

분석적 학습

승실대학교 이 수 원*

| | |
|---------------------|-----------------------|
| ● 목 | 차 ● |
| 1. 서 론 | 4.3 청킹에 의한 탐색제어지식의 학습 |
| 2. 매크로 연산자 | 5. 연구 동향 |
| 3. 설명기반학습 | 5.1 유용성 문제 |
| 4. 청 킹 | 5.2 귀납적 학습과의 결합 |
| 4.1 Soar | 6. 결 론 |
| 4.2 설명기반학습과 청킹과의 관계 | |

1. 서 론

학습(learning)은 새로운 지식을 습득하거나, 새로운 기술(skill)을 습득하거나, 또는 현재의 지식을 새롭게 구조화하는 과정이라고 볼 수 있다[Carb89]. 인공지능 분야에서는 계산적인 방법(computational method)으로써 학습의 모형을 표현하며, 흔히 기계학습(machine learning)이라는 용어를 사용하여 인지과학 분야에서 연구되는 인간의 학습(human learning)과 구별을 하기도 한다.

학습은 여러 가지 관점에서 분류될 수 있다. 첫째, 학습에 사용되는 지식의 종류에 따라 보기(example)에 의한 학습, 경험(experience)에 의한 학습, 영역이론(domain theory)에 의한 학습, 교사(instructor)에 의한 학습 등으로 분류되며, 둘째, 학습방법에 따라, 귀납적 학습(inductive learning), 분석적 학습(analytical learning), 사례 기반 학습(case-based learning), 유추(analogy)에 의한 학습, 유전자 알고리즘(genetic algorithm), 군집화(clustering), 및

신경망 학습(connectionist learning) 등으로 분류되며, 셋째, 학습의 결과에 따라, 지식 습득(knowledge acquisition) 및 기술 습득(skill acquisition) 등으로 분류된다.

본 고에서는 학습방법에 따른 분류 중 분석적 학습에 대하여 기술한다. 분석적 학습은 상대적으로 최근에 본격적인 연구가 시작되었으나, 현재 가장 활발히 연구되고 있는 분야 중의 하나이다. 분석적 학습은 일반적으로 과거 문제 해결(problem solving) 경험으로부터 연역적인 방법을 사용하여 지식을 습득하며, 습득된 지식은 새로운 유사한 문제를 해결하는데 길잡이로 적용되거나 영역지식을 더욱 효율적으로 적용하기 위한 탐색 제어 지식(search control knowledge)을 생성하는데 이용된다. 그러므로, 분석적 학습에 사용되는 지식은 문제해결 경험 및 영역 지식(domain knowledge)이 가장 큰 비중을 차지하게 된다. 분석적 학습은 새로운 개념을 습득한다기보다는 기존의 개념에 대한 새로운 구조적인 기술(structural description)을 습득함으로써 시스템의 정확성이나 일반성을 저하시키지 않으면서 효율성을 향상시키는 데 그 핵심이 있다. 따라서, 학습의 결과는 기존

*중신회원

의 개념에 대한 새로운 영역지식의 습득이라는 측면에서 지식 습득, 그리고 시스템의 효율성 향상의 측면에서 기술 습득, 이 두 가지 관점에서 생각될 수 있다.

분석적 학습은 매크로 연산자(macro operator)[FikeN71, FikeHN72, Korf85] 이후 본격적으로 연구되기 시작하였으며, 최근에는 설명기반학습(explanation-based learning)[MittK86, DeJoM86], 및 청킹(chunking)[LairRN86] 등의 학습방법이 주로 연구대상이 되고 있다. 매크로 연산자는 일련의 단순연산자(primitive operator)들에 의한 문제해결 과정을 하나의 복합연산자로 표현한 것으로서, 매크로 연산자를 습득하여 사용함으로써 단순 연산자를 탐색, 결정하는 과정을 줄일 수 있다. 설명기반학습은 하나의 긍정적 훈련사례(positive training example)를 일반화(generalization)시켜 목표 개념(goal concept)에 대한 새로운 기술(description)을 습득하는 학습방법이다. 청킹은 Soar[LairNR87, RoseLN91]라는 통합적 인공지능 구조에 포함된 학습방법으로서, 설명기반학습과 거의 유사하나 Soar의 문제해결 메커니즘 및 규칙기반구조와 밀접하게 연관되어 있다는 데에 그 특징이 있다.

본 고의 제 2장에서는 매크로 연산자에 대해서, 제 3장에서는 설명기반학습에 대해서, 제 4장에서는 청킹에 대하여 기술하고, 제 5장에서는 분석적 학습에서 해결되어야 할 문제 및 현재 연구 동향에 대하여 기술한다.

2. 매크로 연산자

매크로 연산자의 습득 및 사용에 대한 연구는 STRIPS[FikeN71, FikeHN72]에서부터 체계화되기 시작하였다. 일반적으로 연산자(operator)는 현재상태(current state)에 적용되어 이를 다음 상태로 바꾸어 주는 동작(action)을 표현한다. 매크로 연산자의 기본 개념은 현재상태로부터 목표상태(goal state)를 성취하는 일련의 단순연산자를 표 1과 같은 매크로표(macro table)등으로 표현하여, 단순연산자가 수행될 곳에 습득된 매크로 연산자를 사용함으로써 시스템이 다음 연산자를 결정하는 과정을 줄이고, 따라서 전반적인 성능의 향상을 가져올 수 있다는데 그 핵심이 있다.

STRIPS 이후의 연구로서 Korf는 매크로 연산자가 루빅스의 입방체(Rubik's cube) 문제와 같은 난제에 어떻게 적용될 수 있는지를 보인 바 있으며[Korf85], 이에 대한 정량적 분석을 시도한 바 있다[Korf87]. 분석의 결과를 요약하면, 종속목표(subgoal)들이 선형적으로 분해 가능하고(serially decomposable) 상태들이 각각 k 개의 이하의 값을 갖는 m 개의 상태변수로 구성되었다면 (즉, 총 상태의 가지수 $n = O(k^m)$), 임의의 초기상태로부터 단일의 목표상태를 탐색 없이 성취하기 위해서는 $O(km)$ 개의 매크로 연산자로 충분하다는 것이다. 또한, 종속목표들이 선형적으로 분해 가능하지 않은 경우에도 $O(\sqrt{n})$ 개의 매크로 연산자로

