자율 이동 로봇의 주행을 위한 영역 기반 Q-learning

Region-based Q-learning For Autonomous Mobile Robot Navigation

。차 종 환, 공 성 학, 서 일 홍

한양대학교 전자공학과 (Tel : +82-31-408-5802; Fax : +82-31-408-5803; E-mail: ihsuh@email.hanyang.ac.kr)

Abstract : Q-learning, based on discrete state and action space, is a most widely used reinforcement learning. However, this requires a lot of memory and much time for learning all actions of each state when it is applied to a real mobile robot navigation using continuous state and action space. Region-based Q-learning is a reinforcement learning method that estimates action values of real state by using triangular-type action distribution model and relationship with its neighboring state which was defined and learned before. This paper proposes a new Region-based Q-learning which uses a reward assigned only when the agent reached the target, and get out of the local optimal path with adjustment of random action rate. If this is applied to mobile robot navigation, less memory can be used and robot can move smoothly, and optimal solution can be learned fast. To show the validity of our method, computer simulations are illustrated.

Keywords : RQ-learning, neighboring state, action distribution model, mobile robot, navigation

1. 서론

자율 이동 로봇을 제어하는 방법으로 강화 학습이 많이 사용되는데 이것은 알려지지 환경에서 행동과 보답을 주고 받으며 임의의 상태에서 가장 적합한 행위를 학습하는 방법으로 trial-and-error 책략에만 의존한다.[1]

이러한 방법들 중 Q-learning은 가장 널리 사용되는 방법들 중 하나로 이 학습법은 현재 상태에서의 행위를 미래 행위들로부터 얻게 되는 총 보답을 예측하는 행위값에 대응시키도록 하는 행위 함수를 학습하는 방법이다[3]. 그런데, Q-learning은 오차를 줄이기 위해서 많은 기억 공간과 매우 긴 학습 시간이 필요하다는 점 때문에 실제 환경에 적용하기는 쉽지 않다. 이러한 단점을 보완하기 위해 여러 가지 연구가 진행되어 왔다. 예를 들면, Q-table의 크기를 가변하는 방법과 보답을 실수로 얻는 방법, 그리고 행위값의 초기화나 좋은 탐색 책략을 미리 알려주는 방법이다[4,5,6,7].

하지만, 강화 학습의 대부분의 연구들은 불연속 상태공간과 불연속 행위공간을 기반으로 이루어졌기 때문에 실제 자율 이동 로봇의 주행에 적용시키려면 연속 상태 공간의 연속 행위에 대해 학습하여야 하므로 많은 기억공간과 긴 학습 시간이 필요하며 또한 행위 가 불연속이라는 제한을 갖게 되고, 정의된 상태들이 정확한 현재 상태를 나타낼 수 없기 때문에 오차가 존재하게 된다[8].

이러한 제약은 RQ-learning을 사용함으로써 줄일 수 있게 되는데 기존의 RQ-learning은 보답을 얻기 위해서는 모든 상태에서 목표 지점까지와의 거리를 안다는 가정이 필요하였고, 또, 지역적 최적해를 벗어나지 못 할 수도 있다는 단점이 있다[2]. 본 논문에서는 이러한 점을 보완하여 자율 이동 로봇에 적용하여 보고, 컴퓨터 모의 실험을 통하여 Q-learning을 적용하였을 때의 결과와 비교함으로써 제안한 RQ-learning의 효과적인 점을 검증하여 본다.

2. Q-learning과 RQ-learning

2.1 Q-learning

Q-learning은 대표적인 off-policy 강화학습으로 주어진 환경과 상호작용을 하며 최적의 행위 함수를 학습하는 방법이다. 임의의 상태에서 최적의 행위책략(policy) $p^*(s_t)$ 는 행위 함수로부터 결정된다.

$$p^*(s_t) = \arg\max_a Q(s_t, a) \qquad (1)$$

그리고, 행위 함수는 다음과 같이 갱신된다.

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + a\left(r_{t+1} + g \max_a \{Q(s_{t+1}, a)\} - Q(s_t, a_t)\right) \qquad (2)$$

Q-learning의 알고리즘을 간략히 소개하면 다음과 같다.

