지능형 시스템을 위한 수정된 Q-Learning

정회원 김 영 준*

Modified Q-Learning for Intelligent System

Young Jun Kim* ***Regular Member***

요 약

본 논문에서는 지능형 시스템을 위한 수정된 Q-Learning을 제안한다. 미리 계획되지 않은 상황에 유연하게 적응하기 위해서는 시스템 스스로가 학습하는 능력이 요구된다. 이를 위하여 Q-Learning은 불연속 상태 공간과 행위 공간을 정의하고 현재 상태로부터 목표 상태에 도달하기 위한 최적의 행위 집합을 구하기 위한 통계적인 해결책으로 제안되었다. 그러나, 연속적인 상태공간과 행위공간을 내포하는 실질적인 환경에 Q-Learning을 적용하기 위해서는 너무 많은 양의 기억 공간과 학습시간이 필요하게 된다. 따라서 본 논문에서는 이러한 연속 상태 공간 및 연속 행위 공간에서의 학습을 위해, 기존의 Q-Learning을 영역기반의 reward(보답) 할당과 삼각형태의 Q-value, 모델을 사용함으로써 실질적인 응용에 적용할 수 있는 일반화된 Q-Learning 알고리즘을 개발하였다.

Key Words : Q-Learning, Q-Value, Intelligent System

ABSTRACT

In this paper, for a continuous state space applications, a novel method of Q-learning is proposed, where the method incorporates a region-based reward assignment being used to solve structural credit assignment problem and a convex clustering approach to find a region with the same reward attribution property. Our learning method can estimate a current Q-value of an arbitrarily given state by using effect functions, and has the ability to learn its action ssimilar to that of Q-learning. Thus, our method enables a system to adapt smoothly to a real environment. To show the validity of our method, the proposed Q-learning method is compared with conventional Q-learning method through a simple two dimensional free space navigation problem.

Ⅰ. 서 론

최근 강화 학습에 대한 대부분의 연구들은 불연속 상태 공간과 불연속 행위 공간을 기반으로 하여 이루어졌다. 따라서 불연속 상태 공간으로 모델링 되는 미지의 환경과 상호작용을 통하여 불연속적인 행위들 중 최적의 행위를 학습 할 수 있었다. 이러한 방법들중 Q-learning은 가장 널리 사용되는 방법들 중 하나이다. Q-learning은 현재 행위에 대한 평가를 위해 현재의 행위가 최적의 행위궤적을 따른다고 가정할 때 현재의 행위로부터 미래 행위들에 대한 보답을 감쇠상수를 고려하여 합한 행위값(Q-value)을 정의하여 사용한다. 이러한 행위값들은 각 상태에서 최적의 행위를 수행 할 수 있는 근거자료 역할을 수행하게 된다. 이러한 Q-learning을 실질적인 작업에 적용하기 위해서 많은 노력이 진행되어 왔다.[1][2][4][5] 예로써, Berenji 등에 의해 개발된 Fuzzy Q-leanring 방법은 Q-value 갱신식에 상황에 대한 제약을 첨가하여 현재의 상황을 반영하고자 하였다.[3] 그러나 이러한 Q-learning을 비롯

* 인하공업전문대학 정보통신과 (yjkim@inhatc.ac.kr)
논문번호 : 08006-0125, 접수일자 : 2008년 1월 25일

한 대부분의 강화학습 방법들은 불연속 상태 공간과 행위공간을 학습 환경의 모델로 사용하기 때문에 실질적인 로봇 응용분야에서 이러한 알고리즘을 적용하기에는 다음과 같은 여러 문제점들을 극복하여야만 한다.

(1) 너무 많은 기억 공간을 필요로 한다.
(2) 모든 상태에 대해 학습을 수행하여야 하므로 학습기간이 길다
(3) 출력이 불연속이라는 제한을 가진다.

위에서 언급한 이러한 문제점들을 해결하기 위하여 퍼지의 Q-table을 새로운 퍼지 Rule들로 대체하여 각 Q값들을 퍼지 추론에 의하여 생성하고, 각 Rule의 조건부 파라메터들을 Steeppest deccent 방법으로 조정하고자 하는 새로운 시도가 이루어졌다. 그러나 이 경우 초기 Rule 들의 생성이 어렵고, Steepest descent 방법을 사용함으로써 국부, 극소점에 빠지는 경우 원하는 목표 상태로의 수렴을 보장할 수 없다는 문제점이 있다.