표 1 Eight Puzzle에 대한 매크로표([Korf85]에서 인용). 각 매크로 연산자는 u(up), d(down), l(left), r(right) 네개의 단순연산자의 선형조합으로 구성되어 있다

| | | Tiles | | | | | | |
|---|---|-------|--------------|------------|----------------|----------|---------------|-------|
| | | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| p | 0 | | | | | | | |
| o | 1 | ul | | | | | | |
| s | 2 | u | rdlu | | | | | |
| i | 3 | ur | dlurdlu | dlur | | | | |
| t | 4 | r | ldrurdlu | ldru | rdllurdlu | | | |
| i | 5 | dr | uldrurdlurdl | lurdlldru | ldrulurddlur | lurd | | |
| o | 6 | d | urdlldrul | ulddru | urddlurdlurdl | uldr | rdllurdlldrul | |
| n | 7 | dl | rulddrul | druuldrdlu | rudrdluldrurdl | urdluldl | uldrurdlldur | urdl |
| s | 8 | l | drul | rullddru | rdluldrurdl | rulldr | uldrurdlldur | ruldr |

써 평균탐색을 $O(\sqrt{n})$ 으로 줄일 수 있다는 것이다.

이러한 일련의 연구들은 주로 선형 매크로 연산자(linear macro-operator)에 대한 것으로서, 그 이론적 및 실험적 연구 성과에도 불구하고 선형 매크로 연산자는 유용성(utility)에 문제점이 있다. 유용성 문제에 대하여 선형 매크로 연산자가 갖는 가장 근본적인 한계는 조건이나 순환과 같은 비선형 제어구조(nonlinear control structure)를 효율적으로 표현할 수 없다는데 있다. [ChenC86] 등에서는 비선형 매크로 연산자(nonlinear macro-operator)를 습득함으로써 일반성(generality)을 향상시키고 따라서 시스템의 성능을 향상시킬 수 있는 방법들이 제시되었다. 그러나, 그 단점으로는 비선형 매크로 연산자의 표현방법 및 습득과정에 대한 난이도(complexity)가 증가한다는 것이다. 제어구조가 극단적으로 복잡해지면 매크로 연산자를 습득하는 문제는 임의의 프로그램의 수행과정으로부터 그 프로그램을 역으로 합성하는 문제만큼 어려워지게 된다. 이는 학습에 필요한 요소들에 대한 표현이 복잡해지면 학습 과정에 대한 난이도가 증가한다는 이론과 부합된다. 따라서 좀 더 간단하고 모듈적이고 규칙에 기반을 둔 분석적 학습방법에 대한 연구의 필요성이 요구되었다.

3. 설명기반학습

설명기반학습(explanation-based learning : EBL)은 설명기반 일반화(explanation-based generalization : EBG) 라고도 불리며, 일반화(generalization)를 통한 학습방법이라는 점에

서 귀납적 방법 등과 공통점이 있다. 그러나, 귀납적 학습이 다수의 긍정적(positive) 및 부정적(negative) 훈련사례(training example)들로부터 귀납적 비약(inductive leap)을 통하여 이들 사례들과 부합하는 일반화된 개념을 습득하는데 반하여, 설명 기반 학습은 문제영역에 대한 지식을 기초로 하여 하나의 긍정적 훈련사례를 일반화시켜, 그 연역적 정당성(deductive justification)으로부터 목표가 되는 개념에 대한 새로운 정의를 습득한다.

설명 기반 학습을 위하여는 목표개념에 대한 기존의 정의 및 문제영역에 대한 지식이 필요하며, 주어진 훈련사례가 왜 목표가 되는 개념의 보기인지를 설명할 수 있어야 한다. 즉, 설명기반학습은 네가지의 지식 - 목표개념, 훈련사례, 영역이론, 및 동작화기준 - 으로부터 목표개념의 충분조건이 되고 동작화기준을 만족시키는 훈련사례의 일반화를 구하는 문제로 정의될 수 있다[MitcKK86, DeJoM86]. 그림 1은 설명기반학습에 필요한 지식의 네가지 유형이다.

이러한 지식이 주어졌을 때, 설명기반학습의 과정은 다음 세 단계로 나뉘어질 수 있다 :

- (1) 증명 : 영역이론을 사용하여 훈련사례가 목표개념의 보기인 것을 증명한다.
- (2) 설명 : 증명에 사용된 규칙만으로 이루어진 트리 형식의 설명구조를 구성한다.
- (3) 일반화 : 목표개념을 설명구조를 통하여 회기 시키면서 설명구조가 만족되는 정당화된 충분 조건을 구한다.

예를 들어 그림 2와 같은 문제를 생각하여 보자. 주어진 훈련사례 및 영역이론으로부터 연역될 수 있는 지식은 OBJ1이 OBJ2보다 가벼우므로 $\langle OBJ1, OBJ2 \rangle$ 쌍은 Safe-to-Stack(x, y)

-
- 목표개념(goal concept) - 학습하고자 하는 개념에 대한 기존의 정의
 - 훈련사례(training example) - 목표개념의 보기
 - 영역이론(domain theory) - 훈련사례가 왜 목표개념의 사례인지를 설명하는데 사용되는 규칙이나 사실의 집합
 - 동작화기준(operationality criterion) - 목표개념을 정의하는데 사용되어야 하는 술어(predicate) (현재의 목표개념은 동작화기준에 위배된다고 가정함)
-

그림 1 설명기반학습에 필요한 지식의 네가지 유형

- 목표개념: $\text{Not}(\text{Fragile}(y)) \vee \text{Lighter}(x, y) \Leftrightarrow \text{Safe-to-Stack}(x, y)$
- 훈련사례: $\text{On}(\text{OBJ1}, \text{OBJ2})$
 $\text{Isa}(\text{OBJ1}, \text{BOX}) \text{ Color}(\text{OBJ1}, \text{RED}), \text{Volumn}(\text{OBJ1}, 1), \text{Density}(\text{OBJ1}, .1)$
 $\text{Isa}(\text{OBJ2}, \text{ENDTABLE}) \text{ Color}(\text{OBJ2}, \text{blue})$
- 영역이론: $\text{Volumn}(p1, v1) \wedge \text{Density}(p1, d1) \Rightarrow \text{Weight}(p1, v1 * d1)$
 $\text{Weight}(p1, w1) \wedge \text{Weight}(p2, w2) \wedge \text{Less}(w1, w2) \Rightarrow \text{Lighter}(p1, p2)$
 $\text{Isa}(p1, \text{ENDTABLE}) \Rightarrow \text{Weight}(p1, 5) \text{ (default)}$
 $\text{Less}(0, 1, 5)$
- 동작화기준: 목표개념이 표현되는 술어(예, Volume, Color, Density 등)

