사용자 데모를 이용한 관계적 개체 기반 정책 학습

(Learning Relational Instance-Based Policies from User Demonstrations)

박 찬 영 † 김 현 식 †† 김 인 철 †††
(Chan-Young Park) (Hyun-Sik Kim) (In-Cheol Kim)

요 약 데모-기반 학습은 사용자가 직접 작업을 시연함으로써 로봇에게 쉽게 새로운 작업지식을 가르칠 수 있다는 장점이 있다. 하지만 기존의 많은 데모-기반 학습법들은 상태공간과 정책들을 표현하기 위해 속성-값 벡터 모델을 이용하였다. 속성-값 벡터 모델의 제한성으로 인해, 이들은 학습과정의 효율성도 낮고 학습된 정책의 재사용성도 낮았다. 본 논문에서는 기존의 속성-값 모델 대신 관계적 모델을 이용하는 새로운 데모-기반 작업 학습법을 제안한다. 이 방법에서는 사용자 데모 기록에서 추출한 훈련 예들에 관계적 개체-기반 학습법을 적용함으로써, 동일 작업영역내의 다른 유사한 작업들에도 활용하기 용이한 관계적 개체-기반 정책을 유도한다. 이 관계적 정책은 (상태, 목표) 쌍으로 표현되는 임의의 한 상황에 대해 이것에 대응하는 하나의 실행동작을 결정해주는 역할을 한다. 본 논문에서는 데모-기반 관계적 정책 학습법에 대해 자세히 소개한 후, 로봇 시뮬레이터를 이용한 실험을 통해 이 학습법의 효과를 분석해본다.

키워드 : 데모-기반 학습, 관계적 모델, 개체-기반 정책, 전이학습

Abstract Demonstration-based learning has the advantage that a user can easily teach his/her robot new task knowledge just by demonstrating directly how to perform the task. However, many previous demonstration-based learning techniques used a kind of attribute-value vector model to represent their state spaces and policies. Due to the limitation of this model, they suffered from both low efficiency of the learning process and low reusability of the learned policy. In this paper, we present a new demonstration-based learning method, in which the relational model is adopted in place of the attribute-value model. Applying the relational instance-based learning to the training examples extracted from the records of the user demonstrations, the method derives a relational instance-based policy which can be easily utilized for other similar tasks in the same domain. A relational policy maps a context, represented as a pair of (state, goal), to a corresponding action to be executed. In this paper, we give a detail explanation of our demonstration-based relational policy learning method, and then analyze the effectiveness of our learning method through some experiments using a robot simulator.

Key words : Demonstration-Based Learning, Relational Model, Instance-Based Policy, Transfer Learning

· 본 연구는 경기도의 경기도지역협력연구센터사업의 일환으로 수행하였음
· 이 논문은 2009추계 인공지능연구회 워크샵에서 '사용자 데모를 이용한 관계적 개체-기반 정책 학습'의 제목으로 발표된 논문을 확장한 것임

† 학생회원 : 경기대학교 컴퓨터과학과
cyboys@kyonggi.ac.kr
†† 학생회원 : 경기대학교 전자계산학과
advance7@kyonggi.ac.kr
††† 종신회원 : 경기대학교 컴퓨터과학과 교수
kic@kyonggi.ac.kr
(Corresponding author임)

논문접수 : 2010년 2월 4일
심사완료 : 2010년 2월 23일

정보과학회논문지: 소프트웨어 및 응용 제37권 제5호(2010.5)

1. 서 론

최근 들어 로봇의 적용범위가 점차 가정이나 사무실과 같은 일상 생활공간으로까지 확대됨에 따라 어떻게 로봇에게 이러한 생활공간에서 요구되는 다양한 작업지식을 효과적으로 가르칠 것이냐 하는 문제가 큰 관심을 모으게 되었다. 그동안 로봇 작업 학습을 위해 강화학습이나 확률 그래프 모델, 신경망 등 다양한 학습 방법이 연구되어 왔으나, 이 학습방법들은 긴 시간동안 로봇 스스로 시행착오적 경험을 하도록 요구하거나 다량의 훈련 데이터를 미리 로봇에게 제공할 수 있어야 하는 등의 문제점을 가지고 있다. 이에 반해 데모-기반 학습법(learning from demonstration, demonstration-based learning) [1,2]은 비-전문가인 일반 사용자의 작업 시연만으로도 로봇에게 쉽게 새로운 작업지식을 가르칠 수 있다는 장점 때문에 이에 대한 연구가 활발히 진행되고 있다.