[Q-learning 알고리즘]

1. Q-테이블 및 각 파라미터 (a, g) 들을 알맞게 초기화한다.
2. 다음 내용을 반복한다.
   1) Q-테이블로부터 행위를 결정한다. 이때 임의의 비율로 랜덤 행위를 만들어 주어야 한다.
   2) 행위에 대한 다음 상태와 보답을 얻는다.
   3) 현재 상태의 행위값을 갱신한다.
   4) 행위책략을 갱신한다.

2.2 RQ-learning

기존의 RQ-learning은 모든 상태에서 목표점까지의 거리를 안다는 가정이 필요하고, 지역적 최적해를 벗어나지 못 할 수도 있기 때문에 이를 보완한 RQ-learning을 제안한다.

3.향상된 Region-based Q-learning(RQ-learning)

Q-learning은 실제 환경에 적용하기 위해서는 많은 기억 공간과 긴 학습 시간이 필요하게 되고, 불연속 상태 및 행위 공간에서 학습하기 때문에 출력되는 행위가 부드럽지 못하다. RQ-learning은 이러한 점을 보완하여 연속 상태 및 행위 공간에서 학습하는 방법으로 공간 내의 모든 상태에 대해 학습하지 않고 미리 정해놓은 특정 상태(이웃 상태)에 대해서만 학습하여 이로부터 모든 상태에서의 최적 행위 및 행위값을 얻는다.

3.1 이웃 상태의 정의

이웃 상태는 연속상태 공간내에서 미리 정의된 특정 상태로서 불연속 Q-learning에서 사용되는 상태와 비슷하다고 볼 수 있고, 이웃 상태에서의 모든 행위에 대한 행위값은 행위 분배 모델(action distribution model)에 의해 최대 행위값($Q_{\max}$)과 그 때의 행위(a^p)만으로 유추 가능하다. 또한 실제 상태에서의 행위값은 이웃 상태의 행위값들의 선형 조합으로 표현이 가능하다. 그리고, 실제 상태에서 행위에 대한 보답은 이웃 상태의 상관도(relationship)에 비례하여 할당된다.

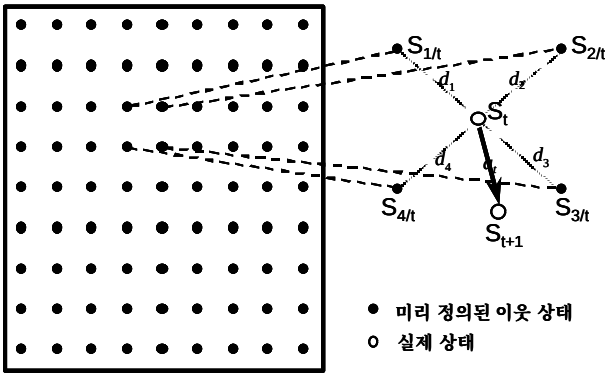

그림 1. 공간 상태 내의 이웃상태와 실제 상태

상관도는 실제 상태가 이웃 상태에 얼만큼의 영향을 주고 받는가에 관계된 값으로, 임의의 실제 상태에서의 상관도들의 합은 항상 1이 되고, 각 상관도는 다음과 같이 구할 수 있다.

$$m_i = \frac{1-d_i}{\sum_j (1-d_j)} \quad , \quad if\ (d_i > 1) \rightarrow d_i = 1 \qquad (3)$$

3.2 행위 분배 모델(action distribution model)

Q-테이블은 임의의 상태에서 모든 행위에 대한 행위값을 저장하는 곳으로 가장 큰 행위값과 그 때의 행위를 얻기 위해 쓰여지는데 연속 행위공간에서는 모든 행위에 대하여 행위값을 가질 수 없게 된다. 그래서, 특정 행위에서 가장 큰 행위값을 갖고 특정행위와 관계가 적을수록 행위값이 작아지는 형태의 Q-테이블을 모델링하여 특정 행위와 그 행위값만으로 모든 행위값을 유추할 수 있도록 하는 것이 행위 분배 모델(action distribution model)이다. 이 모델링은 목표점까지의 거리를 행위값으로 가정하는 것처럼 임의의 상태에서 행위 축에 대한 행위값들이 서로 관계가 있어야 한다.

