https://doi.org/10.7840/kics.2023.48.2.282

강화학습 모델을 활용한 디지털 롤모델 트윈 연구

김 지 완◆, 제 갈 홍*, 이 승 진*, 이 현 석°

Digital Role-Model Twin Using Reinforcement Learning Model

Ji-Wan Kim◆, Hong Je-Gal*, Seung-Jin Lee*, Hyun-Suk Lee°

요 약

대중들이 모사하고자 하는 인물을 '롤모델'이라고 할 때, 우리는 롤모델에 대한 가치판단 정보를 예측하여 제공할 수 있는 시스템을 디지털 롤모델 트윈으로 정의한다. 본 논문에서는 강화학습 기법을 사용하여 디지털 롤모델 트윈을 생성하고, 생성된 롤모델 트윈을 사용자의 가치판단에 활용할 수 있는 시스템을 제안한다. 구체적으로 디지털 롤모델 트윈 생성과정에서는 현재 상황에 대한 맥락 정보와 의사결정 문제를 롤모델에게 제공하고, 각 의사결정 문제에 대한 롤모델의 선택과 행동을 기반으로 롤모델에 관한 정보와 선호 네트워크를 학습하여 디지털 롤모델 트윈을 생성한다. 사용자는 생성된 디지털 롤모델 트윈을 사용하여, 자신의 상황에 대한 정보와 롤모델에 관한 정보를 바탕으로 주어진 의사결정 문제들에 대한 롤모델의 선호도 정보를 얻을 수 있다. 실험을 통해 디지털 롤모델 트윈이 롤모델의 선호도를 적절히 학습하여 롤모델의 선호도 정보를 효과적으로 모사하는 것을 보인다.

키워드 : 강화학습, 딥-Q 러닝, 추천시스템, 디지털 트윈, 롤모델
Key Words : Reinforcement learning, Deep Q-Learning, Recommendation system, Digital twin, Role model

ABSTRACT

In this paper, we define a digital role model twin as a system that can predict and provide value judgment of a role model, where the 'role model' implies a person the public wants to imitate. To construct digital role model twins, we propose a system that can learn the value judgment of role models using reinforcement learning and enables users to utilize the learned value judgment of role models. Specifically, in the procedure of learning a digital role model twin, decision-making problems are provided to a target role model and the target role model chooses its behavior to the problems. Then, based on contextual information and the role model's behaviors, the role model-specific feature information and preference networks are learned to become a digital role model twin of the target role model. By using the digital role model twin, the users can obtain information about the role model's preference for behaviors to given decision-making problems and contextual information. Through experiments, we demonstrate that the digital role model twin learns the role model's preference appropriately to effectively simulate the role model's preference information.

※ 이 논문은 과학기술정보통신부 및 정보통신기획평가원의 대학ICT연구센터육성지원사업의 연구결과로 수행되었음(IITP-2022-2021-0-01816)
◆ First Author : Sejong University, Department of Smart Device Engineering, jiwan1228@naver.com, 학생회원
° Corresponding Author : Sejong University, School of Intelligent Mechatronics Engineering, hyunsuk@sejong.ac.kr, 정회원
* Sejong University, Department of Unmanned Vehicle Engineering, jagrhong@naver.com; leesj6768@gmail.com
논문번호 : 202210-236-C-RE, Received September 28, 2022; Revised December 3, 2022; Accepted December 6, 2022

Ⅰ. 서 론

과거로부터, 사람들은 닮고자 하는 인물의 행동을 분석하고 행동 그대로를 따라 하는 등 자신의 삶을 발전시키기 위해, 타인의 삶의 방식을 배우고 모사하는 경향이 있다. 자신이 모사하고자 하는 인물을 '롤모델'이라고 칭할 때, 현실 세계에서 롤모델과 실제로 만나 그들의 삶에 대해 배우기에는 환경, 시간적 제약이 따른다. 이와 같은 제약으로 모사하고자 하는 롤모델에 대한 정보를 얻기 위해서는 실제로 롤모델을 만나는 것보다 높은 접근성을 가지는 방법이 필요하다.

한편 디지털 트윈이란 현실 세계에서 실존하는 것을 디지털 공간에 실물과 똑같은 '쌍둥이'를 만들고, 이를 이용한 모의실험 등을 통해 실제 생산 이전에 발생할 수 있는 문제를 찾아내는 등의 기술을 일컫는 말이다. 현재 디지털 트윈은 인공지능과 사물 인터넷 같은 기술이 발전함에 따라 이들을 응용하여 산업 현장에서 안정성과 생산성을 향상하기 위한 기술 트렌드로 주목받고 있다. 구체적으로 디지털 트윈은 스마트 시티 등 산업 전반에 걸쳐 활용되고 있으며 항만, 교통 등 다양한 분야에서도 사용되고 있는 기술이다[1,2]. 특히 디지털 트윈은 시각화가 중요한 제조 산업 분야로부터 발전되고 있어 가상 모델을 시각화하는 기능이 강조되어 발전되고 있다. 이에 사람에 대한 디지털 트윈 분야에서는 대상 그대로 복원하는 기술에 치중한 신체적인 트윈에 집중하여 연구가 진행되는 경향이 있다. 이 같은 연구는 특히 의료 분야를 중심으로 사람의 신체적 정보를 통해 정확한 진단, 처방, 수술을 위한 시뮬레이션으로 주로 활용된다[3,4]. 하지만 실제 사람의 가치판단과 같은 선호도 및 생각, 예측, 판단을 고려하는 모델 구현 측면에서는 여전히 관련 연구가 부족한 실정이다. 따라서 사람에 대한 디지털 트윈 분야에서 사람의 가치판단을 구현할 수 있는 기술에 관한 심층적인 연구가 필요하다.

일반적으로 디지털 트윈을 구현하기 위해, 디지털 트윈 생성 기술에서는 모방할 대상의 데이터를 사용해 디지털 공간에서 실제 원본을 시뮬레이션하는 모델을 개발한다. 이는 모방할 객체의 데이터와 행위를 디지털 공간에 투영하여 디지털 모델을 만들고, 시뮬레이션을 통해 모방 대상의 상태나 행위, 체계를 예측 혹은 최적화하고자 한다는 점을 통해 시뮬레이션 모델에 대응된다[5,6]. 이때 디지털 트윈 구현에 필요한 디지털 모델은 일반적으로 인공지능 기반의 데이터 모델링을 통해 만들어진다. 최근에는 모방할 객체로부터 수집되는 데이터들을 기계학습/인공지능 알고리즘으로 학습하여 다 계층 인공신경망 형태로 모델링하고 분석 및 예측, 최적화에 활용하는 추세가 증가하고 있다. 이는 인공지능이나 기계학습을 통해 매우 많은 수치계산을 고속으로 처리하여 실시간 시뮬레이션을 가능하게 한다.

본 논문에서는 앞서 말한 디지털 트윈 및 데이터 모델링 기술을 통해 사람들이 모사하고자 하는 롤모델을 가상세계에 투영하여 롤모델의 가치판단을 구현할 수 있는 디지털 롤모델 트윈을 생성하고자 한다. 디지털 롤모델 트윈을 생성하게 되면 가상세계의 특성으로 인해 롤모델을 직접 만나 정보를 얻는 과정의 환경, 시간 제약이 사라지게 된다. 이에 따라 롤모델의 접근성이 크게 향상될 것을 기대할 수 있다. 그리고 롤모델의 시각적 이미지를 그대로 복원하는 것이 아닌, 롤모델의 선호도와 같은 사람의 가치판단을 구현할 수 있다.

