나누는 N_{T_i}개의 타일들로 이루어진다.

각 타일 f_{ij}는 가중치(weight) w_{ij}과 적합도(eligibility) e_{ij}를 갖는다.

타일의 선택: 각 타일링 T_i에 대하여, 상태(x_q)와 행동(a_q)으로 이루어진 질의 (x_q, a_q)를 포함하는 타일 f_{ij}이 활성화된다. 활성화된 모든 타일들의 집합 $F(x_q, a_q)$는 다음과 같다.

$$F(x_i,a_i)=f_{ij}\in T_i(x_q,a_q)\in f_{ij} \tag{11}$$

질의 평가: 질의에 대한 Q값은 $F(x_q, a_q)$에 포함된 타일들의 가중치 합이다.

$$Q'(x,a)=\sum_{f_{ij}\in F(x,a)} w_{ij} \tag{12}$$

***TD(λ)*의 갱신**: CMAC의 모든 타일의 가중치 w_{ij}는 다음과 같이 갱신된다.

$$\triangle w_{ij}=\alpha(r_t+\gamma Q'_{t+1}-Q'_t)e_f \qquad \forall f_{ij}\in CMAC \tag{13}$$

적합도 갱신: CMAC의 모든 타일들의 적합도(eligibility)는 다음과 같이 갱신된다.

$$e_{ij}\leftarrow\begin{cases}\dfrac{1}{|F(x_q,a_q)|} & \text{if} \quad f\in F(x_q,a_q)\\ \lambda\gamma e_{ij} & otherwise\end{cases} \tag{14}$$

TD(λ)와 적합도의 갱신은 CMAC의 타일의 수만큼 이루어진다. 타일과 타일링의 수가 많을수록 일반화된 좋은 결과를 얻을 수 있지만 기억장소도 비례해서 증가한다.

CMAC는 연속 상태공간의 문제를 해결하고, 훈련 예제 집합을 이용하지 않고 학습하지만 좋은 성능을 보인다.

2.2.3 국소 최소 자승법(Locally Weighted Regression, LWR)

국소 최소 자승법(Locally Weighted Regression, LWR)[6]은 질의와 가까운 훈련예제만을 바탕으로 국소적인 선형 모델을 만든다. 선형회귀(Linear regression)는 전체 훈련 데이터 집합의 예측 에러를 최소화하도록 선형 모델을 맞추지만, LWR은 거리에 따른 가중치를 적용하여 국소적인 함수를 구한다. LWR 학습은 정답이 주어진 모든 학습 예제를 행렬로 저장한다. 다음 알고리즘은 주어진 질의에 대하여 LWR이 어떻게 예측을 하는가를 보여준다.

입력: 훈련 예제 집합 $(\overrightarrow{x_1}, \overrightarrow{y_1}), (\overrightarrow{x_2}, \overrightarrow{y_2}), \ldots (\overrightarrow{x_n}, \overrightarrow{y_n})$
질의 $\overrightarrow{x_q}$
대역폭 h
출력: 예측결과 $\overrightarrow{y_q}$

단계 1: 그림 1과 같이 각 $\overrightarrow{x_i}$ ($\overrightarrow{x_i} = (x_i^1, x_i^2, \ldots x_i^p)$)을 행으로 하는 행렬 A를 만든다.

단계 2: 그림 1과 같이 각 $\overrightarrow{y_i}$ ($\overrightarrow{y_i} = (y_i^1, y_i^2, \ldots y_i^q)$)을 행으로 하는 행렬 b를 만든다.

$$A = \begin{pmatrix} x_1^1 & x_1^2 \ldots x_1^p & 1 \\ x_2^1 & x_2^2 \ldots x_2^p & 1 \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ x_n^1 & x_n^2 \ldots x_n^p & 1 \end{pmatrix}, \quad b = \begin{pmatrix} y_1^1 & y_1^2 \ldots & y_1^q \\ y_2^1 & y_2^2 \ldots & y_2^q \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ y_n^1 & y_n^2 \ldots & y_n^q \end{pmatrix}$$

그림 1. 훈련 예제 집합을 저장한 행렬
Fig. 1. Matrixes saving training samples

단계 3: 각 훈련 예제 ($\overrightarrow{x_i}, \overrightarrow{y_i}$)에 대하여 단계 4부터 단계 8까지 수행한다.