그림 2 설명기반학습 문제의 보기([MitsKK86]에서 인용)

라는 목표 개념을 만족시킨다는 것이다. 즉,

$\text{Volume}(\text{OBJ1}, 1),$
 $\text{Density}(\text{OBJ1}, .1),$
 $\text{Volume}(p1, v1) \wedge \text{Density}(p1, v1) \Rightarrow \text{Weight}(p1, v1 * d1)$

으로부터 OBJ1의 무게는 0.1이 되며,

$\text{Isa}(\text{OBJ2}, \text{ENDTABLE}),$
 $\text{Isa}(p1, \text{ENDTABLE}) \Rightarrow \text{Weight}(p1, 5)$

로부터 OBJ2의 무게는 5가 되므로 OBJ1의 무게는 OBJ2의 무게보다 가볍고, 따라서 $\text{Safe-to-Stack}(\text{OBJ1}, \text{OBJ2})$ 은 연역적으로 증명이 가능하다.

이와 같은 증명과정을 의미망 형태의 설명구

조로 나타내면 그림 3과 같다. 이 설명구조의 하단부에는 훈련사례의 지식 중 증명에 사용된 특성들만 포함되며 불필요한 특성(예, Color)들은 배제된다. 이러한 특성들 및 사용된 영역이론은 훈련사례에 대한 정당화된 일반화 과정의 기초가 된다. 일반화 과정은 Waldinger[Wald77], Nilsson[Nils80] 등에서 제시된 회기 알고리즘의 변형된 형태로서, 오직 훈련사례에 관련된 술어들만으로써 목표개념의 충분조건이 표현된다. 또한 기존의 회기 알고리즘이 가능한 모든 변수바인딩(variable binding)을 고려하는데 비하여, 설명기반학습에서는 훈련사례의 설명에서 사용된 변수바인딩만을 고려한다는 차이점이 있다. 그림 3과 같은 설명구조에 회기 알고리즘을 적용하여 얻어진 Safe-to-Stack 에 대한 새로운 정의는 그림 4와 같다.

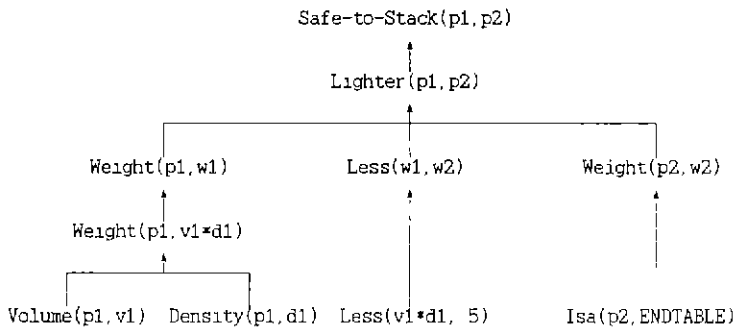


그림 3 Safe-to-Stack에 대한 설명구조

Volume(x, v1) \wedge
 Density(x, d1) \wedge
 Less(v1*d1, 5) \wedge
 Isa(y, ENDTABLE) \Rightarrow Safe-to-Stack(x, y)

그림 4 Safe-to-Stack에 대한 새로운 정의

설명기반학습은 귀납적 학습에 비하여 다음과 같은 장점을 가지고 있다. 첫째, 입력사례(input example)로부터의 귀납적 비약 없이 습득된 개념 기술에 대한 논리적인 정당성을 가지고 있다는 점이다. 둘째, 설명기반학습은 오직 하나의 긍정적 훈련사례만을 필요로 하며 부정적 훈련사례는 필요하지 않다는 점이다. 셋째, 개념에 대한 충분조건만 습득하므로 논리합(disjunctive) 개념을 처리할 수 있다는 점이다. 넷째, 훈련사례에 오류가 있을지라도 훈련사례가 목표개념의 보기인 것이 증명되지만 않으면 학습이 일어나지 않으므로, 훈련사례의 오류를 처리할 수 있다는 점이다. 반면에, 설명기반학습의 귀납적 학습에 대한 단점으로서 충분한 양의 영역이론이 필요하다는 점이다. 또한 설명기반학습에서는 개념공간에서의 탐색이 상대적으로 적게 요구되는 반면 증명공간에서의 탐색은 많이 요구된다는 점도 간과할 수 없는 문제이다.

설명기반학습에서는 목표개념을 탐색과정에서의 성공(success) 또는 실패(failure) 등으로 설정함으로써 탐색제어지식을 습득할 수도 있다. 탐색제어지식은 일반적으로 규칙의 형태로 표현되므로 선형 매크로 연산자와는 달리 비선형 제어구조를 표현할 수 있으며 그 구조 자체도 간단하다는 이점이 있다.

4. 청 킱

청 킱은 Soar라는 통합적인 인공지능구조에 포함된 학습방법이다[LairNR87, RoseLN91]. Soar는 문제공간가정(problem space hypothesis)을 기초로 하여, 문제해결, 계획, 추론, 학습, 자연어처리, 로봇틱스 등의 지능적 행위를 통합적, 유기적으로 구현할 수 있도록 하는 생

성시스템(production system)이다. Soar는 OPS5의 문법구조를 기본으로 하며, 1982년 미국 카네기멜론 대학(CMU)에서 Soar1이 개발된 이래, 현재 Soar6까지 구현되어 세계 30여 대학 및 연구소에서 인공지능의 여러 분야의 관점에서 연구, 확장되고 있다[RoseLN92]. Soar가 기존의 생성시스템과 다른 점은 선호, 종속 목표화, 청 킱 등 Soar 고유의 특징들을 기초로 하여 문제를 해결한다는 데 있다. 이들 특징들은 서로 독립적인 것이 아니라 상호간에 밀접한 연관성을 가지고 있으므로 청 킱에 대한 설명에 앞서 이들 특징들에 대하여 먼저 기술한다.

4.1 Soar

문제공간(problem space): Soar는 모든 목표 지향적 행위가 문제공간에 의하여 표현된다는 “문제공간가정”에 기초를 두고 있다[NeweEA91]. 문제공간은 상태(state)의 집합과 연산자(operator)의 집합으로 구성된다. Soar에서는 문제공간 내에서 연산자를 적용하여 초기 상태에서부터 목적 상태를 성취함으로써 문제를 해결한다. 문제공간가정에 대한 인지과학적인 기초는 [Newe90]에 기술되어 있다.