구체적으로 본 논문에서는 상황에 대한 정보와 롤모델에 대한 정보를 사용하여 강화학습 기법을 통해 롤모델의 선호도와 같은 가치판단을 예측하는 디지털 롤모델 트윈을 생성하는 시스템을 제안한다[7,8]. 그리고 디지털 롤모델 트윈을 이용해 물리적인 실제 롤모델과의 직접적인 상호작용 없이도 롤모델의 선호도 정보를 예측하여 사용자가 자신의 가치판단에 활용할 수 있는 시스템을 제안한다.

Ⅱ. 디지털 롤모델 트윈

디지털 트윈은 실제 세계에 존재하는 객체를 디지털 공간에 모사하여 대상 객체의 특징을 실제 세계로부터 디지털 세계로 옮기는 것이다. 그리고 객체의 특징을 계속해서 동화시키고, 시뮬레이션을 통해 메커니즘을 동적으로 모사할 수 있어야 한다[9].

본 논문에서는 디지털 트윈 기술에서 대상 객체를 롤모델로 지정하여 디지털 롤모델 트윈을 구현하고자 한다. 일반적으로 사람의 특성 및 특징은 그 사람의 선호도에 따른 과거 선택과 행동을 통해 유추할 수 있다. 따라서 롤모델의 특징은 롤모델의 선호도에 따른 선택과 행동을 반영하여 형성된다. 이렇게 형성된 롤모델의 특징을 디지털 공간에 투영하여 디지털 롤모델 트윈을 구현하면, 롤모델이 선호하는 선택과 행동을 예측하는데 사용할 수 있다. 이처럼 롤모델의 선호도 정보를 예측하여 제공할 수 있는 디지털 트윈을 디지털 롤모델 트윈이라 정의한다.

디지털 롤모델 트윈을 활용하면 실제 롤모델로부터 정보를 얻을 때 존재했던 공간적, 시간적 제약이 사라

지게 된다. 그리고 횟수의 제한 없이 사용할 수 있으며, 다수의 롤모델로부터의 디지털 롤모델 트윈을 생성하게 된다면 원하는 상황에 맞추어 서로 다른 롤모델로부터 선호도 정보를 얻을 수도 있다. 이 같은 디지털 롤모델 트윈의 특징은 사람들이 현실에서 마주하는 서로 다른 상황에 맞게 롤모델의 선호도 정보를 손쉽게 생성하고 제공함으로써 사람들의 의사결정에 활용할 수 있게 한다. 예를 들어, 인터넷 뉴스를 읽으려고 할 때 디지털 롤모델 트윈을 활용하면 사용자는 수많은 뉴스 중 자신의 롤모델이 현재 상황에서 가장 선호하는 뉴스를 알 수 있고, 이를 실제 자신의 뉴스 선택에 활용할 수 있다.

Ⅲ. 강화학습 기반 디지털 롤모델 트윈

디지털 롤모델 트윈은 상황에 대한 정보와 과거에 롤모델이 선택했던 정보 및 롤모델에 대한 정보들을 기반으로 롤모델의 선호도를 학습하여 주어진 선택지에 대한 롤모델의 선호도를 예측하는 기능을 가진다. 이 같은 선호도 예측은 사용자의 순차적인 선택을 기반으로 사용자가 선호하는 항목을 추천해주는 강화학습 기반의 추천시스템을 활용하여 달성될 수 있다[10-13]. 따라서 본 논문에서는 딥 러닝 모델을 이용한 강화학습 기반의 추천시스템 구조[14,15]를 바탕으로 디지털 롤모델 트윈 구조를 제안하였다.

제안하는 디지털 롤모델 트윈은 role-model twin agent, Decision Feature(DF), Context Feature(CF), Role-model Feature(RF), 선호 네트워크로 구성된다. Role-model twin agent는 디지털 롤모델 트윈 생성 시 학습하는 주체이자 엔진이다. Decision Feature(DF)는 선택지의 핵심키워드와 같이 롤모델의 선택 혹은 행동을 표현하는 의사결정 특징정보이고 Context Feature (CF)는 롤모델에 관한 정보 외에 의사결정에 영향을 미치는 날씨와 시간과 같은 맥락 정보이다. 그리고 Role-model Feature(RF)는 과거 롤모델이 선택했던 의사결정에 대한 특징정보와 롤모델의 나이, 성별과 같은 롤모델에 관한 정보로 구성된다. 마지막으로 선호 네트워크(Preference network)는 DF, RF, CF를 바탕으로 롤모델이 어떤 행동이나 선택을 선호하게 될지 학습하여 선택지에 대한 롤모델의 선호도를 예측하는 네트워크이다.

디지털 롤모델 트윈은 맥락 정보와 주어진 의사결정 문제들을 롤모델에게 제공해주고, 각 의사결정 문제에 대한 롤모델의 선택과 행동을 기반으로 롤모델에 관한 정보와 선호 네트워크를 학습하는 과정을 반복하여 생성할 수 있다[16]. 구체적으로 맥락 정보 CF와 의사결정 특징정보 DF로 이루어진 의사결정 문제들을 롤모델에게 제공한다. 그리고 롤모델은 자신의 선호도를 기반으로 선택과 행동을 통해 선호 네트워크와 RF를 학습하는 과정을 반복한다. 디지털 롤모델 트윈이 생성되었다면, 사용자는 디지털 롤모델 트윈을 이용하여 주어진 의사결정 문제들에 대한 롤모델의 선호도 정보를 얻을 수 있다. 사용자는 자신의 맥락 정보 CF와 롤모델에 관한 정보 RF를 디지털 롤모델 트윈에 넣어 각 의사결정 문제들에 대한 롤모델의 선호도 정보를 확인하고 자신의 가치판단에 활용할 수 있다. 본 논문에서는 디지털 롤모델 트윈을 생성하는 과정과 활용하는 과정으로 나누어 설명한다.

3.1 강화학습 기반 디지털 롤모델 트윈 생성과정

디지털 롤모델 트윈 생성은 특정 롤모델이 여러 번에 걸쳐 주어진 여러 선택지 중 선호하는 선택지를 고르는 과정을 반복하고 그 롤모델의 선택에 기반을 둔 롤모델의 RF 및 선호 네트워크를 학습하는 과정을 통해 구현한다. 구체적으로, 총 네 단계 과정인 '정보 불러오기', '롤모델에게 제공할 수 있는 선택지 리스트 생성', '선택지 리스트 제공 및 피드백 전달받기', '선호 네트워크 학습'을 반복하여 이루어진다.