단계 4: 질의와 $\overrightarrow{x_i}$의 거리 d_i을 구한다.

단계 5: 거리 d_i에 대역폭 h을 이용한 커널 함수(kernel function)를 적용한다. 커널 함수는 각 예제에 대하여 거리에 따른 가중치를 부여한다. 대역폭 h는 함수의 완만함(smoothness)을 지정하는 파라미터이다. h의 값이 크면 전역 함수가 만들어져서 모든 훈련 예제들이 질의에 영향을 주고, 작은 h는 질의와 가장 가까운 예제들 만이 예측에 영향을 준다. 커널 함수로서 다음 식 (15)의 가우시안(Gaussian) 함수가 대표적으로 이용된다.

$$k_i = e^{-(\frac{d_i}{h})^2} \tag{15}$$

단계 6: 행렬 A와 b의 i번째 행과 k_i을 곱한다.

단계 7: 수정된 행렬 A와 b를 이용하여 선형 회귀를 수행한다.

단계 8: $\overrightarrow{y_q} = f(\overrightarrow{x_q})$을 구한다.

LWR은 각 입력된 질의마다 새로운 국소 모델을 만든다. Smart와 Kaebling은 LWR을 기반으로 한 강화학습 알고리즘인 HEDGER[6,12]와 JQL[6]을 제안하였다. HEDGER와 JQL은 LWR과 같이 훈련 예제를 이용하므로 최적의 성능에 빠르게 도달하는 장점이 있다. 하지만 훈련 예제 집합을 만드는 것은 예제들이 최적이 아니더라도 어려운 작업이다.

3. Fuzzy Q-Map의 소개

본 논문에서는 퍼지 클러스터링의 소속도(membership degree)를 이용하여 Q-함수를 근사화하는 Fuzzy Q-Map을 제안한다. 실세계의 많은 제어문제들은 상태와 행동이 연속값(continuous value)으로 표현되고, 훈련데이터에 대한 분류정보 또는 분포가 사전에 주어지지 않으며, 유사한 데이터 집합사이의 경계가 명확하지 않으므로 함수 근사방법으로 비교사 학습 방법(unsupervised learning)인 퍼지 클러스터링이 적합하다고 보았다.

Fuzzy Q-Map의 바탕이 된 FCM(Fuzzy C Means) 알고리즘은 다음과 같다.

3.1 FCM(Fuzzy C Means)알고리즘

FCM(Fuzzy C Means)알고리즘[7,14]은 u_k ($k = 1, 2, \ldots K$)로 정의된 K개의 데이터를 c개의 퍼지 클러스터로 분할하고, 식 (16)의 목적함수(objective function) J를 최소화하도록 클러스터의 중심을 찾는다. FCM은 K-means 알고리즘에 분할의 fuzziness를 증감시키는 파라미터 $q \in (1, \infty)$를 추가한 것으로, 각 데이터가 여러 클러스터에 소속될 수 있다. 정규 데이터(normal data)에 대해서는 일반적으로 $q = 2.0$이 사용된다. 식 (16)에서 M은 훈련 집합의 각 데이터가 각 클러스터에 속하는 멤버쉽 정도를 저장하는 $c \times K$ 크기의 행렬이다.

$$J(M, c_1, c_2, \ldots, c_c) = \sum_{i=1}^{c} \sum_{k=1}^{K} m_{ik}^q d^2_{ik} \tag{16}$$

M: 소속도 행렬 q : fuzziness c_i : 퍼지 클러스터 i의 중심
$d_{ik} = |u_k - c_i|$:데이터 u_k로부터 클러스터 i의 중심 c_i까지의 유클리드 거리
m_{ik} : 클러스터 i에 대한 데이터 u_k의 소속도

목적함수 J가 최저한도에 도달하기 위해서 다음 식 (17)과 (18)를 만족하여야 한다. 소속도 m_{ik}은 [0,1]사이의 값을 갖고, 각 데이터에 대한 모든 클러스터 소속도의 합은 1이다. 각 클러스터의 중심은 식 (18)에 의하여 갱신된다.

$$m_{ik} = \frac{1}{\sum_{j=1}^{c} (\frac{d_{ik}}{d_{jk}})^{2/(q-1)}}, \qquad \sum_{j=1}^{c} m_{ji} = 1 \tag{17}$$

$$c_i = \frac{\sum_{k=1}^{K} m_{ik}^q u_k}{\sum_{k=1}^{K} m_{ik}^q} \tag{18}$$

FCM은 각 퍼지 클러스터의 중심을 갱신하기 위해서 K개 데이터의 소속도를 저장한 행렬 M을 관리해야 한다. 그러나 Q-Learning에 사용되는 훈련 데이터 집합은 상태공간의 크기에 비례해서 커지므로 lookup table과 마찬가지로 소속도 행렬 M을 관리하기 어렵다.