여러 가지의 강화학습들을 특성화하는 기본적인 논제들 중 credit assignment problem은 " 일련의 sensor-action-feedback 으로부터 어떻게 최적의 행위를 배울 것인가" 로 정의되며, 각 강화학습에서 풀어야 할 기본적인 문제이다. 이러한 credit assignment 문제중 structural credit assignment 문제는 "현재 받은 reward 가 상태 공간내의 각 상태들에게 어떻게 영향을 끼칠 것인가"로 정의 된다. 이러한 관점에서, 기존의 Q-learning은 point-based credit assignment 방법이라고 정의 내릴 수 있다. 따라서 이러한 상태점에 기반으로 보답을 할당하는 Q-learning 을 일반화하기 위해 특정 상태 영역에 보답을 할당하는 Modified Q-learning을 제안하고자 한다.

Ⅱ. Q-learning in Discrete state space

강화 학습에서는 기본적으로 시스템과 환경과의 상호작용을 이산 반복 공정(Discrete Time Cyclic Processes)에서 동작하는 유한 상태를 갖는 두개의 대행자들(환경과 Agent)로 모델링 한다. 이러한 상호작용은 다음과 같다. 먼저 로보트는 환경에 대한 현재상태를 감지하고 적절한 행위를 선택하여 이를 수행한다. 다음으로, 환경은 현재 상태와 수행된 행위에 근거하여 새로운 상태로 전이되고 수행된 행위에 대한 보답(Reward)을 발생시키며, 이를 시스템에게 되돌려 준다. 이러한 상호작용을 통해 시스템은 각 상태에 대한 적절한 행위를 배우게 된다. 이러한 관계를 정리하여 이론화한 Q-learning 알고리즘은 다음과 같다.

Q-learning 알고리즘

[초기화]

1. 초기화 :

(1) 난수 혹은 사전 정보를 이용한 Q-table $Q(s,a)$ 초기화

(2) 초기화된 Q-table를 근거로 Policy f_i 초기화

$$f_i \leftarrow a \quad \text{such that ,}$$

$$Q_i^a(t+1) = \max_{b \in A}\{Q_i^b(t)\} \tag{1}$$

여기서 t는 tth iteration을, i는 *현재상태를*, A는 현재 정의된 행위 집합을 각각 나타내며, 따라서 f_i는 현재 상태에서 최적의 행위계획(policy)을 나타내고, $Q_i^a(t+1)$ 는 다음 iteration의 현재 상태 i에서 수행할 행위 a에 대한 행위값을 나타낸다.

(3) 여러 파라메타(γ , α , ρ)의 초기화

[반복]

2. 현재 상태을 받아들임 (s 현재상태)
3. Policy Table로부터 현재상태에 해당하는 행위 a를 수행 하거나 ρ 만큼의 비율로 임의의 행위를 수행. 여기서 랜덤 행위를 수행하는 것은 최적의 policy를 구하기 위한 필요조건이 된다.
4. 환경으로부터 수행된 행위에 대한 보답(Reward)r를 받음.
5. 다음 식 (2)를 이용하여 현재 상태에서 수행한 행위값(Q-value) $Q(s,a)$를 갱신.

$$Q_i^a(t+1) = \alpha Q_i^a(t) + (1-\alpha)(r_i^a + \gamma \max_{b \in A}\{Q_{i+1}^b(t)\}) \tag{2}$$

여기서 α (0< α <1)는 학습 속도를 나타내며, 는 미래 행위에 대한 보답에 대한 감쇠 상수이다.

6. 식 (1)를 이용하여 Policy f_i 갱신.

Ⅲ. Modified Q-Learning(MQ-Learning)

2장에서 기술하였듯이, 기존의 Q-learning을 실제 환경에 적용하기 위해서는 너무 많은 기억 공간과

학습 시간이 필요하게 된다. 또한, 기존의 Q-learning은 불연속 상태 및 행위 공간에서 사용되기 때문에 출력되는 행위가 부드럽지 못하다. 이러한 제한을 극복하기 위해, 본 논문에서는 먼저 기존의 Q-learning을 영역 기반으로 보답(reward)을 할당하는 영역 기반 Q-learning(Region-based Q-learning)을 개발하였다. 이러한 영역 기반 Q-learning방법은 기존의 현재 상태에만 보답을 할당하는 방법(point-wise Q-learning)을 포함하는 일반화된 방법이라고 할 수 있다. MQ-learning에서는 상태 공간 내의 모든 상태에 대해 학습할 필요가 없다. 즉, 단지 미리 설정한 특정한 상태들(주변 상태)에 대해서만 학습을 수행하며 최적의 행위도 이러한 주변 상태로부터 생성하게 된다.