그동안 대부분의 데모-기반 학습법은 관찰된 사용자 데모와 이것으로부터 유도되는 행동정책(policy) 모두를 속성-값 벡터(attribute-value vector) 모델을 기초로 표현하였다[3]. 이러한 속성-값 모델은 상태 추상화(state abstraction) 기능이 약하고, 속성간의 관계(inter-attribute relation)나 사례간의 관계(inter-example relation)를 직접 표현할 수 없다[4,5]. 따라서 속성-값 모델에 기초한 데모-기반 학습은 일반적으로 학습과정의 효율성과 학습결과의 재사용성이 낮다. 본 논문에서는 기존의 속성-값 모델에서 관계적 모델(relational model)로 확장한 새로운 데모-기반 작업 학습법을 제안한다. 이 방법에서는 사용자 데모를 훈련 예(training examples)로 삼아 관계적 개체-기반 학습법(relational instance-based learning)을 적용함으로써 동일 작업영역에서 발생하는 유사한 다른 작업들에도 이용 가능한 관계적 정책을 유도한다. 이 관계적 개체-기반 정책(relational instance-based policy)은 (상태, 목표) 쌍으로 표현되는 임의의 한 상황(*context*)에 대해 이것에 대응하는 하나의 실행동작(action)을 결정해주는 역할을 한다. 본 논문에서는 데모-기반 관계적 정책 학습법에 대한 자세한 소개와 더불어 로봇 시뮬레이터를 이용한 실험을 통해 이 학습법의 효과를 분석해본다.

2. 데모-기반 관계적 정책 학습

로봇 작업 학습을 위한 한 번의 사용자 데모는 작업 목표를 달성할 수 있는 한 가지 동작 시퀀스만을 제공한다. 따라서 매우 큰 상태공간과 행동공간을 가지는 현실적인 환경에서 작업 목표를 달성할 수 있는 가능한 모든 작업절차를 사용자 데모로 제공하기는 어렵다. 그뿐만 아니라 현실 작업환경은 대부분 로봇의 인식과 동작의 불확실성을 필연적으로 포함하고 있어, 사용자 데모를 그대로 따라 수행한다고 하더라도 작업 목표를 달성할 수 없는 경우가 자주 발생한다. 따라서 로봇 작업 학습을 위해서는 효과적인 상태/행동 추상화(state/action abstraction)와 정책 일반화(policy generalization)가 필요하다.

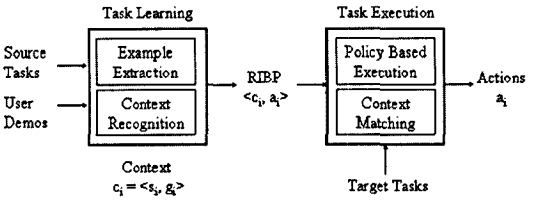

그림 1 데모-기반 관계적 정책 학습 과정

본 논문에서는 상태 추상화를 위해서 속성-값 모델 대신 관계적 모델(relational model)을 사용하고, 정책 일반화를 위해서 개체-기반 학습법(instance-based learning)을 적용한다. 그림 1은 본 논문에서 제안하는 데모-기반 관계적 정책 학습 과정을 간략히 묘사하고 있다. 작업 학습(task learning) 단계에서는 사용자가 로봇에게 가르치고 싶은 원천 작업(source task)의 목표를 제시하고, 이 목표를 달성하기 위한 작업절차인 동작들의 시퀀스를 로봇에게 시연한다.

로봇은 미리 정의된 관계적 상태모델에 기초하여 사용자 데모로부터 데모동작(action) a_i과 그 동작이 수행된 상황(context) c_i을 인식하여, 관계적 개체-기반 학습을 위한 훈련 예(example)들인$<c_i, a_i>$ 쌍들을 추출한다. 이때 하나의 상황 c_i는 현재 상태(state) s_i와 달성하고자 하는 작업 목표(*goal*) g_i의 쌍으로 표현된다.