3.2 Fuzzy Q-Map의 구조와 알고리즘

Fuzzy Q-Map은 소속도를 이용하여 유사한 상태들을 군집하여 lookup table의 크기를 줄이고, 승자 퍼지 클러스터와 가까운 퍼지 클러스터들이 제안하는 행동들도 소속도 만큼 참조하여 학습하므로 학습 속도가 빨라진다.

Fuzzy Q-Map은 각 퍼지 클러스터의 중심과 행동들의 Q 값을 기억하는 구조이다. 상태 공간의 분할을 선행처리 하는 경우에는 도메인을 잘 아는 사용자의 사전지식이 필요하고 분할이 고정 된다. 반면 Fuzzy Q-Map은 온라인으로 Q-Learning을 진행하면서 각 퍼지 클러스터의 중심 값을 새로운 경험에 적응하므로 사전처리가 필요 없고, 분할은 조금씩 변하게 된다.

Fuzzy Q-Map에서 각 클러스터의 중심 값은 규칙의 조건부에 해당하는데, 유사한 상태들을 대표한다. 최종 학습된 결과를 담고 있는 Fuzzy Q-Map은 상태공간을 지역적으로 분석한 결과이고, 별도의 복잡한 규칙 추출과정이 필요 없다.

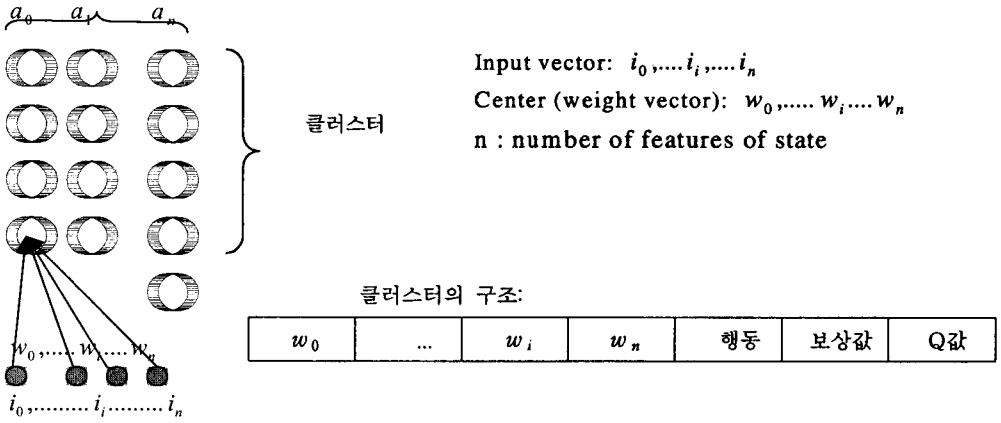

그림 2. Fuzzy Q-Map의 구성
Fig. 2. The structure of Fuzzy Q-Map

Fuzzy Q-Map은 그림 2와 같이 2차원으로, 행의 개수는 사용자가 지정한 클러스터의 개수이고, 열의 개수는 상태공간에서 가능한 행동의 수이다. 그리고 강화학습을 하는 동안 여러 에피소드들(임의의 상태에서 출발하여 목표 상태에 도달할 때까지의 경로)을 경험해야 하는데, 목표상태는 한 에피소드의 끝으로서 다음 상태로 전이되지 않으므로 군집된 다른 상태와 같이 다룰 수 없다. 그러므로 목표상태를 중심으로 갖는 클러스터를 추가하여, Fuzzy Q-Map을 구성하는 노드의 수는 (클러스터의 수 * 가능한 행동의 수) + 1이 된다.