각 과정을 설명하기 전, 시스템의 구성요소와 표기에 대해 먼저 정리하여 설명한다. 디지털 롤모델 트윈 생성과정에서 매 선택이 이루어지는 단위를 time slot $t \in \{1,2...\}$로 나타낸다. 디지털 롤모델 트윈의 구성요소 중 의사결정 특징정보로 이루어진 롤모델에게 제공할 수 있는 모든 선택지를 D 라고 가정할 때, 한 time slot t에서 롤모델에게 제공할 수 있는 선택지들의 집합을 $D_t \subset D$로 표기한다. D_t는 총 N_l의 선택지 개수를 가지며 각 선택지는 $d_t^n \in D_t = \left\{d_t^1, d_t^2, ...d_t^{N_l}\right\}$로 나타낸다. 학습 과정에서 D_t의 존재하는 각 선택지를 뽑아 생성한 롤모델에게 제공할 선택지 리스트는 L_t로 표기한다. 선택지 리스트 L_t는 총 N_f의 선택지 개수를 가지며 각 선택지는 $l_t^k \in L_t = \left\{l_t^1, l_t^2, ...l_t^{N_f}\right\}$로 나타낸다. 그리고 디지털 롤모델 트윈의 구성요소 중 time slot t에서 특정 롤모델에 대한 정보 및 과거 롤모델이 선택했던 의사결정 특징정보를 h_t로 나타낸다. 마지막으로 time slot t에서 롤모델의 대한 정보 외에 의사결정에 영향을 미치는 맥락 정보는 c_t로 나타낸다.

선호도 정보를 저장하고자 하는 롤모델들을 대상으

로 디지털 롤모델 트윈 생성과정을 진행하면 각 롤모델들의 디지털 롤모델 트윈을 생성할 수 있다. 디지털 롤모델 트윈 생성과정을 구체적으로 설명하기 위해, 디지털 롤모델 트윈의 생성 대상을 롤모델 m이라고 가정하고 롤모델 m에 대한 총 네 단계의 디지털 롤모델 트윈 생성과정을 설명한다.

첫 번째 '정보 불러오기' 단계는 롤모델 m에게 제공할 선택지 리스트 L_t을 생성하기 위해, 필요한 정보들을 불러오는 단계이다. 선택지 리스트 L_t을 생성하기 위해서는 time slot t에서 롤모델 m에게 제공해 줄 수 있는 선택지들의 집합인 D_t, 롤모델 m에 대한 특징정보인 h_t, 그리고 맥락 정보 c_t가 필요하므로 각각 불러온다.

두 번째 '롤모델에게 제공할 수 있는 선택지 리스트 생성' 단계는 불러온 정보들인 D_t, h_t, c_t와 선호 네트워크를 이용하여 D_t의 존재하는 각 선택지에 대한 롤모델 m의 선호도를 확인하고 확인한 선호도를 바탕으로 롤모델 m이 선호할 선택지들과 무작위 선택지들을 확률적으로 뽑아 롤모델 m에게 제공할 선택지 리스트 L_t을 생성하는 단계이다. 디지털 롤모델 트윈의 선호 네트워크를 구성하는 입력 feature들은 롤모델에 관한 정보 $h_{role}=(h^1_{role},h^1_{role},...h^{N_{role}}_{role})$와 롤모델이 과거 선택했었던 의사결정 특징정보 $h_p=(h_{p,1},h_{p,2},...h_{p,N_d})$로 이루어진 한 time slot t에서 $h=(h_{role},h^1_p,h^2_p,...h^{N_{pt}}_p)\in H$는 롤모델에 대한 특징정보를 의미하고, 한 time slot t에서 맥락 정보는 $c=(c_1,c_2,...c_{N_c})\in C$, 그리고 하나의 선택지 $a=(a_1,a_2,...a_{N_d})\in A$ 이고 출력 값은 주어진 하나의 선택지에 대한 롤모델의 선호도인 $y\in Y$로 정의한다[17]. 여기서 N_{role}은 롤모델에 대한 정보의 feature 개수, N_d는 의사결정 특징정보 feature 개수, N_{pt}는 롤모델이 과거 의사결정 했었던 시점들의 feature 개수이며 N_c는 맥락 정보 feature 개수를 의미한다. H, C, A는 각각 입력 공간을 의미하고 Y는 출력 공간을 의미한다. 따라서 각 입력 공간은 $H=[0,1]^{N_{role}+N_dN_{pt}}$, $C=[0,1]^{N_c}$, $A=[0,1]^{N_d}$ 와 같음을 가정할 수 있다. 따라서 우리는 선호 네트워크의 입력 feature들과 주어진 선택지에 대한 롤모델의 선호도 관계를 $P: H\times C\times A\rightarrow Y$로 정의한다. 정리하면 롤모델 특징정보 h와 맥락 정보 c는 강화학습 과정에서의 상태(state)에 대응되며, 선택지 a는 행동(action)에 대응되고, 롤모델의 선호도를 예측할 선호도 네트워크에 입력으로 사용된다.

Time slot t에서 선택지들에 대한 롤모델의 선호도를 예측하기 위해, 미리 불러온 정보들인 h_t, c_t, D_t와 위에서 정의된 선호 네트워크를 이용한다. 선호 네트워크의 입력 feature는 h_t, c_t, 그리고 선택지들의 집합 D_t의 존재하는 각 선택지 d^n_t이고 출력 값은 $y^n_t\in y_t=\left\{y^1_t,y^1_t,...y^{N_t}_t\right\}$로 각 선택지 d^n_t에 대한 롤모델 m의 선호도를 의미한다. 따라서 time slot t에서 선호 네트워크 $P: H\times C\times D\rightarrow Y$의 관계는 함수 $y^n_t=P(h_tc_td^n_t)$로 정의한다. 학습의 주체인 Role-model twin agent는 h_t, c_t, d^n_t을 선호 네트워크에 input으로 넣어 각 선택지 d^n_t에 대한 롤모델 m의 선호도 y^n_t 을 얻는다. 모든 선택지에 대한 롤모델 m의 선호도 $y^1_t,y^2_t,y^3_t...y^{N_t}_t$ 확인이 끝난 후, 선택지 리스트 L_t을 생성한다. 선택지 리스트 L_t을 생성할 때 학습 편향을 줄이고 학습의 exploration을 보장하기 위해, ϵ-greedy 방법을 활용하여 D_t에서 총 N_f개의 선택지를 뽑는다[7]. ϵ-greedy 방법을 활용하는 이유는 디지털 롤모델 트윈은 롤모델이 선호하는 선택지를 제공해주면서 다양한 선호도 정보를 담아야 하기 때문이다. 만약 선호도가 높은 선택지 위주로만 제공해주게 된다면 다양한 선택지에 대한 선호도 정보를 담을 수 없고 특정 선택지들로만 편향되어 제공될 수 있다. 그리고 무작위로만 선택지를 제공해주게 된다면 롤모델 m의 선호도를 고려하지 않으므로 롤모델 m이 학습에 참여하게 되는 동기가 없어지는 문제가 생길 수 있다. ϵ-greedy 방법을 활용하는 것은 구체적으로 ϵ확률로는 선택지 집합 D_t 에서 임의의 선택지를 뽑고 (1-ϵ) 확률로는 D_t에서 선호도가 가장 높은 선택지를 뽑는 방법을 의미한다. 이때 선택지 리스트 L_t에 중복된 선택지가 들어가는 것을 방지하기 위해 선택지를 하나씩 뽑되 이전에 뽑히지 않은 선택지 중 하나를 뽑는다. 중복을 방지한 ϵ-greedy 방법을 활용하여 하나씩 선택지를 뽑는 과정을 총 N_f번 반복하여 총 N_f개의 선택지를 가진 선택지 리스트 L_t을 생성한다.