작업 실행(task execution) 단계에서는 달성하고자 하는 작업(target task)의 목표가 주어지면 학습 단계에서 습득한 훈련 예 $<c_i, a_i>$들과 실행 상황 c'을 기초로 관계적 개체-기반 학습을 적용하여, 상황 c'에 적합한 실행동작들을 결정하고 이를 실행해 나간다. 이때, 관계적 개체-기반 학습은 지연 학습(lazy learning)의 하나로서, 훈련 예로부터 미리 전역적 상황-동작 대응함수를 생성하지 않는다. 대신, 목표 개체(target instance)인 하나의 실행 상황 c'이 주어질 때까지 기다렸다가 실행 상황 c'에 대해 관계적 거리(relational distance)가 가장 가까운 훈련 개체(training instance) $<c_i, a_i>$를 정하고, 이 훈련 개체에 따라 실행 상황 c'에 대응되는 실행 동작 a_i를 결정하는 학습방법이다. 따라서 작업 학습과 실행에 사용되는 관계적 개체-기반 정책(Relational Instance-Based Policy, RIBP) $\pi : C \rightarrow A$은 다음과 같이 정의할 수 있다.

$$\pi = <P, d>,$$

이때, P는 튜플 <c_i, a_i>들의 집합, d는 관계적 거리 척도(relational distance metric)를 각각 나타내며, $\pi(c) = \arg_{a_i}\min_{(<c_e a_i> \in P)} d(c, c_i)$이다. 즉, 관계적 개체-기반 정책 π는 관계적 거리 척도 d에 따라 훈련 예들 중에서 현재의 실행 상황 c와 관계적 거리가 가장 가까운 학습 상황 c_i를 정하고, 이 학습 상황 c_i에서 수행한 동작 a_i를 현재 실행 상황 c에서 실행할 동작으로 결정해주는 함수이다.

개체-기반 정책 학습을 위해서는 훈련 상황과 실행 상황간의 차이(difference) 혹은 거리(distance)를 계산할 수 있어야 한다. 훈련 상황과 실행 상황이 모두 속성-값 벡터 모델로 표현되는 경우에는 두 상황간의 거리는 k-최근접 이웃 방법(k-Nearest Neighbors, k-NN) [6]과 같이 단순히 다차원 속성 공간상의 유클리드 거리(Euclidean distance)를 계산하면 된다. 하지만 훈련 상황과 실행 상황이 모두 관계적 모델(relational model) 혹은 서술 논리 모델(predicate logic model)로 표현되는 경우, 두 상황간의 거리를 정의하는 것은 간단하지 않다. 본 논문에서는 Garcia[7]의 연구에서 제안한 관계적 거리 계산법을 확장한 새로운 거리 계산법을 제안한다. 이 관계적 거리 계산법은 동일 서술자(predicate)를 갖는 리터럴(literal)들의 인자(argument)로 사용되는 개체들 사이에 순서 관계(ordering relationship)가 존재하는 경우, 이를 고려하여 두 상황간의 관계적 거리를 계산하는 것이 특징이다. 본 논문에서 제안하는 순서-기반 관계적 거리 계산법은 아래와 같다.

주어진 도메인(domain)에는 서로 다른 K 개의 서술자(predicate)들이 존재한다고 가정하면, 두 상황간의 관계적 거리 $d(c_1, c_2)$는 두 상황 논리식에 공통으로 존재하는 동일 서술자들의 함수로서 식 (1)과 같이 정의할 수 있다.

$$d(c_1, c_2) = \sqrt{\frac{\sum_{k=1}^{K} w_k d_k(c_1, c_2)^2}{\sum_{k=1}^{K} w_k}} \quad (1)$$

이때, w_k는 각 서술자 p_k에 대한 가중치(weight)를 나타내며, $d_k(c_1, c_2)$는 각 서술자 p_k별로 계산되는 거리를 나타낸다. 동일 서술자(predicate)에 대해서도 인자(argument)가 다른 다수의 리터럴(literal)들이 존재할 수 있다. 예컨대, 동일한 서술자 user_in에 대해서 user_in(Tom, BedRoom2), user_in(Smith, Kitchen) 등과 같이 인자가 다른 다수의 리터럴들이 존재할 수 있다. 따라서 각 서술자 p_k별 거리인 $d_k(c_1, c_2)$는 식 (2)와 같이 정의할 수 있다.

$$d_k(c_1, c_2) = \frac{1}{N}\sum_{i=1}^{N} \min_{p \in P_k(c_2)} d'_k(P_k^i(c_1), p) \quad (2)$$