생성시스템(production system): Soar에서 지식은 장기적 기억장소(long-term memory) 및 일시적 기억장소(working-memory)에 저장된다. 장기적 기억장소는 변수화된 생성규칙(variabilized production-rule)의 집합으로 구성되며, 각각의 규칙은 전제조건/동작(condition/action)의 쌍으로 구성된다. 불변의 사실, 절차, 그리고 탐색 제어에 대한 정보 등은 모두 장기적 기억장소에 규칙의 형태로 저장된다. 일시적 기억장소는 실세계의 현재상태를 표현한다. Soar에서는 일시적 기억장소의 현재 내용을 장기적 기억장소의 각 규칙들의 전제조건과 부합(matching)시키고, 부합된 규칙들의 변수를 일시적 기억장소의 내용으로 사례화(instantiation)시켜, 사례화된 규칙의 동작으로써 일시적 기억장소의 내용을 변환한다.

선호(preference): 사례화된 규칙의 동작은 “선호”라는 Soar 고유의 자료 구조로써 표현된다. 선호의 종류에는 용인(acceptable), 거절(reject), 최상(best), 최하(worst), 상위(better), 하위

(worse), 무관(indifference), 재고(reconsider) 등이 있으며, 이들 선호들은 결정프로시듀어(decision procedure)에 의하여 일시적 기억장소의 내용을 어떻게 바꿀 것인지의 결정에 사용된다. 일시적 기억장소의 내용이 바뀌어지면 바뀐 내용으로부터 부합 및 결정프로시듀어를 반복적으로 적용하여 문제를 해결한다.

중속목표화(subgoaling): 만일 시스템 내의 현재의 지식이 불충분하여 일시적 기억장소 갱신에 대한 결정을 할 수 없는 경우에는 “난관(impasse)”이 발생한다. 난관에는 다수의 동일 선호를 가진 연산자가 제시되었을 때 생기는 선택난관(selection impasse), 현 상태에서 연산자가 더 이상 제시되지 않았을 때 생기는 발생난관(generation impasse), 및 제시된 연산자를 수행시킬 수 없을 때 생기는 수행난관(execution impasse) 등이 있다[RoseLU90]. 난관이 발생하면 Soar에서는 Soar의 중속목표를 자동적으로 생성하고, 중속목표 내에서 탐색을 통하여 발생된 난관을 해결한다. 이 경우 이전의 목표는 현재의 중속목표의 상위목표가 된다. 난관을 해결할 수 있는 충분한 지식이 탐색되면 중속목표는 성취되며, 그 결과로써 새로운 선호가 상위목표에 생성되고 이를 통하여 상위목표에서 발생한 난관이 해결된다.

칭킹(chunking): 중속목표가 성취되었을 때 Soar는 칭킹이라는 학습방법에 의하여 새로운 규칙을 생성하고 이를 장기적 기억장소에 저장한다. 새로운 규칙의 전제조건 부분은 중속목표의 결과를 생성하는데 사용된 상위목표의 일반화된 상황으로 이루어지며, 동작 부분은 중속목표의 일반화된 결과로 이루어진다. 새로운 규칙이 저장되면 이 규칙에 부합되는 유사한 상황이 발생하였을 때 난관이 발생하지 않고 바로 문제를 해결할 수 있게 된다.

4.2 설명기반학습과 칭킹과의 관계

칭킹 과정은 설명기반학습 방법과 거의 유사하나 Soar와 같은 생성시스템의 문제해결 방법과 밀착 결합(tightly coupled)되어 있다는 점에서 차이가 있다. 그림 5는 설명기반학습에서 사용되는 네가지 유형의 지식과 칭킹에서 사용되는 지식과의 대응관계를 보여주고 있다[RoseL

86]. 예를 들어, 그림 2와 같은 Safe-to-Stack 문제에 대하여, Soar는 훈련사례를 현재상태에 저장하고 영역이론을 장기적 기억장소에 문제공간별로 저장한 다음, 현재의 목표 Safe-to-Stack(x, y)을 성취하기 위한 연산자(예, Safety(x, y))를 수행하려 한다. 그러나, 주어진 훈련사례만으로는 Safety(x, y)를 바로 수행할 수 없으므로(즉, 동작화기준이 만족되지 않으므로) 수행난관이 발생하고 이에 따른 중속목표를 생성한다. 중속목표 내에서 Soar는 영역이론을 탐색하여 Safety(x, y)를 구현하게 되고 그 결과가 칭킹에 의하여 새로운 규칙으로 생성된다. 새로운 규칙의 형태는 Safety(x, y)를 구현하는 지식이며 이는 그림 4의 지식과 유사한 형태, 즉 Safe-to-Stack(x, y)을 훈련사례의 술어들을 일반화시켜 정의한 형태가 된다. 새로운 규칙이 습득되면 유사한 훈련사례에 대하여 Safety(x, y)가 바로 적용될 수 있으므로 수행난관이 발생하지 않는다.

설명기반학습 과정의 세 단계, 즉, 증명, 설명, 및 일반화 과정과 Soar에서의 칭킹 방법과의 대응은 그림 6과 같다.

| 설명기반학습 | 칭킹 |
|--------|------------------------|
| 목표개념 | ⇔ 목표 |
| 훈련사례 | ⇔ 중속목표 전의 상태 |
| 영역이론 | ⇔ 문제공간별 생성규칙 |
| 동작화기준 | ⇔ 목표들 정의하는데 사용되어야 할 술어 |

그림 5 설명기반학습에 사용되는 지식과 칭킹에서 사용되는 지식의 대응

| 설명기반학습 | 칭킹 |
|--------|------------------|
| 증명 | ⇔ 문제해결 |
| 설명 | ⇔ 점화된 생성규칙들의 역추적 |
| 일반화 | ⇔ 역추적 및 변수화 |

그림 6 설명기반학습의 과정과 칭킹과정과의 대응

4.3 칭킹에 의한 탐색제어지식의 학습

4.2의 보기에서는 칭킹에 의하여 목표개념 Safe-to-Stack(x, y)에 대한 새로운 정의를 습득하는 과정을 기술하였다. 칭킹에 의하여 습득

될 수 있는 다른 형태의 지식으로는 탐색제어 지식을 들 수 있다. Soar에서 목표개념에 대한 새로운 정의가 연산자 ‘수행난관’으로부터 습득되는 것과는 달리 탐색제어지식은 연산자 ‘선택난관’으로부터 습득된다. 탐색제어지식은 일반적으로 연산자 제시 규칙(operator proposal rule)에 최상, 최하, 상위, 하위 등의 선호를 첨가하여 표현되며, 선택난관은 제시된 연산자가 모두 디폴트선호인 ‘용인’을 가질 때 발생한다. 선택난관에 대한 종속목표의 역할은 상위목표에서 제시된 연산자를 선택을 할 수 있도록 하는 새로운 선호, 즉 탐색제어지식의 습득이다. 이를 위하여 종속목표에서는 ‘성공’ 및 ‘실패’ 등과 같이 탐색경로를 평가할 수 있는 기준으로써 그 목표개념을 설정한다.