3.1 주변 상태(neighboring states)의 행위값(Q-value) 결정.

N-차 상태 공간을 이루는 각 상태 축들이 l 개의 분해기능을 갖는다고 가정하자. 이러한 상태 공간 구조에서 주변 상태(*neighboring state*)란 현재상태가 포함되어있는 hyperbox의 각 꼭지점에 위치한 상태로 정의 하고자 한다. 이때, 임의의 hyperbox내의 임의의 위치에 있을 수 있는 현재 상태 s_i에서 얻은 보답(reward)을 r_i 라고 정의하고, 현재 상태의 j번째 주변 상태 $s_{i,j}$ 의 보답을 r_j 라 정의한다. 현재 상태의 보답과 주변 상태로 전파되는 보답과의 관계를 *effect function* $_{i,j}(s_i,s_{i,j})$ 로 정의한다면 현재 상태의 보답으로부터 주변 상태로 전파되는 보답은 식 (3)와 같이 정의 될 수 있다.

$$r_j = \mu_{i,j} r_i \tag{3}$$

그림 1에서 볼 수 있는 것과 같이, 특정 주변 상태의 보답은 $_{i,j}(s_i,s_j)$와 r_i을 곱함으로써 얻을 수 있다. 따라서 최적의 policy를 따를 때 정의되는 s_{ij}에 전달되는 보답의 감쇠 합인 Q-value는 다음 식 (4)과 같이 사용할 수 있다.

$$Q_j^{a_i} = \sum_{n=0}^{\infty} \gamma^n \mu_{i+n,j} r_{i+n} \tag{4}$$

이러한 새롭게 정의된 Q-value에 대해 다음의 *Theorem 1*이 성립한다.

Theorem 1

현재 상태에서, 식(3)에서 같이 $r_j = \mu_{i,j} r_i$ 에 의해 정의된 보답을 사용하는 식 (4)의 을 기존의 Q-value 갱신식에 의해 갱신 시키면, iteration이 증가함에 따라 최대의 최적 행위로 수렴하게 된다.

만일 각 hyperbox의 모든 꼭지점들이 hyperbox의 중앙으로 집중된다면, l 만큼 등 간격으로 떨어진 점인 hyperbox을 얻을 수 있다. 이러한 경우, 임의의 hyperbox에 대한 s_{ij} and $s_{i,j+1}$사이의 유클리디안 거리를 나타내는 $d(s_{i,j}, s_{i,j+1})$ 은 hyperbox의 용적이 0이기 때문에 0이 되어야 함을 알 수 있다. 따라서, 마찬가지의 경우에 현재 상태에 기인하는 보답은 모든 주변 상태에서도 역시 동일한 양만큼의 영향을 주게 된다. 결국, 모든 존재할 수 있는 상태들은 이산 상태 공간으로 정의되어 지고, 이러한 상태들에서의 최적 행위 생성을 위한 Q-value들은 기존의 Q-learning의 Q-value 갱신식 (2)에 의해 구할 수 있게 된다. 위의 고찰을 통해, 식 (4)에서 정의된 $Q_j^{a_i}$ 에 대한 정의가 기존의 Q-learning에 의해 얻은 $Q_j^{a_i}$ 값 을 포함하는 일반적인 형태가 되기 위해서는 함수에 대해 다음과 같은 특성이 성립해야 한다.

$$\lim_{d(s_i,s_j)\to 0} \mu_{i,j} \to 1 \tag{5}$$

여기서, 각 hyperbox에 대해 독립적인 *effect function*이 존재할 수 있으므로, *effect functions*의 개수는 최대 주변 상태 수 $(N^l\text{-}1)^n$ 만큼 존재하게 된다. 위의 특성들을 만족하는 모든 *effect functions*들을 구한다는 것은 매우 힘든 일이므로, 우리는 모든 hyperbox에 대해 다음 식(6)과 같은 현 상태 로부터 멀어짐에 따라 보답의 영향이 단순 감소 하는 종형의 *effect functions*를 사용하고자 한다.