세 번째 '선택지 리스트 제공 및 피드백 전달받기' 단계는 생성된 선택지 리스트 L_t의 각 선택지를 롤모델 m에게 순서대로 전달해주고, 롤모델 m은 이를 바탕으로 각각 의사 결정하며 이 결과에 따라 피드백이 결정되어 불러왔던 정보들과 선택지, 선택지에 대

한 피드백을 모두 replay memory B에 저장하는 단계이다. 먼저 두 번째 단계에서 생성한 선택지 리스트 L_t의 각 선택지 l_t^k를 롤모델 m에게 순서대로 제공한다. 롤모델 m은 받은 각 선택지 l_t^k에 대해 자신의 선호도에 따라 의사결정을 진행하고 의사결정이 완료되면 의사결정 결과가 l_t^k에 대한 피드백 정보가 된다. 피드백 정보를 바탕으로 롤모델 m이 선택지 l_t^k를 선택했으면 1, 선택하지 않았으면 0으로 decision reward가 결정된다. 따라서 decision reward는 $r_t^k \in \{0,1\}$로 정의한다. 선택지 l_t^k에 대한 롤모델 m의 피드백 전달 및 decision reward가 결정된 후, role-model twin agent는 l_t^k와 r_t^k를 비롯해 선호 네트워크 학습에 사용하게 될 필요한 정보들이 담긴 experience tuple을 replay memory B에 저장해야 한다. 이때 다음 time slot $t+1$에서의 선호도 예측이 필요하고, 이를 위해서는 c_{t+1} , D_{t+1}과 t에서의 선택지 l_t^k에 따라 변화된 다음 time slot $t+1$에서의 h_{t+1}이 필요하다. 따라서 role-model twin agent는 학습에 필요한 h_t, c_t, l_t^k, h_{t+1}, c_{t+1}, D_{t+1} ,r_t^k를 모두 얻어 (h_t, c_t, l_t^k, h_{t+1}, c_{t+1}, D_{t+1} ,r_t^k) experience tuple을 replay memory B에 저장한다. 세 번째 단계를 선택지 리스트 L_t에 존재하는 모든 선택지에 대해 반복 진행하여 모든 선택지에 대한 experience tuple을 저장한다.

네 번째 '선호 네트워크 학습' 단계는 replay memory B에 저장되어있는 정보들을 바탕으로 롤모델 m의 선호도를 담도록 선호 네트워크를 학습하는 단계이다. 일반적으로 데이터를 이용하여 네트워크를 학습할 때 전체 데이터에 대한 target value와 네트워크가 예측한 값의 차이로 손실을 계산하여 다음 네트워크 가중치를 업데이트하는 방식을 사용한다. 그러나 데이터가 많으면 전체 데이터에 대한 손실을 구하는 것이 힘들고 계산량도 많아져 한 번 가중치를 업데이트할 때 오랜 시간이 걸리게 된다. 이를 해결하기 위해 mini-batch 학습 방법을 활용한다. Mini-batch 학습 방법은 전체 데이터에 대해 학습하지 않고 지정한 batch size만큼 나눈 mini-batch sample을 활용하여 학습하는 방법이다. 세 번째 단계가 종료되면, 선택지 리스트 L의 모든 선택지에 대한 h_t, c_t, l_t^k, h_{t+1}, c_{t+1}, D_{t+1} ,r_t^k 정보가 replay memory D에 저장된다. 우리는 mini-batch 방법을 활용해 네트워크를 학습하므로 저장된 experience tuple들 중 batch size N_b로 나누어 훈련에 사용할 mini-batch sample을 만든다.

Mini-batch sample의 선택지는 l, decision reward는 r, 롤모델에 대한 특징정보는 h, 다음 time step에서 변화된 롤모델에 대한 특징정보는 h', 다음 time step에서 변화된 맥락 정보는 c', 그리고 target value를 y^{TARGET}로 일반화하여 표기한다. 이제 생성한 mini-batch sample의 target value y^{TARGET}와 네트워크가 예측한 값 간의 차이를 계산하여 손실을 계산할 수 있다.

Mini-batch sample의 target value y^{TARGET}은 선택지 l의 현재 decision reward인 r과 reward에 기반한 미래의 롤모델 선호도 기댓값을 총합한 현재와 미래를 모두 고려하는 롤모델의 누적 선호도 기댓값으로, 아래 수식 (1)과 같이 표현할 수 있다.

$$y^{TARGET} = r + \gamma P(h', c', \arg\max_{d' \in D'} P(h', c', d'; \theta); \theta') \tag{1}$$

수식 (1)에서 θ는 네트워크 가중치, θ'는 타겟 네트워크 가중치, γ는 discount factor, D'는 다음 time step에서 롤모델 m에게 제공할 수 있는 선택지들의 집합을 의미한다. 수식 (1)의 y^{TARGET}을 선호 네트워크의 학습에 이용하는 것은 의사결정 상황이 동적으로 변화하므로 미래의 선택에 관한 고려가 없다면, 선호 네트워크가 현재 의사결정 결과에 따라 편향되게 학습될 가능성이 있기 때문이다. 그리고 수식 (1)에서는 네트워크 가중치 θ의 안정적인 학습을 위해 다음 time step에서 롤모델 m이 가장 선호할 선택지의 예측은 네트워크 가중치 θ를 이용하여 수행하고, 미래의 롤모델 누적 선호도 기댓값은 네트워크 가중치 θ가 아닌 타겟 네트워크 가중치 θ'을 이용하여 $P(h', c', \arg\max_{d' \in D'} P(h', c', d'; \theta); \theta')$와 같이 계산 한다[18].

선호 네트워크가 예측한 선택지 l에 대한 롤모델 m의 선호도는 h, c, l을 입력 feature로 사용하여 얻을 수 있는 출력 값 $y = P(h,c,l;\theta)$이므로 y^{TARGET}와 선호 네트워크가 예측한 출력 값 y의 차이로 손실을 계산하며 손실을 최소화하는 최적의 값을 찾아 다음 네트워크 가중치를 업데이트해야 한다. 이때 손실을 최소화하는 방법으로 경사 하강법을 활용한다. 경사 하강법을 활용하면 y^{TARGET}와 y의 오차를 제곱하는 식을 손실 함수 (2)로 이용하여 경사 하강을 통해 손실을 최소화하는 최적의 값을 구할 수 있다.