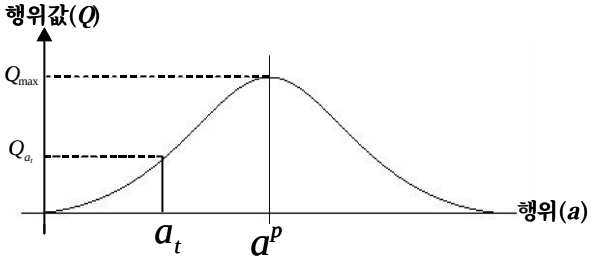

그림 2. 행위에 대한 행위값의 모델링

임의의 행위 a_t 에 대한 행위값은 다음과 같이 구할 수 있다.

$$Q_i(s_{i/t}, a_t) = f(Q_{i,\max}, a^p, a_t) = \frac{Q_{i,\max}}{D^2 + 1 - 2 \times \vec{a^p} \bullet \vec{a_t}}$$
$$a^p = \arg(Q_{i,\max}),\ D = 2 \qquad (4)$$

3.3 RQ-learning 알고리즘

[RQ-learning 알고리즘]

1. 특정 상태들의 행위값과 여러 파라미터를 초기화한다.
2. 다음 내용을 반복한다.
   1) 현재 상태로부터 이웃 상태를 결정하고 상관도를 구한다.
   2) 이웃 상태들로부터 행위를 구한다.
   3) 랜덤 비율에 따라 행위를 결정한다.
   4) 결정된 행위를 수행하고, 그에 대한 보답을 얻는다.
   5) 다음 상태에 대한 최대 행위값을 구한다.
   6) 현재 상태에 대한 이웃 상태의 행위값들을 갱신한다.

Q-learning에서는 Q-테이블로부터 행위를 결정하지만 RQ-learning에서는 실제 상태의 이웃 상태들로부터 행위를 구한다.

$$a_t = \sum_{i=1}^{2^N} m_i a_i \ , \quad \text{N:dimension} \qquad (5)$$

행위를 결정할 때는 일정한 비율의 랜덤 행위가 필요한데 학습 초기에는 Q-테이블의 최적 행위가 올바르지 않을 확률이 높고 학습할수록 그 확률은 높아지기 때문에 랜덤 비율을 학습 초기에는 높고 학습할수록 낮아지는 것이 바람직하다. 그리고, 지역적 최적해를 피하려면 일정 주기마다 한번씩 랜덤 비율을 크게 해줄 필요가 있다. 랜덤 비율을 그림으로 보면,

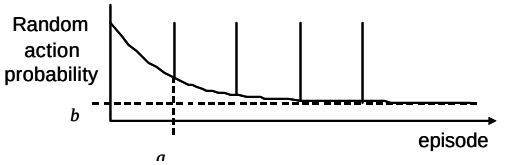

그림 3. episode 진행에 따른 랜덤 행위 비율

행위의 결과가 목표에 도달한다면 그에 대한 보답으로 1이 얻어지고, 그 외의 경우에는 0이 된다.

$$r = \begin{cases} 1 : \text{when it achieve the goal} \\ 0 : \text{otherwise} \end{cases} \qquad (6)$$

행위의 결과 현재 상태에서 다음 상태로 이동하게 되고 현재 상태의 행위값들을 갱신하기 위해 필요한 다음 상태에서의 최대 행위값은 행위 분배 모델을 이용하여 다음과 같이 구한다.(식(4) 이용)

$$\max_a \{Q(s_{t+1}, a)\} = \max_a \left\{ \sum_{i=0}^{2^N} m_i Q_i (s_{i/t+1}, a) \right\} \qquad (7)$$

이웃 상태의 행위값들을 갱신하는 식은 Watkin's action-value update equation에 따라 다음과 같이 구할 수 있다.

$$Q_i(s_{i/t}, a_t) \leftarrow Q_i(s_{i/t}, a_t) + a\left(r_{i/t+1} + g \max_a \{Q(s', a)\} - Q(s_{i/t}, a_t)\right)$$

$$r_{i/t+1} = m_i r_{t+1}, \quad a_t : s_t \rightarrow s_{t+1}, \quad s_{i/t} \rightarrow s' \qquad (8)$$

만일 임의의 행위에 대한 다음 상태로의 이동이 유일하다면 $a = 1$이 되기 갱신식이 때문에 다음과 같이 바뀐다[1].

$$Q_i(s_{i/t}, a_t) \leftarrow r_{i/t+1} + g \max_a \{Q(s', a)\} \qquad (9)$$

이렇게 얻은 이웃 상태의 행위값이 최적 행위값보다 클 때는 새로 얻은 행위값을 기준으로 재모델링한다.