$$\mu_{i,j}(s_i, s_{i,j}) = exp(-\lambda \cdot d^2(s_i, s_{i,j})) \tag{6}$$

여기서, $d^2(s_i,s_{i,j})$ 는 상태i와 상태j간의 유클리디안 거리를 나타내며, 함수의 형태를 결정한다. 간단한 계산으로 식 (6)이 식(5)를 만족함을 알 수 있을 것이다. 위와 같이, 현재상태의 보답과 주변 상태의 보답간의 관계를 사용하여 갱신된 주변 상태의 현재 행위에 대한 Q-value는 주변 상태에 존재하는 삼각형의 Q-value 모델을 갱신하도록 영향을 준다.

3.1 삼각 형태의 Q-value 모델.

Q-table의 설립 목적은 모든 행위 중 가장 큰 Q 값을 갖는 행위를 찾아 이를 새로운 Policy의 요소로 등록하기 위해 사용되므로, 특정 행위에서 최고치를 갖고 특정 행위와 관계가 멀수록 일정하게 Q-값이 작아지는 형태의 Q-table를 고려해 볼 수 있다. 행위간의 관계를 단지 행위 벡터 공간 내의 유클리디안 거리로 정의하고, 현재상태의 Q-table내의 Q-value들을 특정 행위에서 최고의 Q값을 갖고 거리에 비례적으로 단순 감소하는 특성이 있는 함수로 모델링하면, 특정 상태에서 특정 행위축에 대한 모든 행위들의 Q-value들을 그림 1와 같이 Cone-shaped function으로 나타낼 수 있으며, 이를 특정 상태에서 Q-value model이라고 정의한다.

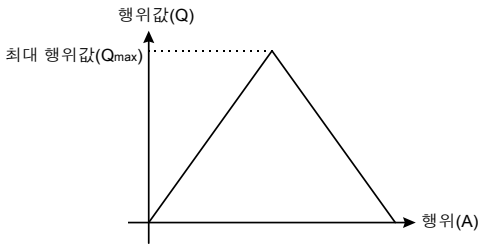


그림 1. 행위값의 삼각 모델링

3.2. Q-value model로부터 최대 Q값 및 최적 행위 생성

삼각형 Q-value 모델을 사용하여 식 (13)에서와 같이 현재 상태에 대한 최적 행위는 현재 상태에서 가능한 모든 행위들의 Q-value들 중 최대치로 결정된다.

$$a_i = \arg(\max_{\forall a}\{\sum_{j=1}^{N}\mu_{i,j}Q_{i,j}^{a}\}) \tag{7}$$

이러한 최대 Q값을 구하는 것은 삼각형의 최대와 최소 점들 만 고려하면 된다. 우선, 정의에 의해 삼각형의 최대와 최소점사이의 $Q_{i,j}^{a}$ 은 선형직선을 이루고, $m_{i,j}$ 값은 상수이므로 선형직선 역시 $\mu_{i,j}Q_{i,j}^{a}$ 선형직선이 된다. 또한, 각 삼각형의 최대와 최소 점들을 순서대로 나열하여 이루어진 점들의 집합에서 이웃하는 2점사이의 사이에 존재하는 각 삼각곡선의 선분들은 선형이므로 이들의 합은 선형직선이고 따라서 집합내의 모든 점들 사이에는 선형선분이 존재한다. 이러한 여러 삼각형의 합으로 이루어진 곡선의 최대와 최소 점은 각 삼각형의 최대 최소 점들만 고려하여 쉽게 얻을 수 있다. 따라서, 식 (8) 의 해를 구할 수 있다.