각 퍼지 클러스터는 그림 2와 같이 중심과 행동, 그 행동을 수행했을 때 받은 보상값, t시점에서의 Q값을 기억하고 있다. 상태는 n차원 벡터이고, 현재 상태 $i_0, i_1, ...i_n$와 각 퍼지 클러스터의 중심 $w_0, w_1, ...w_n$과의 유클리드 거리를 측정하여 소속도를 구한다. 소속도는 행동의 선택, Q값 참조, 클러스터의 중심과 Q값의 갱신에 사용된다.

제안한 Fuzzy Q-Map을 이용한 Q-Learning을 정리하면 다음과 같다.

단계 1: Fuzzy Q-Map의 각 클러스터 중심을 랜덤하게 초기화한다. 각 클러스터의 보상값과 Q값은 0으로 초기화한다.

단계 2: 입력 상태 u_t를 랜덤하게 선택한다. t는 하나의 상태를 처리하는 시점을 표시하며 지금까지 훈련에 사용된 상태의 개수이기도 하다.

훈련 데이터 집합은 에피소드들의 집합으로서 상태공간을 탐험하는 과정에서 만나게 되는 상태들로 이루어져 있다. u_t는 한 에피소드를 구성하는 t 시점의 입력 상태이다.

단계 3: u_t가 각 클러스터 i에 속하는 소속도 m를 구한다. u_t에 대한 m ($1 \le i \le c$) 의 합은 1이다. q는 fuzziness의 파라미터로서 사용자가 입력한다.

$$m = \frac{1}{\sum_{j=1}^{c} (\frac{d}{d_{jt}})^{2/(q-1)}}, \qquad \sum_{i=1}^{c} m = 1 \tag{19}$$

단계 4: 상태 u_t에서 $\varepsilon-greedy$ 정책을 이용하여 행동 a_t을 선택 한다.

최적의 행동 a_t^*은 다음과 같이 식 (20)에 의하여 계산한다. 각 클러스터 i가 제안하는 행동 a_t^i을 소속도 m만큼 참조한다. a_t^i는 각 클러스터에서 가능한 행동 중 Q값이 가장 큰 행동이다. c_i는 각 클러스터의 중심이다.

$$a_t^* = \frac{\sum_{i=1}^{c} m_{ik} \cdot a_t^i}{\sum_{i=1}^{c} m_{ik}} = \sum_{i}^{c} m_{ik} \cdot a_t^i \tag{20}$$

$$a_t^i = \max_{\arg a} Q(c_i, a), \qquad a \in A \qquad A: action\ set$$

u_t에서 선택한 행동 a_t의 평가값인 Q값은 다음과 같이 계산된다.

$$Q(u_t, a_t) = \sum_{i=1}^{c} m \cdot qval(a_t) \tag{21}$$

$qval(a_t)$: 각 클러스터에서 행동 a_t에 대한 Q 값

상태 u_t에서 최대의 Q값은 다음과 같이 각 클러스터의 최대 Q값과 소속도로부터 계산한다.

$$f(u_t) = \sum_{i=1}^{c} m \cdot \max\ Q(c_i, a), \ a \in A \tag{22}$$

단계 5: 선택한 행동 a_t를 수행한 후, 환경으로부터 다음 상태 u_{t+1}와 보상값 r_{t+1}를 얻는다. 식 15를 이용하여 소속도가 가장 높은 승자 클러스터의 Q값을 갱신한다. β는 Fuzzy Q-Map의 학습률로서 0.5로 초기화 되고, 식 (26)과 같이 t에 따라 감소한다.

$$Q(u_t, a_t) = Q(u_t, a_t) + \beta(r_{t+1} + \gamma f(u_{t+1}) - Q(u_t, a_t)) \tag{23}$$

단계 6: 입력데이터 u_t에 대한 소속도가 가장 높은 Fuzzy Q-Map의 승자 클러스터의 중심 c_w^t와 Q값을 갱신한다. 소속도 행렬을 이용하여 각 클러스터의 중심을 갱신하는 FCM과 다르게 Fuzzy Q-Map은 새로운 입력과 기존의 값의 에러와 소속도를 이용하여 승자클러스터의 중심을 갱신한다.

$$c_w^t = c_w^t + (u_t - c_w^t) \cdot m_{wt} \cdot \beta \tag{24}$$

$$Q^t(c_w^t, a_t) = Q^t(c_w^t, a_t) + (Q^t(c_w^t, a_t) - Q(u_t, a_t)) \cdot m_{wt} \tag{25}$$

$$\beta = 0.5 \cdot 0.9^{\frac{t}{1000}} \quad (26)$$

단계 7: 만일 u_{t+1}이 목표상태라면 상태 u_t를 랜덤하게 초기화한다.