표 1. 제안 알고리즘
Table 1. Proposed Algorithm

Algorithm 1

Procedure of the Digital Role-model Twin generation

Notations: discount factor γ, epsilon ϵ , replay memory B, replay memory maximum size N_r , network parameter θ, target network parameter θ', role-model feature h, context feature c, decision feature D, decision reward r, preference list L, training batch size N_b , target network replacement frequency N'

Initialize θ

Initialize $\theta' \leftarrow \theta$

Initialize empty B

for time-slot $t \in \{1,2,...\}$ **do**

  Obtain h_t, c_t, $D_t \in D$

  for $n \in \{1,2,...N_l\}$ **do** ▷ in parallel

    Set $d_t^n \in D_t = \{d_t^1, d_t^2, ...d_t^{N_l}\}$

    Calculate output $y_t^n \in \{y_t^1, y_t^2, ...y_t^{N_l}\}$ using h_t, c_t, d_t^n

    $y_t^n = P(h_t, c_t, d_t^n)$

  end for

  for $k \in \{1,2,...N_f\}$ **do**

$$d^k = \begin{cases} Randomly\ choose\ d^k\ from\ D_t & with\ probability\ \epsilon \\ Choose\ d^k\ with\ the\ largest\ y_t^k\ from\ D_t & otherwise \end{cases}$$

    Remove d^k from D_t

    $L_t \leftarrow L_t \cup \{d^k\}$

  end for

  Obtain c_{t+1} and $D_{t+1} \in D$ in next time-slot $t+1$

  for $k \in \{1,2,...N_f\}$ **do**

    Send $l_t^k \in L_t = \{l_t^1, l_t^2, ...l_t^{N_f}\}$ to Role-model

    Receive feedback of l_t^k from Role-model $r_t^k \in \{0,1\}$

  end for

  Observe h_{t+1} in next time-slot $t+1$ from the feedback r_t^k of l_t^k's

  for $k \in \{1,2,...N_f\}$ **do**

    Store tuple $(h_t, c_t, l_t^k, h_{t+1}, c_{t+1}, D_{t+1}, r_t^k)$ to B, replacing the oldest tuple if $|B| \geq N_r$

  end for

  Sample a mini-batch of N_b tuples $(h_t, c_t, l_t^k, h_{t+1}, c_{t+1}, D_{t+1}, r_t^k) \sim \mathrm{Unif}(B)$

  Construct target values, one for each of the N_b tuples:

  Define d' that gives maximum future reward is selected according to parameter θ

  $y_j = r_j + \gamma P(h_{j+1}, c_{j+1}, \underset{d' \in D_{j+1}}{\arg\max} P(h_{j+1}, c_{j+1}, d'; \theta); \theta')$

  Do a gradient descent step with loss $\|y_j - P(h_j, c_j, l_j; \theta)\|^2$ w.r.t. the network parameter θ

  Replace target parameters θ' as θ every N' step

end for

$$\|y^{TARGET} - P(h,c,l;\theta)\|^2 \quad (2)$$

따라서 (2)의 손실 함수를 계산하여 손실을 최소화하는 네트워크 가중치인 θ를 업데이트한다. 마지막으로 타겟 네트워크 가중치인 θ' 는 주기적으로 N'시기마다 θ로 바꿔주어 업데이트한다.

롤모델 m이 첫 단계부터 네 번째 단계까지 과정을 반복하여 진행한다면 선호 네트워크가 롤모델 m의 선호도로 충분히 학습되어 롤모델 m의 선호도를 담은 디지털 롤모델 트윈을 생성할 수 있다. 그리고 time slot t에서 전체 네 단계를 다수의 롤모델이 병렬적으로 진행한다면 공통된 선호 네트워크를 공유하는 다수의 디지털 롤모델 트윈을 동시에 생성할 수 있다. 디지털 롤모델 트윈 생성의 전체 과정에 대한 알고리즘은 표 1.에서 확인할 수 있다.

3.2 디지털 롤모델 트윈 시스템 구성 및 활용

하나 이상의 디지털 롤모델 트윈이 생성되었다면, 사용자들은 디지털 롤모델 트윈을 자신의 가치판단에 활용할 수 있다. 디지털 롤모델 트윈 활용은 디지털 롤모델 트윈에 존재하는 특정 롤모델이 과거 선택했었던 의사결정 특징정보 및 롤모델에 관한 특징정보 RF, 충분한 학습을 통해 특정 롤모델의 선호도를 담은 선호 네트워크와 특정 롤모델에 대한 정보 외에 의사결정에 영향을 미치는 특징정보 CF, 의사결정 특징정보로 이루어진 특정 롤모델의 선호도 정보를 얻고자 하는 의사결정 문제 DF'들로 이루어진 집합 DF로 구성된다.

디지털 롤모델 트윈 활용은 사용자들이 자신의 컨텍스트 정보와 특정 롤모델의 선호도 정보를 얻고자 하는 의사결정 문제들을 롤모델들의 각 디지털 롤모델 트윈을 이용하여 의사결정 문제들에 대한 각 롤모델의 선호도를 확인하고 이를 자신의 가치판단에 활용하는 과정으로 구현한다. 디지털 롤모델 트윈 활용 과정을 구체적으로 확인하기 위해, 선호도 정보를 얻고자 하는 대상을 한 명의 롤모델 m이라고 가정하고 롤모델 m의 디지털 롤모델 트윈 생성과정을 거쳐 롤모델 m의 선호도를 충분히 학습한 디지털 롤모델 트윈을 생성하였다고 가정한다. 그리고 롤모델 m에 대한 선호도 정보를 얻어 자신의 가치판단에 활용하고자 하는 사람을 한 명의 사용자 u라고 가정한다. 롤모델 m의 디지털 롤모델 트윈에 존재하는 롤모델 m에 대한 특징정보를 $\hat{h}$, 사용자 u가 선호도 정보를 얻고자 하는 의사결정 문제 집합을 $\hat{D}$, 그리고 사용자 u의

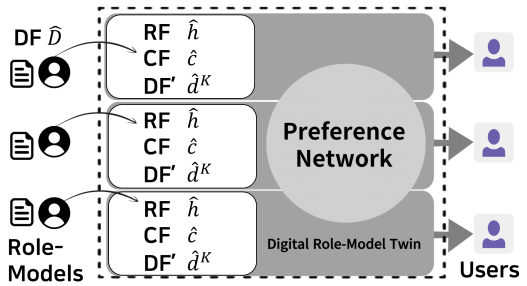

그림 1. '디지털 롤모델 트윈' 시스템 활용 구조도
Fig. 1. Structure of utilizing 'Digital Role-model Twin' system

맥락 정보를 $\hat{c}$로 표기한다. 디지털 롤모델 트윈 시스템 구성 및 활용 과정에 대한 구조도는 그림 1에서 확인할 수 있다.

디지털 롤모델 트윈 활용 과정은 총 세 단계 '정보 불러오기', '롤모델의 선호도 확인하기', '의사결정 상황에 활용하기'로 이루어진다.