현재 행위를 수행한 후, 현재 상태는 다음 상태로 변하고, 현재 상태에서 수행된 행위에 대한 보답을 받게 된다. 다음으로 Q-value 갱신식에 의해 다음 iteration에서 사용하게 될 현재상태의 Q-value가 계산된다. 이렇게 계산된 Q-value에 근거하여 삼각형의 Q-value모델을 좌우 혹은 위쪽으로 조정하여 현재 행위에 대한 Q-value가 Q-value 모델에 의해 표현될 수 있도록 한다. 따라서 현재 상태에서의 최적의 행위도 역시 이에 따라 조정되어 진다. 즉, 갱신된 Q-value 에 의해 Q-value 모델을 고치는 것은 2가지 형태가 존재할 수 있다. 첫 번 째로, 갱신된 Q-value가 현재 가장 큰 Q-value보다 작으면, 현재 행위의 Q-value가 Q-value모델의 Q-value보다 낮은 경우는 현재 행위에서 멀어지는 방향으로, 또는 현재 행위의 Q-value가 Q-value모델의 Q-value보다 높은 경우는 현재 행위에 가까워지는 방향으로 조정하여 Q-value모델에서 현재 행위의 Q-value가 실제로 갱신된 Q-value를 만족할 수 있도록 한다. 이러한 조정은 각 행위 축별로 수행되며, 다음 식(8)로 나타낼 수 있다.

$$a_{i,j}^{k}(t+1) = a_{i,j}^{k}(t) + \eta\,\mathrm{sgn}(a_{i,j}^{k}(t) - a_{i}^{k}(t)) \tag{8}$$

여기서, $\eta = \frac{2a_{\max}^{k}}{Q_{\max}}\left|dQ_{i,j}^{a_i}(t+1)\right|$

식(8)에서 는 k번째 행위축의 갱신할 행위의 변위를 결정하기 위해 사용된다. 또한 sgn() 행위의 변위 방향을 결정한다. 두 번째로, 만일 갱신된 Q-value가 현재 상태의 Q-value 모델의 최대 Q-value보다 크면, 식 (9)에서와 같이 최대 Q-value를 의미하는 현재상태 최적의 행위는 현재 행위로 대체된다.

$$a_{i,j}^{k}(t+1) = a_{i}^{k}(t) \tag{9}$$

이러한 Q-value 모델의 갱신은 그림 3에 나타내었다.

앞에서 기술한 Q-value 모델에 근거한 최적 행위 갱신을 포함한 전체적인 MQ-Learning Algorithm은 다음과 같이 요약할 수 있다.

MQ-learning 알고리즘

[초기화]

1. 초기화:여러 파라메타(γ , α , ρ)의 초기화

[반복]

2. 현재 상태를 받아들임 (s 현재상태)
3. 상태 s와 주변 상태와의 관계를 식 (8)를 이용하여 구한다.
4. 상태 s에서 실행해야 할 행위는 식 (9)에 의해 구한다. (때로는 Random action 수행)
5. 결정된 행위를 수행하고, Reward를 받는다.
6. 주변 상태들에서의 Q값은 식 (8)를 사용하여 구한다.
7. 주변 상태들에서의 최적의 행위 및 Q값을 식 (8),(9)을 사용하여 갱신한다.

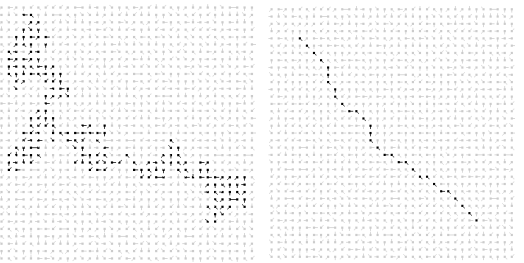
Ⅳ. Simulation Results

초기 위치를 (50,50)으로 하고 최종 위치 (320,320)라 할 때, Q-learning의 경우 position 상태 공간을 각 축마다 35개의 resolution으로 나누어야 목표 지점에 도달할 수 있다. 그림4.(a), 5.(a)은 아무런 사전 정보 없이 이동하는 초기 iteration에서는 여러 방향으로 탐색하는 과정을 보여준다. 그러나 그림 4.(b)에서와 같이 Q의 경우 약 400번의 iteration을 수행한 뒤에 원하는 상태 근처로 수렴하는 반면에, FQ의 경우 그림 5.(b)에서와 같이 47번의 iteration후에 수렴하는 것을 볼 수 있다. 또한 Q와 MQ의 iteration 별 step수는 약 40번 내외로 수렴됨을 알 수 있었다. MQ-learning 시뮬레이션 결과 우수한 수렴 속도와 Q-learning보다 부드러운 행위 집합으로 학습함을 알 수 있었다.