예를 들어, 그림 7과 같이 현재상태 및 목적상태가 주어졌을 때 현재상태로부터 목적상태를 성취하는 연산자의 순서를 찾는 계획(planning) 문제를 생각해 보자. 목적상태는 세계의 계획목표(planning goal) On(A, B), On(B, C) 및 On(C, D)로 구성되어 있다. 만일 어떤 계획목표가 우선적으로 성취되어야 하는지에 대한 지식이 현재의 일시적 기억장소에 없다면 [Velo89] 등과 같은 비선형계획기에서는 세계의 계획목표를 동시에 고려하게 되고, 따라서 각각의 계획목표를 성취할 수 있는 연산자 MOVE(A, B), MOVE(B, C), 및 MOVE(C, D)를 디폴트선호인 ‘용인’을 첨가하여 제시하게 된다. 이 경우 제시된 세계의 연산자는 모두 용인 선호만을 가지므로 선택난관이 발생한다.

| 현재상태 | 목적상태 |
|--------------|----------|
| On(A, Table) | On(A, B) |
| On(B, Table) | On(B, C) |
| On(C, Table) | On(C, D) |

그림 7 4-Block-Stacking 문제의 보기

이 난관에 의하여 생성된 종속목표 내에서는 세계의 연산자를 무작위로 하나씩 선택하여 탐

색공간 내에서 평가하게 된다. 이 경우 만일 MOVE(A, B)가 먼저 선택되어 탐색공간 내에서 수행되고 그 다음 MOVE(B, C)가 선택되었다면 ‘보호성(Protection)’이라는 제약을 위반하게 되어 평가결과가 실패가 된다[RoseLU93, LeeR92]. 보호성이란 한번 성취된 계획목표는 전체 목적상태가 성취될 때까지 성취된 채로 남아 있어야 한다는 제약으로서, 계획시간을 단축하고 짧은 경로의 계획을 찾을 수 있게 해 주는 긍정적인 제약이다. 즉, MOVE(A, B)가 먼저 성취된 경우, MOVE(B, C)를 성취하기 위해서는 그 전에 MOVE(A, B)에 의하여 성취된 계획목표 On(A, B)를 다시 미성취된 상태로 바꾸어 주어야 하므로 보호성에 위반된다. 따라서 현재의 Soar 종속목표의 결과(즉, 실패)로부터 상위목표에서 제시된 연산자 MOVE(A, B)의 선호는 최하가 되며, 이에 대한 일반화된 탐색제어규칙이 습득된다. 그림 8은 MOVE(A, B)가 평가되고 나서 습득된 탐색제어규칙 chunk-1을 보여준다. (<x>는 x가 변수라는 의미이다).

chunk-1.

현재의 문제공간은 Blocks-world
 On(<x> <y>)는 미성취된 계획목표
 On(<y> <z>)는 미성취된 계획목표
 Block(<x>)
 Block(<y>)
 Clear(<x>)
 Clear(<y>)
 MOVE(<x> <y>)는 제시된 연산자
 ⇒
 MOVE(<x> <y>)의 선호는 최하

그림 8 Blocks-world에서 습득된 탐색제어지식의 예

여기서 한가지 유의하여야 할 점은 MOVE(A, B)의 평가결과에 의하여 생성된 chunk-1은 변수로써 일반화되어 있다는 점이다. 따라서 chunk-1은 연산자 MOVE(B, C)에도 적용이 되어 그 연산자에 최하선호를 첨가한다. 그 이유는 chunk-1의 변수 중 <x>는 B에, <y>는 C

에, <g>는 D에 각각 바인딩이 가능하며 On(C, D)가 아직 미성취된 계획목표이므로, On(B, C)를 성취하기 위한 연산자 MOVE(B, C)가 먼저 수행이 된다면 On(C, D)를 성취하기 위하여 On(B, C)에 대한 보호성을 위반하여야만 하기 때문이다. 이와 같이 문제해결 과정에서 습득된 지식을 그 문제 내에서 사용하는 것을 '문제해결 중 전이(within-trial transfer)'라 한다[LairRN86]. Soar처럼 선호를 사용하고 설명기반학습의 기능을 갖춘 PRODIGY/EBL [Mint88, MintEA89a, MintEA89b] 등에서는 문제해결 중 전이가 불가능하다는 점에서 차이가 있다.

그림 9는 그림 7과 같은 문제에서 획득된 탐색제어규칙 chunk-1이 다른 유사한 문제에도 적용될 수 있음을 보여주고 있다. 즉, 상태-1과 같은 초기상태로부터 상태-5와 같은 목적상태를 성취하는 문제에서 chunk-1과 같은 1개의 탐색제어 지식만으로 탐색 없이 문제를 해결할 수 있음을 보여준다. 상태-1에서 MOVE(D, E)를 제외한 나머지 연산자는 모두 chunk-1에 의하여 최하선호를 가지므로 MOVE(D, E)가

선택되어 수행되고, 그 결과로서 현재상태가 상태-2로 바뀌게 된다. 상태-2에서 제시된 연산자 MOVE(C, D)는 On(D, E)가 이미 성취되었으므로 더이상 chunk-1의 전제조건과 부합하지 않는다. 따라서 chunk-1의 동작에 의하여 첨가된 최하선호는 '철회(retract)'되고 MOVE(C, D)가 유일하게 최하선호를 가지지 않으므로 선택, 수행된다. 현재상태에 더 이상 부합되지 않는 규칙에 의하여 생성된 지식을 일시적 기억장소로부터 철회하는 이와 같은 방법은 JTMS와 그 개념에 있어서 유사성이 있다고 할 수 있다.