그렇지 않다면 u_t를 u_{t+1}로 갱신한다.

단계 8: 종료조건을 만족하지 않으면 단계 3으로 간다.

4. 실험과 분석

본 논문에서 제안한 Fuzzy Q-Map을 평가하기 위하여, 연속적인 상태공간을 갖는 마운틴 카 문제에 적용하였다.

마운틴 카 문제는 그림 3과 같이 자동차가 목표에 도달하기 위하여 적절한 가속을 얻도록 관성(coast), 전진(forward), 후진(backward)의 행동을 학습하는 것이다. 마운틴 카 문제에서 상태공간을 구성하는 요소는 위치(P)와 속도(V)이고, 각각 일정한 범위에 속하는 연속값을 갖는다. 위치(P)는 $-1.2 \le P \le 0.5$ 범위내의 값이고, 속도(V)는 $-0.07 \le V \le 0.07$ 사이의 값을 갖는다. 자동차는 동력이 약한 엔진과 중력의 제약조건을 이기고 목표지점인 0.5에 도달하여야 한다.

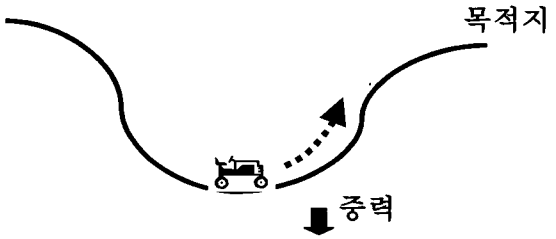

그림 3. 마운틴 카 문제
Fig. 3. Mountain car problem

4.1 Fuzzy Q-Map의 적용

본 실험에서 Fuzzy Q-Map의 노드의 개수는 301개(100개의 클러스터*3개의 이산 행동 +1)이고, fuzziness는 2로 지정하였다. 훈련 집합은 상태공간을 탐험하는 과정에서 만나게 되는 상태들로 이루어진다. 테스트 집합들은 1000개의 상태들로 구성되고, 상태공간에서 랜덤하게 선택된다. 보상함수는 지정한 목표에 도달하면 매우 큰 보상값 99999를 값으로 주고, 나머지 경우는 -1을 주었다. Fuzzy Q-Map의 성능은 중심값의 이동, 위치별 Q값, 이동횟수를 이용하여 평가하였다.

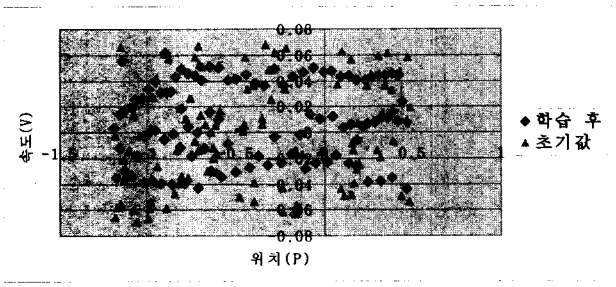

그림 4. 클러스터 중심의 이동
Fig. 4. The movement of the centroid of each cluster in Fuzzy Q-Map

그림 4는 학습에 따른 클러스터 중심의 이동을 보여준다. 학습이 시작되면 중심은 랜덤하게 초기화되고, 목표에 도달한 횟수가 5000번이 될 때까지 약 37만개이상의 상태들을 이용하여 훈련한 결과이다. 학습이 끝난 후에, 적응된 중심들은 초기값을 변경해도 결국 비슷한 좌표로 이동하였다.