첫 단계 '정보 불러오기' 단계는 사용자 u가 롤모델 m의 선호도 정보를 얻는 데 필요한 특징정보들을 불러오는 단계이다. 롤모델 m의 선호도 정보를 얻기 위해서는 롤모델 m의 디지털 롤모델 트윈에 저장되어있는 롤모델에 대한 특징정보인 $\hat{h}$를 비롯해 사용자 u의 맥락 정보 $\hat{c}$, 그리고 의사결정 문제 집합 $\hat{D}$가 필요하다. 이때 의사결정 문제 집합 $\hat{D}$에 존재하는 의사결정 문제는 총 $N_{\hat{f}}$개라고 정의하고, 의사결정 문제 집합 $\hat{D}$에 존재하는 각 의사결정 문제는 $\hat{d}^K \in \hat{D} = \{\hat{d}^1, \hat{d}^2, ... \hat{d}^{N_{\hat{f}}}\}$로 정의한다. 두 번째 '롤모델의 선호도 확인하기' 단계는 $\hat{h}$, $\hat{c}$, $\hat{D}$를 디지털 롤모델 트윈을 활용해 각 의사결정 문제 $\hat{d}^K$에 대한 롤모델의 선호도를 확인하는 단계이다. 롤모델 m의 디지털 롤모델 트윈에는 롤모델 m의 선호도 정보를 담은 선호 네트워크가 존재한다. 선호 네트워크의 입력 feature는 불러온 $\hat{h}$, $\hat{c}$와 하나의 의사 결정 문제 $\hat{d}^K$이고 출력 값은 $\hat{d}^K$에 대한 롤모델 m의 선호도인 $\hat{y}^K \in \hat{y} = \{\hat{y}^1, \hat{y}^2, ... \hat{y}^{N_{\hat{f}}}\}$로 정의한다. 따라서 롤모델 m의 선호 네트워크는 함수 $\hat{y}^K = P(\hat{h}, \hat{c}, \hat{d}^K)$로 정의한다. 사용자 u는 불러온 $\hat{h}$, $\hat{c}$, $\hat{D}$를 롤모델 m의 선호 네트워크를 활용하여 $\hat{D}$의 존재하는 각 선택지 $\hat{d}^K$에 대한 롤모델 m의 선호도 $\hat{y}^K$를 확인한다. $\hat{D}$에 존재하는 의사결정 문제는 총 $N_{\hat{f}}$개이므로 총 $N_{\hat{f}}$번 반복하여 모든 의사결정 문제에 대한 롤모델 m의 선호도 $\hat{y}^1, \hat{y}^2, ... \hat{y}^{N_{\hat{f}}}$를 모두 확인한다.

세 번째 '의사결정 상황에 활용하기' 단계는 두 번째 단계에서 얻은 롤모델 m의 선호도 정보를 사용자 u가 활용하는 단계이다. 사용자 u는 롤모델 m의 디지털 롤모델 트윈을 활용하여 의사결정 문제들에 대한 롤모델 m의 선호도 정보를 모두 확인하고 자신의 의사결정 상황에 활용할 수 있다. 이때, 선호도는 단순히 예측된 결괏값으로 제공될 수 있을 뿐만 아니라 예측된 결과를 활용하여 다양한 형태로 사용자에게 제공될 수 있다. 예를 들어, 선호도를 softmax 함수를 사용하여 각각의 의사결정 문제를 선택할 확률로 변환하여 사용자에게 제공할 수 있다.

디지털 롤모델 트윈의 활용은 사용자가 선호도 정보를 얻고자 하는 롤모델이 다수일 때도 적용할 수 있다. 사용자가 다수의 롤모델의 선호도 정보를 얻고자 한다면 두 번째 단계에서 $\hat{c}$, $\hat{D}$, 각 롤모델의 $\hat{h}$를 각 디지털 롤모델 트윈의 선호 네트워크 입력 feature로 사용하면 각 롤모델의 선호도 정보를 얻을 수 있다. 그리고 다수의 사용자가 다수의 롤모델의 선호도 정보를 얻고자 할 때 역시 사용자별 $\hat{c}$와 $\hat{D}$, 각 롤모델의 $\hat{h}$를 각 디지털 롤모델 트윈의 선호 네트워크 입력 feature로 사용하면 각 사용자는 의사결정 문제들에 대한 선호도 정보를 얻고자 하는 롤모델들의 선호도 정보를 얻을 수 있다. 더 나아가, 만약 롤모델의 선호도가 변화하더라도 변화한 선호도를 디지털 롤모델 트윈에 적용할 수 있다. 디지털 롤모델 트윈을 구현한 강화학습은 온라인 학습 특성을 가지므로 디지털 롤모델 트윈을 구성하는 $\hat{h}$와 선호 네트워크를 롤모델의 변화한 선호도에 따른 선택에 따라서 계속 갱신할 수 있다.

Ⅳ. 모의실험

이 섹션에서는 뉴스추천 시나리오를 적용한 실험을 통해 제안된 강화학습 기반 디지털 롤모델 트윈이 롤모델의 선호도를 성공적으로 반영하는지 확인한다. 뉴스추천 시나리오는 여러 롤모델들에게 선호 네트워크를 통해 예측된 각 롤모델에게 추천하는 세 개의 뉴스들을 제공해주고 추천받은 뉴스들을 각 롤모델이 자신의 관심도와 선호도에 따라 뉴스를 선택하여 읽는 과정을 반복하며 진행한다. 선호 네트워크는 각 롤모델이 과거에 선택했었던 뉴스들의 카테고리 정보인 롤모델 특징정보와 날씨와 같은 맥락 정보를 바탕으

로 각 롤모델의 뉴스 카테고리에 대한 선호도 정보를 가지도록 학습한다. 선호 네트워크가 충분히 학습되면, 각 롤모델의 뉴스 카테고리에 대한 선호도 정보를 가진 공통된 선호 네트워크를 공유하는 디지털 롤모델 트윈들을 동시에 생성한다.

본 실험에서는 뉴스추천 시나리오를 모의할 수 있는 Python 시뮬레이터를 구현하여 진행하였다. 실제 뉴스들의 카테고리를 분류하여 '경제, 사회, 연예, 스포츠, 날씨'를 만들고 롤모델에게 추천해줄 때 각 카테고리 뉴스들이 하나씩 존재한다고 가정한다. 디지털 롤모델 트윈 생성과정에서 롤모델에게 뉴스를 제공하고 롤모델이 뉴스를 선택하는 횟수는 충분한 학습을 위해 총 5000번으로 가정한다. 이때, 실제 사람의 선호도는 정량화하여 평가하기 어려우므로 본 실험에서는 각자 다른 특정 뉴스 카테고리들을 선호하는 가상의 롤모델들을 만들어서 활용한다. 각 가상의 롤모델은 다양한 카테고리를 갖는 실제 뉴스들을 추천받았을 때, 미리 설정된 선호도에 따라 선호도가 높은 뉴스 카테고리의 뉴스를 높은 확률로 선택하여 읽도록 한다. 본 실험에서는 총 두 명의 가상 롤모델들을 고려하며 각각의 롤모델은 총 5개의 뉴스 카테고리 중 각자 두 가지 뉴스 카테고리를 선호한다고 가정한다. 각 가상의 롤모델은 특정 뉴스가 주어질 때, 해당하는 뉴스가 선호하는 카테고리의 뉴스라면 80%의 확률로 선택하여 읽고, 그 외의 뉴스 카테고리들은 20%의 확률로 선택하여 읽는다고 가정한다. 이때 두 명의 가상 롤모델들을 A, B로 표현하고 가상 롤모델 A는 뉴스 카테고리 경제와 사회를, 가상 롤모델 B는 연예와 스포츠를 선호한다고 가정한다. 그리고 본 실험에서 맥락 정보로 날씨를 고려한 '맑음, 흐림, 바람, 눈/비'로 구성된 네 가지의 상황을 설정하고, 해당 정보가 뉴스를 추천할 때마다 변화한다고 가정한다. 만약 맥락 정보가 눈/비가 오는 날씨일 경우에는 가상 롤모델들이 날씨에 관한 뉴스 카테고리를 가장 선호하여 선택해 읽는다고 가정한다.