이와 더블어, 2 자유도(DOF)를 갖는 SCARA 로보트에 대해 시각 추적 작업을 실시하였다. 이를 위해 4개의 특징들로(로보트 좌표계에서 본 각 Robot 팔의 각도, 화면 좌표계에서 본 물체의 x, y 속도 성분) 이루어진 4-D의 상태 공간이 정의되었다. 또한 행위 공간은 각각 Robot 팔의 각속도로 정의 하였다. 여기서, 보답은 다음 식 (10)과 같이 정의 되었다.

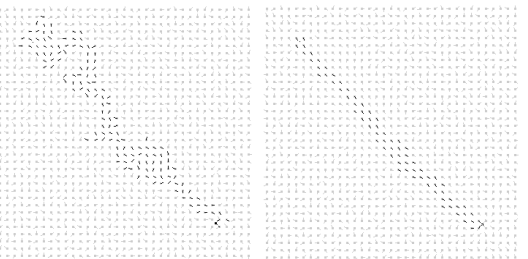
$$r = \frac{(\|\boldsymbol{x}_{i+1} - \boldsymbol{x}_g\| - \|\boldsymbol{x}_i - \boldsymbol{x}_g\|)}{K} \qquad (10)$$

여기서, $\boldsymbol{x}_i$ 와 $\boldsymbol{x}_{i+1}$ 는 각각 현재 상태와 다음 상태를 나타내며, $\boldsymbol{x}_g$ 는 최종 목표 상태를, K는 Robot 팔이 최대 각속도로 움직이어 변할 수 있는 상태 변이를 나타낸다. 매 샘플링 시에 로보트는 본 논문에서 제안하였던 MQ-learning을 이용하여 이동하는 목표 물체 따라 움직이는 것을 학습하게 된다. 본 시뮬레이션에서는 평균 1500~2000정도의 iteration을 거친 후 로보트가 최적의 행위를 수행함을 알 수 있었다.



(a) (b)

그림 4. The Q-learning 시뮬레이션 결과: (a)1st iteration (b) 400th iteration



(a) (b)

그림 5. MQ-learning 시뮬레이션 결과: (a)1st iteration (b)47th iteration.

Ⅴ. 결 론

본 논문에서는 연속 상태 공간에서 각 상태에서의 적절한 연속된 행동 양식을 학습하기 위해서 MQ-learning 알고리즘을 제안 하였다. 또한 이의 효율성을 증명하기 위해 기존의 Q-learning 알고리즘과의 비교 모의실험을 수행하였으며, 실제 상황과 동일하게 모델링된 로보트의 시각 추적(visual tracking) 시뮬레이션을 수행하였다. 시뮬레이션 결과, 본 논문에서 제안 하였던 방법은 연속 상태와 행위 공간을 가정하더라도 비슷한 수준의 수렴 속도를 나타내며, 특히, 환경에 대한 정보가 부족한 실제 상황, 즉 연속 공간에 대한 연속된 행위를 수행하여야 하는 시각 추적(visual tracking)과 같은 응용 분야에 효과적으로 적용될 수 있음을 보였다.

참 고 문 헌

〔1〕 C. Watkins, P. Dayan, Q-learning, technical note, Machine Learning, Vol. 8, pp.279-292, 1992.
〔2〕 C. Watkins, Learning from delayed rewards, Ph.D. Thesis, University of Cambridge, England, 1989.
〔3〕 P. Y. Glorennec, Fuzzy Q-learning and dynamical fuzzy Q-learning, IEEE Conference on R&A, vol. 1, pp. 474-479, 1994.
〔4〕 H. R. Berenji, Fuzzy Q-learning: A new approach for fuzzy dynamic programming, IEEE Conference on R&A, vol. 1, pp. 486-491, 1994.
〔5〕 T. Horiuchi, A. Fujino, O. Katai, and T. Sawaragi, Fuzzy interpolation-based Q-learning with continuous states and actions, IEEE Conference on Fuzzy Systems, vol. 1, pp. 594-600, 1996.

김 영 준 (Young Jun Kim) 정회원

1986년 한양대 전자공학과 학사
1991년 한양대 전자공학과 석사
2001년 한양대 전자공학과 박사
1996년~2001년 혜천대학 정보시스템계열 조교수
2001년 9월~현재 인하공업전문대학 정보통신과 부교수

<관심분야> AI, 지능망, 영상처리, 통신정책, 보안. Mobile IP