그림 9의 보기에서처럼 과거 문제의 해결과정 또는 결과로부터 습득된 지식을 새로운 유사한 문제에 전이하는 방법을 '문제간 전이(across-problem transfer)'라 한다. 특히, 특정한 크기의 문제로부터 습득한 지식으로써 임의의 크기의 같은 형태의 문제를 해결할 수 있을 때 이를 'N으로의 일반화(generalization-to-N)'라 한다. 문제간 전이에서의 한 가지 문제점은, 단일 '성공'이 목표개념이 될 경우 최적이 아닌 경로로 목표상태를 성취하여도 보호성과 같은 제

| 상 태 | 미성취된 계획목표 | 제시된 연산자 (선호) | 선택된 연산자 |
|--|--|--|------------|
| 상태-1: On(A, TABLE) On(B, TABLE) On(C, TABLE) On(D, TABLE) | On(A, B) On(B, C) On(C, D) On(D, E) | MOVE(A, B) (응인, 최하) MOVE(B, C) (응인, 최하) MOVE(C, D) (응인, 최하) MOVE(D, E) (응인) | MOVE(D, E) |
| 상태-2: On(A, TABLE) On(B, TABLE) On(C, TABLE) On(D, E) | On(A, B) On(B, C) On(C, D) | MOVE(A, B) (응인, 최하) MOVE(B, C) (응인, 최하) MOVE(C, D) (응인) | MOVE(G, D) |
| 상태-3: On(A, TABLE) On(B, TABLE) On(C, D) On(D, E) | On(A, B) On(B, C) | MOVE(A, B) (응인, 최하) MOVE(B, C) (응인) | MOVE(B, C) |
| 상태-4: On(A, TABLE) On(B, C) On(C, D) On(D, E) | On(A, B) | MOVE(A, B) (응인) | MOVE(A, B) |
| 상태-5: On(A, B) On(B, C) On(C, D) On(D, E) | | | |

그림 9 chunk-1을 이용한 5-Block-Stacking 문제의 해결

약만 위배되지 않으면 그림 8과 같은 연산자 제어규칙에 '최상' 선호가 첨가되어, 유사한 문제가 주어졌을 때 반복적으로 최적이지 아닌 경로를 택한다는 점이다. 이는 [TambR93]에서 언급된 메스킹(masking)효과 문제로서 칭킹뿐만 아니라 일반적인 설명기반학습 계열의 학습방법에서 생길 수 있는 문제이다. [RoseLU93]에서는 칭킹의 목표개념으로써 '실패'만을 사용하여 메스킹 문제를 제어하였다.

칭킹에 의해 습득된 탐색제어지식은 그 표현성에 있어서 특성이 있다. 즉, 일반적으로 계획을 표현하는 자료구조인 완전순서계획(total-order plan) 또는 부분순서계획(partial-order plan) 등이 주어진 문제에 대한 계획을 명시적으로 표현하는데 반하여[AlleHT90], 탐색제어지식은 계획을 암시적으로 표현한다고 할 수 있다. 예를 들어 그림 7과 같은 문제가 입력으로 들어오면 일반적인 계획 시스템에서는 MOVE(C, D) → MOVE(B, C) → MOVE(A, B) 등과 같은 완전순서계획, 또는 이에 상응하는 부분순서계획을 출력하지만, 칭킹을 이용한 계획 시스템에서는 그림 8과 같은 탐색제어지식을 출력(즉, 장기적 기억장소에 저장)하게 된다. 이러한 계획표현 방법은 부합과정이 필요하다는 단점이 있으나, 변수화 되어 있으므로 문제 간 전이가 용이하고, 또한 여러 유형의 탐색제어지식을 조합하여 새로운 복잡한 문제에 대해서도 다음 연산자를 효과적으로 제시할 수 있다는 장점이 있다.

5. 연구 동향

5.1 유용성 문제

일반적으로 분석적 학습은 목표개념에 대한 동작화된 정의 또는 탐색제어지식을 습득함으로써 시스템의 효율성을 증가시킨다. 그러나, 학습된 지식을 부합시키는데 드는 비용(cost)이 오히려 그 규칙을 사용함으로써 절감된 비용을 능가하는 경우에는 문제가 생긴다. 이와 같은 문제를 유용성(utility) 문제라고 한다[Mint88]. 유용성 문제는 크게 비싼청크 문제(expensive chunk problem) 및 평균증가효과(average growth effect)로 나눌 수 있다.

비싼청크 문제는 하나의 학습된 규칙에 대한 부합비용이 지나치게 높은 경우에 생기는 문제인데, 학습된 규칙의 전체조건에 사용된 구체화되지 않은 다수의 변수가 일시적 기억장소의 내용과 지수적으로 많이 부합되는 경우에 발생할 수 있다[TambNR90]. 이에 대한 해결책으로는 첫째, 부합비용을 낮추는 방향으로 학습된 규칙의 전체조건을 재구성하는 방법이 있다[Mint88]. 이 방법은 부합비용이 낮은 규칙을 생성하는데는 효과적이거나, 비싼청크 문제의 모든 원인을 추출하여 해결하기에는 충분하지 못한 방법이다. 또한 재구성하는데 드는 비용이 무시할 수 있는 정도가 아니라면 또 다른 형태의 유용성 문제가 될 수 있다[KimR93]. 둘째 방법은 유용성 평가함수(utility evaluation function)를 사용하여 혜택이 비용보다 상대적으로 높은 규칙만 저장하고 나머지는 제거하는 방법이다[Mint88]. 이 방법 역시 유용성 평가함수를 학습된 규칙마다 적용하는데 무시할 수 없는 비용이 소모되며, 궁극적으로 제거될 규칙이라면 그 규칙을 학습하는데 드는 비용이 낭비된다는 문제가 있다. 셋째 방법은 규칙의 표현성(expressiveness)에 제약을 주어 부합비용의 상한선을 제한한다는 접근방법이 있다[TambNR90]. 이 방법이 가지고 있는 취약점은 표현성의 제약으로 말미암아 학습된 규칙의 일반성이 감소하고, 따라서 학습된 규칙의 갯수가 증가하거나 학습된 규칙의 전이성이 감소한다는 점이다. 그 밖의 방법으로는 학습된 규칙의 집합으로부터 순환적(iterative) 또는 재귀적(recursive) 구조를 추출하여 이를 장기적 기억장소에 저장하는 방법[Shav90, ShelC91], 및 문제해결과정에 사용된 탐색제어지식을 학습된 규칙에 첨가시킴으로써 학습된 규칙의 부합비용을 낮추는 방법[KimR93] 등이 있다.