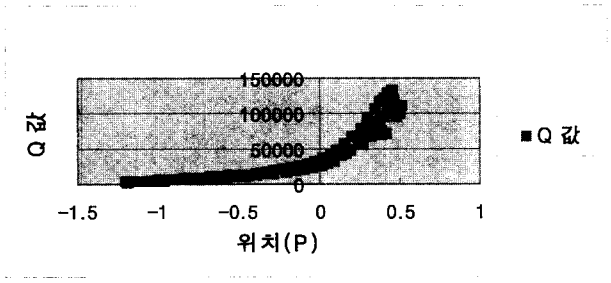

그림 5. Fuzzy Q-Map의 위치별 Q값
Fig. 5. Q values at each position in Fuzzy Q-Map

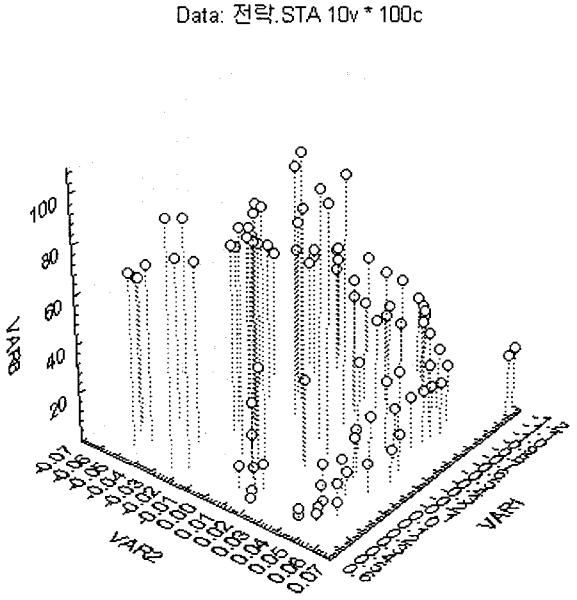

그림 6. Fuzzy Q-Map을 이용한 이동횟수
Fig. 6. The number of movements using Fuzzy Q-Map

그림 5는 학습이 끝난 후, Fuzzy Q-Map에서 각 클러스터 중심의 Q값을 그래프로 그린 것이다. 목표(0.5)에 가까울수록 Q값은 커지므로 목적을 쉽게 이룰 수 있음을 의미한다.

그림 6은 임의의 위치(VAR 1)와 임의의 속도(VAR2)에서 출발하여, 목표에 도달하기 위해 필요한 이동 횟수를 나타낸다. 시작 위치가 목표에 가깝고 속도가 0.01이상이면 적은 수의 이동으로 목표에 도달할 수 있고, 또한 목표에서 멀지만 속도가 양수이면 쉽게 목표에 도달 할 수 있다. 그러나 위치와 상관없이 속도가 0미만이면 목표에 도달하기 위한 가속을 얻어야 하므로 후진이 필요하여 이동수가 많아진다.

4.2 비교

본 실험 에서는 Fuzzy-Q Map과 CMAC, L\..R을 마운틴

카 문제에 적용하여 학습 속도와 예측률을 비교하였다. CMAC와 LWR의 실험 결과는 William D. Smart의 논문을 참조하였다. CMAC는 Fuzzy Q-Map과 같이 최소한의 사전 지식을 가지고 학습한다. LWR은 최적은 아니더라도 훈련 예제 집합을 이용한다. Smart는 상태 500개의 훈련 데이터 집합을 반복해서 경험하고, 상태공간에 골고루 분포된 상태 100개의 테스트 집합을 이용해서 학습 결과를 평가하였다. 각 테스트 데이터는 지금까지 학습한 결과를 바탕으로 목표에 도달할 때까지, 아니면 200번의 시도(time steps)까지 처리된다. 보상함수는 자동차가 목표에 도달하면 보상값 1을, 목표에서 멀어지면 페널티로서 -1을, 그 외의 경우는 0의 값을 준다. Smart는 모든 실험에서 할인율 γ는 0.99로 정하고, 학습율 α는 0.2로 지정하였다.

Fuzzy Q-Map의 실험은 CMAC, LWR과 같은 테스트 집합을 이용하고, 훈련 데이터 집합의 크기는 탐색 시간만큼 점차적으로 증가한다. 보상함수는 4.1의 실험과 같다. 탐험 전략인 $\varepsilon-greedy$에서 ε는 0.1로 지정하고, 할인율 γ는 0.9로, 학습율은 0.5를 초기값으로 하고 시간에 따라서 감소시켰다.