이때, $P_k(c_i)$는 논리식 c_i에 포함되어 있는 서술자 p_k를 공유하는 리터럴들의 집합을 나타내며, N은 집합 $P_k(c_1)$의 크기를 나타낸다. 또한, $P^i_k(c_1)$는 집합 $P_k(c_1)$의 i 번째 리터럴을 나타내며, $d'_k(p^1_k, p^2_k)$는 동일한 서술자 p_k를 갖는 서로 다른 리터럴들인 p^1_k와 p^2_k사이의 거리를 나타낸다. 서술자 p_k가 M 개의 인자(argument)들을 가진다면, $d'_k(p^1_k, p^2_k)$는 식 (3)과 같이 정의할 수 있다.

$$d'_k(p^1_k, p^2_k) = \sqrt{\frac{1}{M}\sum_{l=1}^{M} \delta(p^1_k(l), p^2_k(l))} \quad (3)$$

이때, $p^i_k(l)$은 리터럴 P^i_k의 l 번째 인자를 나타내고, $\delta(p^1_k(l), p^2_k(l))$는 l 번째 인자들 사이의 거리를 나타낸다.

만약 리터럴들인 user_distance(Tom, Contact)와 user_distance(Smith, Far)간의 거리를 계산할 때, 각각 첫 번째 인자들인 Tom과 Smith간의 거리는 Tom과 Michael간의 거리와 마찬가지로 동일하다고 볼 수 있으나, 두 번째 인자들인 Contact와 Far간의 거리는 Near와 Far간의 거리보다 더 멀다고 가정하는 것이 타당하다. 즉, Person 집합에 속한 개체들인 Tom, Smith, Michael 등의 개체들 사이에는 특별한 순서 관계(ordering relationship)가 존재하지 않는 반면, Distance 집합에 속한 Contact, Near, Far 등의 개체들 사이에는 하나의 순서 관계가 존재한다고 가정하면, 이들 간의 거리 계산은 이러한 개체들 간의 순서 관계를 고려하여야 한다. 이러한 가정에 기초하여 l 번째 인자들 사이의 거리인 $\delta(p^1_k(l), p^2_k(l))$를 식 (4)와 같이 정의할 수 있다.

$$\begin{aligned}\delta(p^1_k(l), p^2_k(l)) &= 0, \text{ if } p^1_k(l) = p^2_k(l) \\ &= \varepsilon, \text{ else if no ordering relationship} \\ &\quad \text{between } p^1_k(l) \text{ and } p^2_k(l) \\ &= |\mathrm{Order}(p^1_k(l)) - \mathrm{Order}(p^2_k(l))|, \text{ otherwise}\end{aligned} \quad (4)$$

이때, ε는 하나의 일정한 수치 상수(numeric constant)를 나타내고, $\mathrm{Order}(p^i_k(l))$는 l 번째 인자로 사용된 개체 $p^i_k(l)$가 동일 유형의 개체들 집합 S내에서 위치하는 순서를 나타낸다.

3. 로봇 작업 학습의 예

앞서 소개한 데모-기반 관계적 정책 학습법을 적용하기 위한 로봇 작업 환경을 Microsoft Robotics Developer Studio(MSRDS)[8]의 3D 로봇 시뮬레이터를 이용하여 그림 2와 같이 구현하였다. 그림 2의 로봇 작업 환경은 하나의 가상 가정환경으로서, 6개의 방, 10개의 가구 및 가전제품, 그리고 그 공간 안에서 활동하는 2명의 사용자와 하나의 학습 로봇으로 구성된다.

학습 로봇에게 가르치고자 하는 작업들은 그림 2에 나타낸 것과 같이 주로 가정 내 임의의 위치에서 시작하여 특정 위치에 있는 사용자를 찾아가는 이동 작업(navi-