유용성 문제의 또다른 문제인 평균증가효과는 부합비용이 비싼 규칙은 존재하지 않으나 규칙의 갯수가 많아지거나 규칙들 간의 상호관계로 말미암아 전체 시스템의 성능이 저하될 수 있다는 문제이다[DoorTN92]. 이 문제에 대한 일반적인 접근방법으로서 학습 전의 처리로서 선택적 주의(selective attention) 및 선택적 경험(selective experience), 학습 후의

처리로서 선택적 저장(selective storing), 선택적 기억(selective retention) 및 선택적 사용(selective utilization) 등을 들 수가 있다 [GreiL89, MarkS89, Moon89]. 또한, 최근의 연구로서는 Dispatcher-Soar에서 10,000개의 규칙을 학습하여도 전체 시스템의 성능이 규칙의 개수에 대하여 선형 증가만 한다는 실험 결과가 있다[DoorTN92].

5.2 귀납적 학습과의 결합

설명기반학습은 귀납적 학습 등 다른 학습방법과 결합하여 사용되기도 한다. mEBG(multiple example EBG) [Keda85]는 설명기반학습과 유사하나 최종적인 개념 정의가 다수의 사례들로부터 생성된다는데 차이가 있다. mEBG의 과정은 다음과 같다:

- (1) 각각의 사례에 대한 설명트리(explanation tree)를 생성한다.
- (2) 생성된 설명트리들을 비교하여 모든 사례에 의해 공유되는 최대의 부분트리(largest subtree)를 구한다
- (3) EBG 알고리즘을 적용하여 설명트리를 일반화한다.

IOE(induction over explanations) [FlanD89]는 다수의 사례들로부터 학습한다는 점에서 mEBG와 유사하다. 그러나 최대의 공통 설명트리를 생성하는 과정에서 IOE는 사례들 간의 상호 유사성에 의한 학습(similarity-based learning)을 한다는 점에서 차이가 있다. UNIMEM [Lebo85]에서는 유사성에 의한 학습에 의해 경험적 일반화를 습득한 다음 이를 영역이론으로써 설명하고 정당화한다. OCCAM [Pazz88]에서는 영역이론이 불완전한(incomplete) 경우, 불완전한 설명을 완성시키는 새로운 규칙을 경험적인 방법을 통해서 습득하고, 만일 영역이론이 부정확한(incorrect) 것이었다면 습득된 규칙을 수정하는 방법을 제시하였다.

6. 결 론

본 고에서는 분석적 학습방법 중 대표적인 방법인 설명기반학습, 청킹, 및 매크로 연산자의 습득에 관하여 고찰하였다. 매크로 연산자에

비하여 설명기반학습이나 청킹은 그 표현성 및 응용성에 있어서 더욱 일반적인 방법이라 볼 수 있다. 설명기반학습과 청킹은 개념적으로 유사하나 기초가 되는 문제해결 방법과의 밀착결합의 정도에 따라 구현상의 차이가 있으며, 그 응용으로서 목표개념에 대한 새로운 정의의 습득 및 탐색제어지식의 습득을 들 수 있다.

설명기반학습 및 청킹에서 연구되어야 할 문제는 유용성 문제가 있으며 그 외에도 영역이론이 불완전한 경우, 영역이론이 부정확한 경우, 영역이론이 완전하고 정확하나 이론공간이 너무 방대하여 목표개념을 한정된 시간 내에 설명하지 못하는 경우 등이 있다. 또한 분석적 학습방법과 경험적 학습방법을 결합하여 상호 보완적인 학습방법을 고안하는 접근방법으로도 많은 연구가 있다. 결론적으로 분석적 학습은 다른 학습방법과 마찬가지로 아직도 연구되어야 할 문제가 산재해 있는 분야로서, 이에 대한 연구는 당분간 인공지능에서 핵심 연구분야의 하나가 될 것으로 생각된다.

참 고 문 헌

- [AlleHT90] Allen, J., Hendler, J. and Tate, A. *Readings in Planning*. Morgan Kaufmann, 1990.
- [DoorTN92] Doorenbos, B., Tambe, M., and Newell, A. Learning 10,000 chunks : What's it like out there? In *Proceedings of AAAI-92*, pp. 830-836, 1992.
- [Carb89] Carbonell, J. G. Introduction : Paradigms for Machine Learning. *Artificial Intelligence* 40 : 1-9, 1989.
- [ChenC86] Cheng, P. W., and Carbonell, J. G. The FERMI system : Inducing iterative macro-operators from experience. In *Proceedings of AAAI-86*, pp. 490-496, 1986.
- [DeJoM86] DeJong, G. and Mooney, R. Explanation-based learning : An alternative view. *Machine Learning*, 1(2) : 145-176, 1986.
- [Etzi90a] Etzioni, O. Why PRODIGY/EBL works. In *Proceedings of the Eighth National Conference on Artificial Intelligence*, pp. 916-922, 1990.