그림 7은 LWR과 Fuzzy Q Map의 학습 속도와 예측률을 보여준다. LWR은 훈련 예제 집합과 같은 일종의 지표를 이용하므로, 25개의 상태만을 경험해도 최고의 예측률까지 근접할 정도로 학습 속도가 빠르다. Fuzzy Q-Map은 상태 150개 이상을 훈련했을 때, 100개 중 80개 이상의 데이터가 목표에 도달함을 보인다. Fuzzy Q-Map의 학습 속도는 LWR보다 느리지만, 그림 8과 같이 CMAC와 비교한다면 빠르다. CMAC는 약 2500개의 상태를 경험했을 때, 80개 이상을 성공하고, 최고 90개까지 목표에 도달함을 보인다. Fuzzy Q Map은 약 200개의 상태를 경험했을 때, 약 86개의 테스트 데이터가 목표에 도달하고 훈련 데이터 집합이 커짐에 따라 다소 감소함을 보인다.

Fuzzy Q-Map 알고리즘의 단점은 훈련 집합에 따라서 학습 결과가 달라진다는 것이다. 훈련 집합의 크기가 125개일 때, 5개의 훈련 집합은 최고 95개부터 최소 30개의 테스트 데이터가 목표에 도달하였다. 크기 150개 이상부터 500개까지는 65개부터 97개까지 성공하였다. Fuzzy Q Map은 훈련 집합의 크기가 커짐에 따라 안정적이지만, 훈련 데이터 모두에 적응해야 하므로 간섭 현상이 발생해서 오히려 예측률은 감소한다.

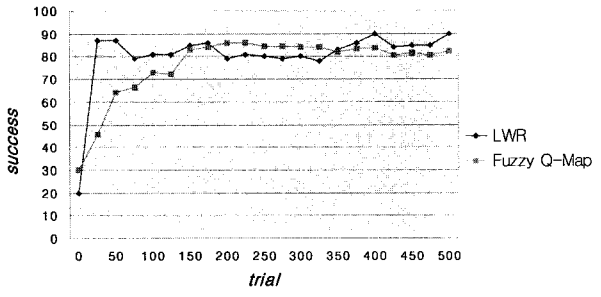

그림 7. LWR과 Fuzzy Q Map의 비교
Fig. 7. The comparision between LWR and Fuzzy Q Map

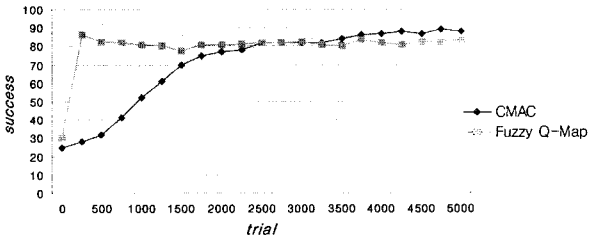

그림 8. CMAC와 Fuzzy Q-Map의 비교
Fig. 8. The comparision between CMAC and Fuzzy Q Map

5. 결론 및 향후 연구 과제

강화학습은 에이전트와 환경이 서로 상호작용을 하면서 얻은 경험으로부터 제어 규칙을 학습하는 방법이다. 강화학습은 환경의 모델이 알려지지 않은 실세계의 제어 문제 학습에 적합하다. 그러나 실세계의 많은 문제들은 연속적인 상태와 행동을 가지므로 이산 데이터를 다루는 Q-Learning 알고리즘을 그대로 이용할 수 없다. 또한 Q learning은 사전지식이 없는 상태에서 학습을 시작하기 때문에 학습 초기에는 랜덤하게 행동을 선택해야 한다. 그러므로 최적에 가까운 학습 결과에 수렴하는 속도가 느리다. 본 논문에서는 사전지식이 없는 강화학습의 학습 속도의 개선과 기억 장소의 문제를 해결하기 위한 함수 근사 방법으로 퍼지 클러스터링이 적합하다고 보았고, 퍼지 클러스터링 알고리즘인 FCM을 기초로 한 Fuzzy Q Map을 제안했다. 퍼지 클러스터링은 비교사 학습 방법(unsupervised learning)이며, 강화학습과 같이 사전 정보가 없는 상태에서 학습한다. Fuzzy Q-Map은 소속도를 이용하여 각 퍼지 클러스터가 제안하는 행동을 조합하여 선택한다. 각 클러스터의 중심과 Q값의 갱신에 TD 에러와 소속도를 이용한다. 제안한 알고리즘을 마운틴 카문제에 적용한 결과, 4.2절에서 제시한 바와 같이 훈련 예제를 이용하는 LWR과 비슷한 학습 속도를 보였다.