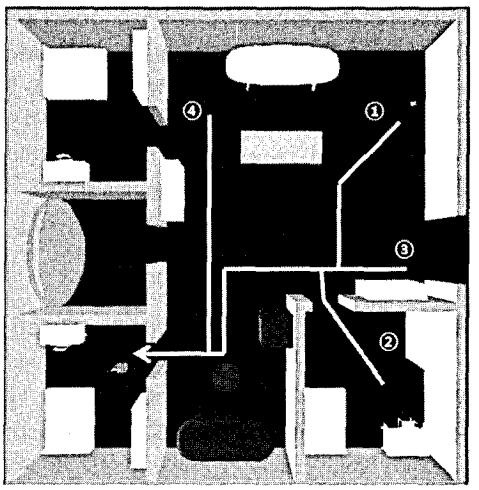

그림 2 로봇 작업 환경

gational task)들이라고 가정한다. 학습 로봇은 비젼처리를 통해 각 사용자와 랜드마크들(방, 가구, 대형 가전제품 포함)의 위치와 방향을 인식할 수 있고, 레이저 거리 측정기를 이용하여 주변의 장애물들과의 거리를 인식할 수 있도록 구현하였다. 또한, 직진 및 회전과 같은 단위 동작들을 수행할 수 있도록 구현하였다. 효과적인 작업 학습을 위해 사용자는 해당 작업의 목표를 명시적으로 제시할 뿐 아니라, 로봇이 수행할 다음과 같은 단위 동작들을 음성 명령(voice command)으로 로봇에게 제시함으로써 원격제어방식으로 작업데모를 시연한다.

A = {긴 직진(Forward_Long), 중간 직진(Forward_Medium), 짧은 직진(Forward_Short), 왼쪽 30도 돌기(Turn_Left), 왼쪽 15도 돌기(Left), 오른쪽 30도 돌기(Turn_Right), 오른쪽 15도 돌기(Right)}

그러면 학습 로봇은 작업 데모 과정동안 수행하는 자신의 동작과 함께 그때의 상황정보를 함께 기록함으로써 개체-기반 정책 학습을 위한 훈련 예들을 생성한다. 이때 학습 로봇은 센서 데이터에 대한 이산화(discretization)와 심볼 그라운딩(symbol grounding) 등 일련의 상황인식 과정을 수행한 다음, 해당 상황을 미리 정의된 서술자들을 이용하여 기록한다. 작업환경의 상태와 목표는 아래의 집합 L에 포함된 총 8 개의 서술자들을 이용하여 기술하는 것으로 정하였다.

L = {robot in (?R-Room),
room_distance (?R-Room ?RD-Room_Distance),
room_direction (?R-Room ?RI-Room_Direction),
user in (?U-User ?R-Room),
user_distance (?U-User ?UD-User_Distance),
user_direction (?U-User ?UI-User_Direction),
obstacle_distribution (?OI-Obstacle_Direction ?S-Distribution),
landmark_direction (?L-Landmark_ID ?LI-Landmark_Direction)}

각 서술자의 인자(parameter)에 대응되는 사물, 사용자, 그리고 로봇의 개체 수를 고려하면, 이 작업환경의 각 상태는 총 34 개의 서로 다른 리터럴들로 표현된다. 그림 2의 화살표들은 침실에 있는 Tom을 찾아가는 작업을 가르치기 위해, 사용자가 학습 로봇에게 데모로 시연해 보이는 다양한 작업경로들을 나타낸다.

4. 실험 및 평가

데모-기반 관계적 정책 학습법의 효과를 분석해보기 위해 3장에서 설명한 작업환경을 이용한 실험을 전개하였다. 실험의 첫 번째 목적은 본 논문에서 제안한 데모-기반 관계적 정책 학습법과 순서-기반 관계적 거리 계산법이 학습 대상인 원천 작업의 성공에 미치는 영향을 분석하는 것이고, 두 번째 목적은 원천 작업을 통해 학습한 관계적 정책이 다른 유사한 목표 작업들에도 적용가능한지 정책의 전이성을 분석하는 것이다. 첫 번째 실험에서는 아래의 G_1, G_2와 같이 서로 다른 작업 목표를 가진 원천 작업(source task) Task1과 Task2 각각에 대해 서로 다른 관계적 거리 계산법에 기초한 데모-기반 관계적 정책 학습법을 적용하여 작업 성공률의 변화를 측정하여 보았다.

G_1 = {user_in(Tom, Bedroom2), user_distance(Tom, Contact), user_direction(Tom, Front_Left)}
G_2 = {user_in(Tom, Bedroom2), user_distance(Tom, Near), user_direction(Tom, Right)}

각각의 작업에 대한 데모는 각기 다른 4 개의 초기 위치에서 시작하여 목표에 도달할 때까지 이루어졌고, 같은 경로에 대해 각각 5회까지 반복 데모를 수행하였다. Task1과 Task2는 각각의 작업경로가 평균 약 22 개의 연속 단위 동작들로 이루어진 비교적 길고 복잡한 작업들이다. 그림 3은 데모 횟수가 증가함에 따라 학습 후 작업실행의 성공률을 측정한 실험결과를 보여준다. 실험에서는 실행 상황과 작업 목표사이의 관계적 거리가 0.125 이내로 좁혀지면 작업을 성공적으로 완료한 것으로 판단하였다.