본 실험에서 가상 롤모델이 뉴스를 선택하여 읽으면 decision reward는 1로 결정되고, 읽지 않으면 0으로 결정된다. 또한, 특수한 경우의 선호도에 가중치를 두는 경우를 확인하기 위하여, 눈/비가 오는 날씨일 때 뉴스 카테고리 5번을 선택하여 읽는다면, decision reward는 기존의 decision reward의 두 배로 결정한다. 롤모델에 대한 특징정보는 최근 10일 동안 가상 롤모델이 선택했었던 뉴스 카테고리들로 이루어진다. 실험은 DDQN 구조에 따르며, 64의 노드를 가지는 세 개의 hidden layer들로 구성된 fully-connected neural network를 사용한다. 그리고 활성화 함수는 ReLU를 사용하였고, 학습률은 10^{-3}, 선호 네트워크의 타겟 네트워크 가중치 업데이트 간격은 20, 선호 네트워크 학습에 사용되는 batch size는 32로 설정하였다. 이때 학습에 사용할 batch sample은 가상 롤모델에 대한 특징정보, 맥락 정보, 가상 롤모델이 추천받은 뉴스 카테고리, 추천받은 뉴스 카테고리에 대한 가상 롤모델의 선택에 따른 decision reward, 다음 time step에서 변화한 가상 롤모델에 대한 특징정보, 다음 time step의 맥락 정보, 다음 time step에서 가상 롤모델에게 가장 추천하는 뉴스 카테고리로 이루어진다.

가상 롤모델들의 선택을 바탕으로 선호 네트워크를 학습하여 각 디지털 롤모델 트윈을 생성한 후, 각 디지털 롤모델 트윈이 롤모델의 선호도를 성공적으로 반영하는지 확인하기 위해 생성된 각 디지털 롤모델 트윈에 해당하는 롤모델에 관한 특징정보와 새로운 맥락 정보들을 바탕으로 뉴스 카테고리들에 대한 롤모델의 선호도를 예측하여 가장 높은 선호도를 갖는 뉴스 카테고리 두 가지를 선택한다. 이 같은 과정을 총 1000번 반복하여 결과를 평균 내어 각 카테고리가 선택받은 비율을 보인다. 각 실험의 variance를 낮추기 위해, 총 5번의 실험을 통하여 평균 낸 결과를 사용하였다.

날씨 특징정보가 눈/비가 아닌 상황에서 디지털 롤모델 트윈을 이용하여 각 롤모델 별 가장 높은 선호도를 보이는 두 가지 카테고리를 누적 저장한 결과를 그림 2, 그림 3에 나타내었다. 그림 2를 확인하면, 눈/비가 아닌 상황에서 가상 롤모델 A는 경제와 사회 뉴스 카테고리를 선호할 것으로 예측한 것을 확인할 수 있다. 그리고 그림 3을 확인하면, 마찬가지로 눈/비가 아닌 상황에서 가상 롤모델 B는 연예와 스포츠 뉴스 카테고리를 선호할 것으로 예측한 것을 확인할 수 있다.

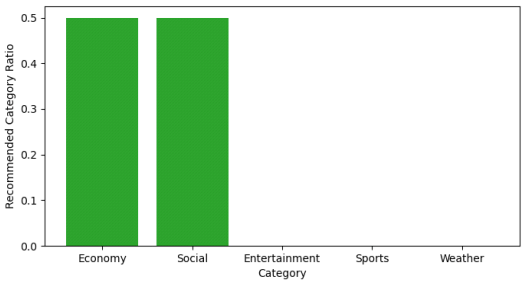

그림 2. 가상 롤모델 A의 선호도를 고려한 뉴스 카테고리 선택 비율
Fig. 2. Ratio of recommended news category considering virtual Role-model A's preference

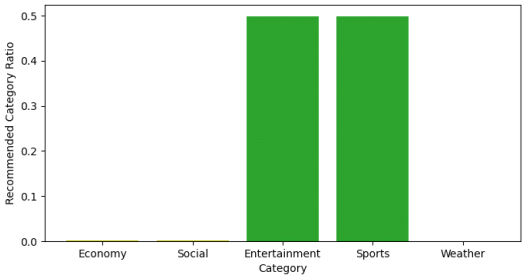

그림 3. 가상 롤모델 B의 선호도를 고려한 뉴스 카테고리 선택 비율
Fig. 3. Ratio of recommended news category considering virtual Role-model B's preference

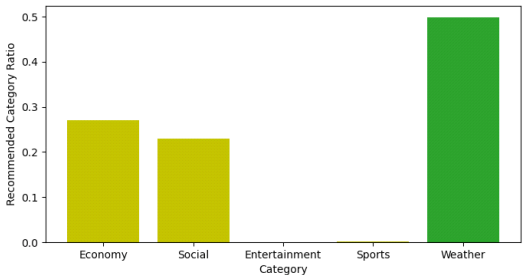

그림 4. 눈/비 날씨 정보를 줬을 때 가상 롤모델 A의 선호도를 고려한 뉴스 카테고리 선택 비율
Fig. 4. Ratio of recommended news category considering virtual Role-model A's preference with rain/snow weather conditions

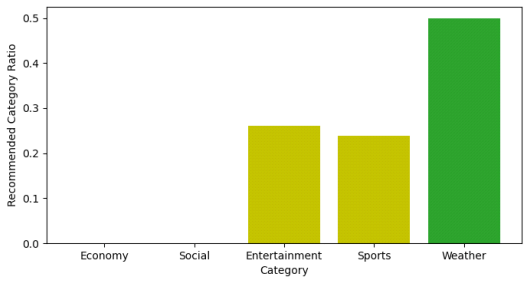

그림 5. 눈/비, 날씨 정보를 줬을 때 가상 롤모델 B의 선호도를 고려한 뉴스 카테고리 선택 비율
Fig. 5. Ratio of recommended news category considering virtual Role-model B's preference with rain/snow weather conditions

눈/비 날씨 맥락 정보가 주어진 상황에서 디지털 롤모델 트윈을 이용하여 각 롤모델 별 가장 높은 선호도를 가지는 두 가지 카테고리를 누적 저장한 결과를 그림 4, 그림 5에 나타내었다.

그림 4를 확인하면, 눈/비 날씨가 진행되는 상황에서는 경제와 사회 뉴스 카테고리를 추천하되 날씨에 관한 뉴스 카테고리를 가장 많이 추천해준 것을 확인할 수 있다. 이때, 눈/비 날씨 정보가 주어진 환경에서는 날씨 뉴스에 더 큰 가중치를 주었던 대로, 다른 뉴스에 비해 날씨 뉴스가 더 높은 선택 비율을 갖는 것을 확인할 수 있다.

그리고 그림 5를 확인하면, 그림 4와 유사하게 연예와 스포츠 뉴스 카테고리를 추천하되 마찬가지로 날씨에 관한 뉴스 카테고리를 가장 많이 추천해준 것을 확인할 수 있다. 이 같은 결과는 앞서 시나리오에서 가정한 각 가상 롤모델의 뉴스 카테고리 선호도와 같은 경향을 띠며, 눈/비와 같이 주어진 맥락 정보를 적절히 고려하여 추천해주는 것을 확인할 수 있다. 따라서 디지털 롤모델 트윈이 주어진 상황을 고려하며 롤모델의 선호도를 적절히 학습해서 롤모델의 선호도 정보를 예측하는 것을 확인할 수 있다.