위의 과정들을 반복하여 학습함으로써 미리 정의된 이웃 상태들은 최적의 행위들로 학습되어진다.

4. 모의 실험 결과

모의 실험은 장애물이 존재하는 연속 공간을 25*25로 나누어 Q-learning과 RQ-learning의 방법을 사용하여 학습시킨 후 각 상태의 학습 효과 및 학습 시간을 비교한다. 그림 4와 그림 5는 Q-learning 방법을 사용한 결과로서 상태 공간 내의 모든 상태가 학습되어지는 과정을 나타낸다. 그림 6과 그림 7은 RQ-learning 방법을 사용한 결과로서 연속 상태공간 내의 미리 정의된 이웃 상태들이 최적행위로 학습되어지는 과정을 나타낸다. 두 학습 방법을 비교하여 보면 RQ-learning이 Q-learning보다 학습시간이 적게 소요되며 부드러운 행위로 학습되어진다.

그림 4. Q-learning , 89 episode 로 학습된 최적 행위 결과

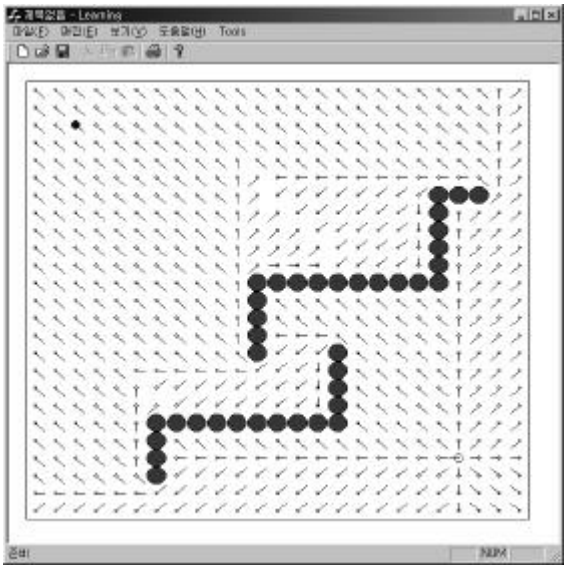
그림 5. Q-learning , 124 episode 로 학습된 최적 행위 결과

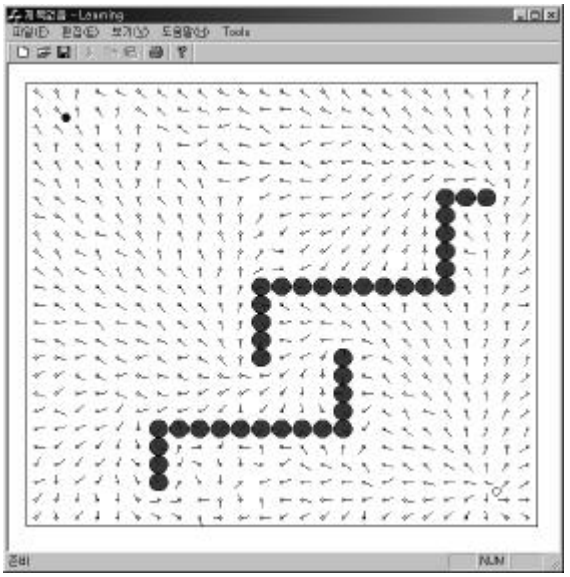
그림 6. RQ-learning , 54 episode 로 학습된 최적 행위 결과