그림 3에서 Task(OrdRelDist)는 본 논문에서 제안한 순서-기반 관계적 거리 계산법을 적용한 경우의 작업 성공률을, Task(NOrdRelDist)는 Garcia의 방법처럼 순

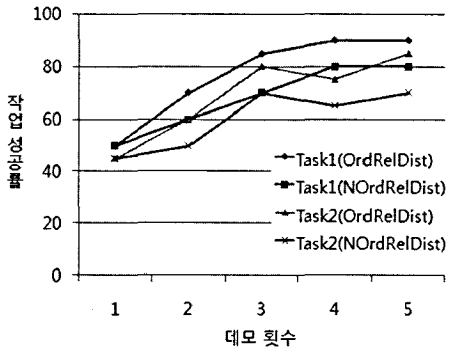

그림 3 원천 작업들의 성공률

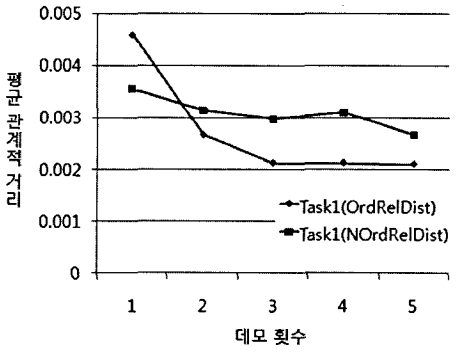

그림 4 가장 근접한 학습-실행 상황간의 관계적 거리

서에 무관한 관계적 거리 계산법을 적용한 경우의 작업 성공률을 각각 나타낸다. Task1과 Task2 두 작업 모두에서 약 3회의 데모가 이루어진 시점부터 70% 이상의 높은 작업 성공률을 보여주고, 특히 순서-기반 관계적 거리 계산법을 적용한 경우에는 5회의 데모만으로 90%가 넘는 작업 성공률을 보여준다. 이것을 통해 우리는 데모-기반 관계적 정책 학습법의 높은 학습 효율성과 작업 성능을 확인할 수 있다.

또, 동일한 작업에 대해 순서-기반 관계적 거리 계산법을 적용한 경우(OrdRelDist)가 그렇지 않은 경우(NOrdRelDist)보다 더 빠르게 작업 성공률이 향상되었음을 그림 3을 통해 확인할 수 있다.

한편, 그림 4는 Task1에 대한 학습 상황과 실행 상황간의 일치도를 최근접 학습-실행 상황간의 관계적 거리로 나타낸 것이다. 이 그림에서도 순서-기반 관계적 거리 계산법을 적용한 경우(OrdRelDist)가 그렇지 않은 경우(NOrdRelDist)보다 더 빠르게 학습-실행 상황간의 관계적 거리를 좁혀가는 것을 알 수 있다. 이와 같은 실험결과들을 통해 우리는 순서-기반 거리 계산법의 긍정적 효과를 확인할 수 있다.

두 번째 실험에서는 Task1과 Task2에서 학습한 관

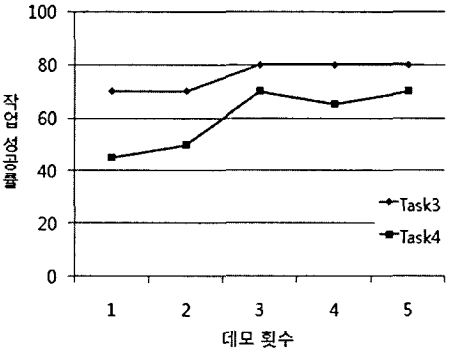

그림 5 다른 목표 작업들의 성공률

계적 개체 기반 정책을 같은 작업환경내의 다른 목표 작업(target task)인 Task3와 Task4를 달성하는데도 재사용 가능한지 실험하여 보았다. 새로운 목표 작업 Task3와 Task4의 작업 목표는 다음과 같다.