- [Etzi90b] Etzioni, O. *A Structural Theory of Explanation-Based Learning*. Ph. D. Thesis, School of Computer Science, Carnegie Mellon University, 1990.
- [FikeN71] Fikes, R. E. and Nilsson, N. STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence* 2(3) : 189-208, 1971.
- [FikeHN72] Fikes, R. E., Hart, P. E., and Nilsson, N. J. Learning and executing generalized robot plans. *Artificial Intelligence* 3 : 251-288, 1972.
- [FlanD89] Flann, N. S., and Deitterich, T. G. A study of explanation-based methods for inductive learning. *Machine Learning* 4 : 187-226, 1989.
- [GratD90] Gratch, J. M. and DeJong, G. F. A framework for evaluating search control strategies. In *Proceedings of the Workshop on Innovative Approaches to Planning, Scheduling, and Control*, pp. 337-347, Morgan Kaufmann, 1990.
- [GreiL89] Greiner, R., and Likuski, J. Incorporating redundant learned rules : A preliminary formal analysis of EBL. In *Proceedings of IJCAI-89*, 1989.
- [Keda85] Kedar-Cabelli, S. T., and McCarry, L. T. Purpose-directed analogy : A summary of current research. *Proceedings of the Third International Conference on Machine Learning*, pp 80-83, 1985.
- [KimR93] Kim, J. and Rosenbloom, P. S. Constraining learning with search control. In *Proceedings of the Tenth International Conference on Machine Learning*, pp. 174-181, 1993.
- [Korf85] Korf, R. E. Macro-operators : A weak method for learning. *Artificial Intelligence* 26 : 35-77, 1985.
- [Korf87] Korf, R. E. Planning as search. *Artificial Intelligence* 33 : 65-88, 1987.
- [LairRN86] Laird, J. E., Rosenbloom, P. S., and Newell, A. Chunking in Soar : The anatomy of a general learning mechanism. *Machine Learning* 1 : 11-46, 1986.
- [LairNR87] Laird, J. E., Newell, A. and Rosenbloom, P. S. Soar : An architecture for general intelligence. *Artificial Intelligence*, 33(1) : 1-64, 1987.
- [Lair88] Laird, J. E. Recovery from incorrect knowledge in Soar. In *Proceedings of AAAI-88*, 1988.
- [LeeR92] Lee, S. and Rosenbloom, P. S. Creating and coordinating multiple planning methods. In *Proceedings of the Second Pacific Rim International Conference on Artificial Intelligence*, pp. 88-95, 1992.
- [LeeR93] Lee, S. and Rosenbloom, P. S. Granularity in multi-method planning. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, pp. 486-491, 1993.
- [Lebo85] Lebowitz, M. Concept learning in a rich input domain : Generalization-based memory. In R. S. Michalski, J. G. Carbonell and T. M. Mitchell (Eds.), *Machine Learning : An Artificial Intelligence Approach*. Vol. 2. Morgan Kaufmann, 1985.
- [MarkS89] Markovitch, S., and Scott, P. D. Utilization filtering . A method for reducing the inherent harmfulness of deductively learned knowledge. In *Proceedings of IJCAI-89*, 1989.
- [Mint88] Minton, S. *Learning Effective Search Control Knowledge: An Explanation-Based Approach*. Ph. D. Thesis, Computer Science Department, Carnegie Mellon University, 1988.
- [MintEA89a] Minton, S., James, G. C., Knoblock, C. A., Kuokka, D. R., Etzioni, O., Gil, Y. Explanation-based learning : A problem solving perspective. *Artificial Intelligence* 40 : 63-118, 1989.
- [MintEA89b] Minton, S., Knoblock, C. A., Kuokka, D. R., Gil, Y., Joseph, R. L., and Carbonell, J. G. *PRODIGY 2.0: The manual and tutorial*, Technical Report CMU-CS-89-146, School of Computer Science, Carnegie Mellon University, 1989.
- [MitcKK86] Mitchell, T. M., Keller, R. M., and Kedar-Cabelli, S. T. Explanation-based generalization : A unifying view. *Machine Learning*, 1(1) : 47-80, 1986.
- [Moon89] Mooney, R. J. The effect of rule use on the utility of explanation-based learning.

In *Proceedings of IJCAI-89*, 1989.

[Newe90] Newell, A. *Unified Theories of Cognition*. Harvard University Press, 1990.

[NeweEA91] Newell, A., Yost, G. R., Laird, J. E., Rosenbloom, P. S. and Altmann, E. Formulating the problem space computation model. In R. F. Rashid, editor, *Carnegie Mellon Computer Science: A 25-Year Commemorative*. Addison-Wesley/ACM Press, 1991.

[Nils80] Nilsson, N. J. *Principle of Artificial Intelligence*, Palo Alto, CA : Tioga, 1980.

[Pazz88] Pazzani, M. J. *Learning causal relationship: An integration of empirical and explanation-based generalizer*. Doctoral dissertation, Computer Science Department, University of California. Los Angeles, 1988.

[RoseL86] Rosenbloom, P. S. and Laird, J. E. Mapping explanation-based generalization onto Soar. In *Proceedings of the Fifth National Conference on Artificial Intelligence*, pp. 561-567, 1986.

[RoseLU90] Rosenbloom, P. S., Lee, S., and Unruh, A. Responding to impasse in memory driven behavior: A framework for planning. In *Proceedings of the Workshop on Innovative Approaches to Planning, Scheduling, and Control*, pp. 181-191, Morgan Kaufmann, 1990.

[RoseLN91] Rosenbloom, P. S., Laird, J. E., and Newell, A. A preliminary analysis of the Soar architecture as a basis of general intelligence. *Artificial Intelligence*, 47 : 289-325, 1991.

[RoseLN92] Rosenbloom, P. S., Laird, J. E., and Newell, A. (Eds.) *The Soar Papers*, MIT Press, 1992.

[RoseLU93] Rosenbloom, P. S., Lee, S., and Unruh, A. Bias in planning and explanation-based learning. In S. Minton(Ed.), *Machine Learning Methods for Planning*, Morgan Kaufmann, 1993. Also available in Chipman S. and Meyrowitz A., (Eds.) *Machine Learning Induction, Analogy and Discovery*. Kluwer Academic Publishers, 1993.

[ShelC91] Shell, P., and Carbonell, J. Empirical and analytical performance of iterative operators. In *the 13th Annual Conference of The Cognitive Science Society*, pp. 898-902. Lawrence Erlbaum Associates, 1991.

[Shav90] Shavlik, J. W. Acquiring recursive and iterative concepts with explanation-based learning. *Machine Learning* 5 : 39-70, 1990.

[TambNR90] Tambe, M., Newell, A., and Rosenbloom, P. S. The problem of expensive chunks and its solution by restricting expressiveness. *Machine Learning*, 5(3) : 299-348, 1990.

[Velo89] Veloso, M. *Nonlinear problem solving using intelligent casual-commitment*. Technical Report CMU-CS-89-210, School of Computer Science, Carnegie Mellon University, 1989

[Wald77] Waldinger, R. Achieving several goals simultaneously, In E. Elcock & Michie, D. (Eds.), *Machine Intelligence* 8, London : Ellis Horwood, Limited, 1977.

이 수 원



1982 서울대학교 자연과학대학
계산통계학과 학사
1984 한국과학기술원 전산학과
석사
1984~1987 금성사 중앙연구소
주임연구원
1987~1988 서강대학교 전산학
과 강사
1994 University of Southern
California 전산학과 박사
1994~1995 USC/ISI 및 USC/
IRIS 연구원

1995~현재 숭실대학교 컴퓨터학부 전임강사
관심분야 : AI, Planning, Machine Learning, Agent Mo-
deling, Integrated AI Architecture 등