향후 연구 과제는 다음과 같다. 첫째, Fuzzy Q-Map 알고리즘은 훈련 집합의 크기와 구성에 따라서 학습 결과가 달라진다. 훈련 집합을 구성하는 방법과 간섭현상을 줄이는 연구가 필요하다. 둘째, 유사한 전략의 클러스터들이 Fuzzy Q Map존재하였다. 중복된 클러스터들을 제거하는 연구가 필요하다.

References

[1] Stephan ten Hagen and Ben Krose, "Q learning for System with continuous state and action spaces", BENELEARN 2000, 10th Belgian-Dutch conference on Machine Learning.

[2] Chris Gaskett, David Wettergreen, and Alexander Zelinsky, "Q learning in continuous state and action spaces", Australian Joint Conference on Artificial Intelligence 1999.

[3] 전효병, 이동욱, 김대준, 심귀보, "퍼지추론에 의한 리커런트 뉴럴 네트워크 강화학습", 한국퍼지 및 지

능 시스템 학회 '97년도 춘계학술대회 논문집.
[4] Richard S. Sutton, Andrew G. Barto "Reinforcement Learning: An Introduction". The MIT Press, Cambridge, MA., 1998.
[5] Juan Carlos Santamaria, Richard S. Sutton, Ashwin Ram, "Experiments with Reinforcement Learning in Problems with Continuous State and Action Spaces", COINS Technical Report 96-88, December 1996.
[6] William Donald Smart, "Making Reinforcement Learning Work on Real Robots", Ph D Thesis, Department of Computer Science, Brown University, 2002.
[7] Jan Jantzen, "Neurofuzzy Modelling", Technical Report, Technical University of Denmark 1998.
[8] 정석일, 이연정, "분포 기여도를 이용한 퍼지 Q-learning", 퍼지 및 지능시스템학회 논문지 2001, Vol. 11, No.5 pp.388-394.
[9] Pierre Yves Glorennec, Lionel Jouffe, "Fuzzy Q-learning", Proceedings of Fuzz-Ieee'97, Sixth International Conference on Fuzzy Systems, P719-724, Barcelona, july,1997.
[10] Lionel Jouffe, "Fuzzy Inference System Learning by Reinforcement Methods", Ieee Transactions on System, Man and Cybernetics, vol.98, no 3, August,1998.
[11] Andrea Bonarini, "Delayed Reinforcement, Fuzzy Q-learning and Fuzzy Logic Controllers", In Herrera, F., Verdegay, J. L. (Eds.) Genetic Algorithms and Soft Computing, (Studies in Fuzziness, 8), Physica-Verlag, Berlin, D, 447-466.
[12] William D. Smart, Leslie Pack Kaelbling, "Practical Reinforcement Learning in Continuous Spaces", Proceedings of the sixteenth International Conference on Machine Learning, 2000.
[13] William D. Smart, Leslie Pack Kaelbling, "Reinforcement Learning for Robot Control", In "Mobile Robots XVI", 2001.
[14] Artistidis Likas, "A Reinforcement Learning Approach to On-Line Clustering", Neural Computation, vol. 11, no. 8, pp. 1915-1932, 1999.

저 자 소 개

이영아

1992년 : 동덕여자대학교 전자계산학과 (이학사)
1994년 : 동덕여자대학교 대학원 전자계산학과 (공학석사)
1999년~현재 : 경희대학교 컴퓨터공학과 박사과정

관심분야 : 강화학습, 에이전트, 데이터마이닝, 로보틱스

e-mail : leeyaa@iislab.kyunghee.ac.kr

정태충

1980년 : 서울대학교 전자공학과(학사)
1982년 : 한국과학기술원 전자공학전공 (공학석사)
1987년 : 한국과학기술원 전자공학전공 (공학박사)
1987년~1988년 : KIST 시스템 공학센터 선임연구원
1988년~현재 : 경희대학교 컴퓨터공학과 교수

관심분야 : 기계학습, 보안, 최적화, 에이전트

e-mail : tcchung@khu.ac.kr