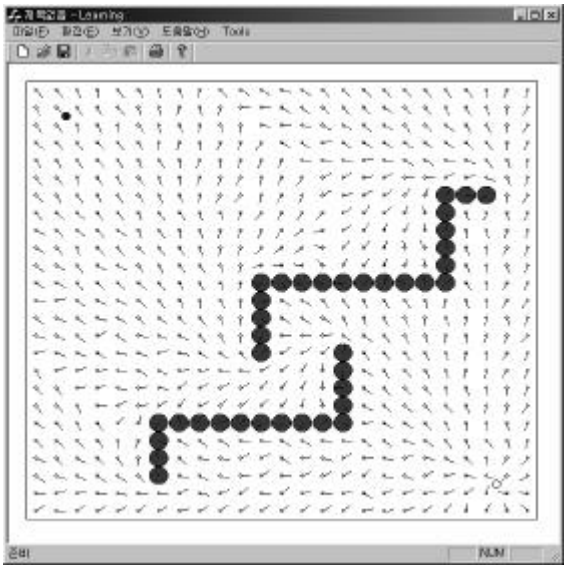
그림 7. RQ-learning , 62 episode 로 학습된 최적 행위 결과

모의 실험에서는 다음과 같은 몇 가지 가정을 하였다.

1. 로봇은 상태공간 내의 절대 위치를 항상 알 수 있다.
2. 행위를 구하기 전에 먼저 장애물의 위치를 파악하여 그 상태에서 가능한 행위 집합을 구한 후 그 집합 내에서 행위를 구한다.
3. 로봇이 목표지점에 도달하는 지를 알 수 있다.

이러한 가정하에 모의실험을 여러 번의 반복하여 학습 시간의 평균을 측정해 보면 다음과 같은 결과를 얻을 수 있다.

표 1. 학습 수행 시간.

	Q-learning		RQ-learning	
	episode	총 step수	episode	총 step수
목표물까지 최적행위	89	57245	54	36214
모든 상태 최적행위	124	81969	62	48542

5. 결론 및 추후과제

본 논문에서는 연속 상태 및 행위 공간에서 Q-learning보다 우수한 특성을 나타내는 학습 방법을 제안하였다. 이를 위해, 기존의 RQ-learning의 보답 알고리즘을 분석하고 효율적인 보답 방법을 제안하였으며, 지역적 최적해를 보완하기 위한 방법을 제안하였다. 제안된 RQ-learning의 우수성을 보이기 위해, 이동 로봇의 자율 주행에 적용하는 모의 실험에서 Q-learning의 방법과 RQ- learning의 방법을 사용하였고, 각각의 학습 결과를 비교하였다. 모의 실험 결과 자율 이동 로봇의 주행에는 학습 시간 측면과 최적 행위의 학습결과 측면에서 RQ-learning이 효과적으로 적용될 수 있음을 보였다.

추후 과제로 제안된 알고리즘을 실제 이동 로봇에 적용하는 연구가 진행 중에 있다.

참고문헌

[1] R. S. Sutton and A. G. Barto, Reinforcement Learning. Cambridge, MA : MIT Press, 1998.

[2] 김재현, 서일홍, "지능형 로봇 시스템을 위한 영역기반 Q-learning," 제어계측자동화로보틱스연구회 합동학술발표회, pp. 271-276, 1997.

[3] C. Watkins, P. Dayan, "Q-learning, technical note," Machine Learning, Vol. 8, pp. 279-292, 1992.

[4] Yoichi Hirashima, "Q-learning Algorithm Using an Adaptive-Sized Q-table," Proceedings of the 38th Conference on Decision & Control, Phoenix, pp. 1599-1604, December, 1999.

[5] T. D'Orazio, G. Cicirelli, "Continuous Reward versus Discrete Reward in a Qlearning Agent," The Fifth ICARCV'98, Singapore, pp. 584-588, December, 1998.

[6] J. del, R. Millan, "Rapid, safe, and incremental learning of navigation strategies," Sys. Man and Cybernetics, 26(3), 1996.

[7] T. D'Orazio, G. Cicirelli, C. Distante and G. Attolico, "Learning actions from vision-based positioning in goal-directed navigation," IROS'98 International Conference on Intelligent Robots and Systems, 1998.

[8] Yasutake Takahashi, Masanori Takeda, and Minoru Asada, "Continuous Valued Q-learning for Vision-Guided Behavior Acqusition," Proceeding of the 1999 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, Taipei, Taiwan, R.O.C, pp. 2555-260, August, 1999.