G_3 = {user_in(Smith, Bedroom2),
user_distance(Smith, Contact),
user_direction(Smith, Front_Left)}
G_4 = {user_in(Tom, Bedroom2),
user_distance(Tom, Contact),
user_direction(Tom, Front_Right)}

주목할 점은 Task3은 Task1과 비교했을 때 찾아가야 할 사용자가 Tom 대신 Smith로 바뀐 작업이며, Task4는 Task1, Task2 등과 비교했을 때 최종 목표 상태에서 요구되는 사용자와의 거리와 방향이 바뀐 작업이다.

그림 5는 원천 작업(source task) Task1과 Task2에 대한 데모 횟수가 증가함에 따라 목표 작업(target task) Task3과 Task4의 작업 성공률이 어떻게 변화하는지 그 추이를 나타내고 있다. 약 3회의 원천 작업에 대한 데모가 제공된 시점부터 두 목표 작업 Task3, Task4에서 각각 80%, 70%가 넘는 높은 작업 성공률을 기록하였음을 보여주고 있다. 이것을 통해 우리는 관계적 개체-기반 정책들이 유사한 다른 작업들에도 매우 효과적으로 이용될 수 있음을 알 수 있다.

5. 관련연구

사용자 데모로부터 로봇 작업 지식을 학습하는 대표적인 기존 연구 중의 하나인 Saunders의 연구[6]에서는 사용자의 작업 데모로부터 일반화된 작업 정책(task policy)을 얻기 위해 개체-기반 학습법의 하나인 k-최근접 이웃 방법(k-Nearest Neighbors, k-NN)을 이용

한다. 하지만 관계적 상태 모델을 이용하는 본 연구와는 달리, 이 연구에서는 로봇의 센서 데이터로 구성된 벡터 모델을 이용하여 상태를 표현함으로써 전체적으로 매우 큰 상태 공간을 갖게 되는 문제점이 있다.

Chernova의 연구[9]에서는 사용자 데모와 로봇 인식의 불확실성(uncertainty)을 고려하여 다수의 가우시안 혼합 모델(Gaussian Mixture Model, GMM)들을 이용하여 작업 정책을 표현하고 학습하였다. 가우시안 혼합 모델 기반의 작업 정책은 각 동작의 실행 확률을 계산해주는 가우시안 혼합 모델들의 출력을 결합하여, 현재 상태에서 실행해야 할 하나의 동작과 신뢰값(confidence value)을 결정해준다. 학습자인 로봇은 신뢰값이 충분히 높지 않은 경우에만 사용자에게 추가 데모를 요청하며, 새로운 훈련 예들을 확보함에 따라 가우시안 혼합 모델들을 점진적으로 새롭게 갱신한다. Chernova의 작업 학습법은 확률 기반의 정책 모델을 이용함으로써 학습과 연관된 불확실성 문제을 어느 정도 극복할 수 있으나, Saunders의 연구와 마찬가지로 상태 표현을 위해 속성-값 벡터 모델을 이용함으로써 학습의 효율성이 낮다. 또한, 정책 표현에 작업 목표를 포함하지 않아 특정 작업을 위해 학습된 정책을 다른 목표 작업에 재사용하는데 한계가 있다. 이에 반해 본 연구에서는 상태와 작업 목표를 모두 포함한 상황(context)을 기초로 정책을 표현함으로써, 동일한 환경에서 이루어지는 다양한 작업 데모들로부터 정책을 학습할 수 있을 뿐 아니라 유사한 다른 작업들에도 학습된 정책을 이용할 수 있다.

Veeraraghavan의 연구[10]에서는 사용자 데모로부터 단순히 하나의 작업 정책(task policy)을 학습하는 것이 아니라, 변수(variable), 순차구조(sequence), 순환구조(loop), 조건부 분기(conditional branch) 등을 포함한 프로시져 형태의 일반화된 작업 계획(generalized task plan)을 학습하려는 시도를 보였다. 이러한 작업 계획은 적용되는 실제 상황에 따라 실행 가능한 다양한 작업 계획들로 구체화됨으로써, 영역-의존적 계획기(domain-specific planner)로도 불린다. Veeraraghavan의 연구에서는 휴머노이드 로봇의 작업 학습을 위해 Winner의 계획 학습 알고리즘[11]을 이용하고 있다. 본 논문에서 제안한 데모-기반 관계적 정책 학습법을 위해서는 작업 데모에 쓰인 단위 동작들의 이름만 필요할 뿐 각 동작의 정형 모델을 요구하지 않는다. 하지만 Veeraraghavan의 작업 계획 학습법을 위해서는 각 동작의 전-조건(precondition)과 효과(effect)를 나타내는 동작 모델(action model)이 요구된다.

6. 결 론

본 논문에서는 서비스 로봇에게 효과적으로 작업 지식을 가르치기 위한 데모-기반 관계적 정책 학습법을 제시하였다. 이 학습법은 상태 추상화와 정책 일반화 능력이 우수하여, 학습과정의 효율성과 학습결과로서 작업 성공률이 높다. 또한, 원천 작업들을 통해 학습된 관계적 개체-기반 정책은 같은 작업 환경내의 유사한 다른 작업들에도 효과적으로 재사용될 수 있음을 실험을 통해 확인하였다. 이러한 데모-기반 관계적 정책 학습법은 비단 서비스 로봇을 위한 작업 학습뿐만 아니라 가상 공간상의 지능형 캐릭터 에이전트를 위한 행위 학습 등 다양한 응용 분야에 활용될 수 있을 것이다.

참 고 문 헌

[1] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A Survey of Robot Learning from Demonstration," *Robotics and Autonomous Systems*, vol.57, pp.469-483, 2009.

[2] M. Nicolescu and M. Mataric, "Methods for Robot Task Learning: Demonstrations, Generalization and Practice," *Proc. of the 2nd International Joint Conference on Autonomous Agents and Multi-Agent Systems*, AAMAS'03, pp.241-248, 2003.

[3] E.F. Morales and C. Sammut, "Learning to Fly by Combining Reinforcement Learning with Behavioral Cloning," *Proc. of the 21th International Conference on Machine Learning*, ICML'04, pp.76-81, 2004.

[4] L. D. Raedt, *Logical and Relational Learning*, Cognitive Technologies, Springer, Berlin, 2008.

[5] P. Tadepalli, R. Givan, and K. Driessens, "Relational Reinforcement Learning: An Overview," *Proc. of the 21th International Conference on Machine Learning*, ICML'04, Workshop on Relational Reinforcement Learning, 2004.

[6] J. Saunders, C. L. Nehaniv, and K. Dautenhahn, "Teaching Robots by Moulding Behavior and Scaffolding the Environment," *Proc. of the 1st International Conference on Human-Robot Interaction*, HRI'06, pp.118-125, 2006.

[7] R. Garcia-Duran, F. Fernandez, and D. Borrajo, "Nearest Prototype Classification for Relational Learning," *Proc. of the 16th International Conference on Inductive Logic Programming(ILP-2006)*, pp.89-91, 2006.

[8] K. Johns and T. Taylor, *Professional Microsoft Robotics Developer Studio*, Wiley, 2008.

[9] S. Chernova and M. Veloso, "Confidence-Based Policy Learning from Demonstration Using Gaussian Mixture Models," *Proc. of the 6th International Joint Conference on Autonomous Agents and Multi-Agent Systems*, AAMAS'07, pp.1315-1322, 2007.

[10] H. Veeraraghavan and M. Veloso, "Teaching Sequential Tasks with Repetition through Voice

and Vision," *Proc. of the 7th International Joint Conference on Autonomous Agents and Multi-Agent Systems*, AAMAS'08, pp.518-527, 2008.

[11] E. Winner and M. Veloso, "LoopDISTILL: Learning Domain-Specific Planners from Example Plans," *Proc. of the International Conference on Automated Planning and Scheduling*, ICAPS'07 Workshop on AI Planning and Learning, 2007.

박 찬 영

2009년 경기대학교 정보과학부 전자계산학전공(학사). 2009년~현재 경기대학교 일반대학원 컴퓨터과학과 석사과정. 관심분야는 기계학습, 지능로봇, 자동계획, 에이전트시스템

김 현 식

2001년 경기대학교 전자계산학과(학사) 2004년 경기대학교 일반대학원 전자계산학과(이학석사). 2005년~현재 경기대학교 일반대학원 전자계산학과 박사과정 관심분야는 자동계획, 기계학습, 지능로봇, 지능형 에이전트

김 인 철

1985년 서울대학교 수학과(학사). 1987년 서울대학교 전산과학과(이학석사). 1995년 서울대학교 전산과학과(이학박사). 1996년~현재 경기대학교 자연과학대학 컴퓨터과학과 교수. 관심분야는 자동계획, 기계학습, 지능로봇, 지능형 에이전트