다음으로 디지털 롤모델 트윈이 실제 사용자에게 활용되는 상황에서의 추천 정확도를 확인한다. 디지털 롤모델 트윈의 성능평가를 위해 디지털 롤모델 트윈과 비교할 총 세 가지의 모델들을 구현하고 뉴스추천 시나리오와 같은 도구, 환경, 뉴스 데이터 및 선택/추천 횟수로 실험을 진행하였다. 비교를 위한 추천 정확도는 아래 수식과 같이 정의한다.

$$\text{추천정확도} = \frac{\text{롤모델 선호 카테고리 중 추천된 개수}}{\text{롤모델 선호 카테고리 총 개수}}$$

추천 정확도를 비교할 때 맥락 정보에 따른 성능을 확인하기 위해 모델별로 눈/비가 오는 날씨와 눈/비가 오지 않는 날씨의 추천 정확도 결과를 비교한다. 디지털 롤모델 트윈과 비교할 각 모델의 정의는 다음과 같다.

Random : 가상 롤모델에게 매번 세 개의 무작위 뉴스 카테고리를 추천해주는 모델이다. 무작위 추천이므로 추천 정확도의 하한성능으로 고려된다.

NC-DRT(No Context - Digital Role model Twin) : 상황에 대한 정보를 제외한 디지털 롤모델 트윈 생성/활용 과정을 가지는 모델이다. 맥락 정보를 학습 과정에 이용하는 것에 영향을 확인하기 위해 고려한다.

IS-DRT(Interval State- Digital Role model Twin): 디지털 롤모델 트윈과 같은 생성/활용 과정을 가지지만, 생성/활용에 이용되는 롤모델에 관한 정보(RF)의 업데이트가 매번 이루어지지 않고 주기적으로 한 번씩만 이루어진다. 구체적으로 IS-DRT(N_{IS}) 모델은 전체 생성과정 중 총 N_{IS}번의 업데이트가 이루어지는 모델을 의미하며, RF 업데이트의 중요성 및 업데이트

횟수의 영향을 확인하기 위해 고려한다.

각 모델의 추천 정확도 실험 결과는 표 2.에서 확인할 수 있다. 먼저 눈/비가 오지 않는 날씨일 때, 디지털 롤모델 트윈은 롤모델의 선호도를 적절히 학습하여 롤모델이 가장 선호하는 선택지들만 추천해주므로 약 99% 이상의 추천 정확도를 보인다. NC-DRT 모델은 상황에 대한 정보를 제외하고 오로지 롤모델의 선호도로만 학습하므로, 맥락 정보가 고정된 상황에서는 디지털 롤모델 트윈과 거의 같게 약 99%의 추천 정확도를 보여준다. IS-DRT 모델들의 경우 디지털 롤모델 트윈과 같은 방식으로 학습하였지만, 롤모델에 관한 정보와 맥락 정보가 매우 적게 업데이트되므로 디지털 롤모델 트윈에 비해 추천 정확도가 저하된 것을 확인할 수 있다. IS-DRT(10)은 전체 학습 기간 5000번 중 총 10번만 업데이트하는 모델로 약 82%의 정확도를 보인다. IS-DRT(30)은 총 30번만 업데이트하는 모델로 약 85%의 정확도를 보인다. 마지막으로 총 100번의 업데이트 하는 모델인 IS-DRT(100)은 다른 IS-DRT 모델들에 비해 높아진 약 94%의 추천 정확도를 보인다. Random 모델의 경우 가상 롤모델의 선호도나 상황과 무관하게 다섯 개 뉴스 카테고리들 중 무작위로 세 가지의 뉴스 카테고리들을 추천해주므로, 추천 정확도는 약 59%인 것을 확인할 수 있다. 정리하면 간단한 뉴스 시나리오에서 맥락 정보가 고정되어있는 눈/비가 오지 않는 날씨일 때, 디지털 롤모델 트윈과 NC-DRT 모델이 약 99%의 매우 높은 추천 정확도를 보인다.

눈/비가 오는 특수한 맥락 정보가 있는 상황일 때는 가상의 롤모델들이 기존의 선호 카테고리들과 날씨 카테고리를 함께 선호하게 된다. 디지털 롤모델 트윈은 롤모델의 선호도와 상황에 따른 선호도를 모두 적절히 학습하므로 여전히 약 99% 이상의 매우 높은 추천 정확도를 보인다. 하지만 특수한 맥락 정보가 있는 상황에서, 맥락 정보를 생성/활용 과정에 포함하지 않는 NC-DRT 모델은 맥락 정보를 학습하지 않기 때문에 맥락 정보가 고정되어있는 상황에서의 추천 정확도보다 조금 저하된 약 90%의 추천 정확도를 보인다. IS-DRT 모델들은 각각 맥락 정보가 고정되어있을 때와 비슷한 추천 정확도를 보인다. 구체적으로 IS-DRT(10) 모델은 약 81%, IS-DRT(30)는 약 86%, IS-DRT(100)는 약 93%의 추천 정확도를 보인다. Random 모델의 경우 맥락 정보가 고정되어있는 상황과 마찬가지로 약 60%의 추천 정확도를 보인다. 정리하면 디지털 롤모델 트윈이 다른 모델들에 비교하여 상황에 따라 적절히 가상 롤모델의 선호도를 학습하여 모든 상황에서 매우 높은 추천 정확도를 보이는 것을 확인할 수 있다.

더 나아가 IS-DRT 각 모델 및 디지털 롤모델 트윈의 추천 정확도를 비교하면 롤모델 특징정보 RF의 업데이트 횟수에 따라 추천 정확도의 차이가 있는 것을 확인할 수 있다. IS-DRT(10), IS-DRT(30), IS-DRT(100), DRT 순으로 상태에 대한 업데이트 횟수가 많아지며 추천 정확도 역시 높아진다. 이를 통해 디지털 롤모델 트윈의 구성요소 중 RF 업데이트 횟수와 같은 변수가 추천 정확도의 차이를 만든다는 것을 확인할 수 있다.

마지막으로 강화학습 기반 디지털 롤모델 트윈의 생성과정 학습 손실 값을 통해 디지털 롤모델 트윈의 수렴성을 확인한다. 롤모델의 선호도를 학습하는데 활용되는 학습률을 각각 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}로 설정하고 50000회의 학습을 진행하며 학습 손실 값을 이동 평균하여 그림 6.에서 비교하였다. 그림 6.를 확

표 2. 뉴스 시나리오에서 모델별 추천 정확도 비교
Table 2. Results of modified recall accuracy of each model in news scenario

	눈/비가 오지 않을 때	눈/비가 올 때
DRT	99.565%	99.180%
NC-DRT	99.500%	90.310%
IS-DRT(10)	82.515%	81.343%
IS-DRT(30)	85.685%	94.725%
IS-DRT(100)	94.725%	93.113%
Random	59.000%	60.200%

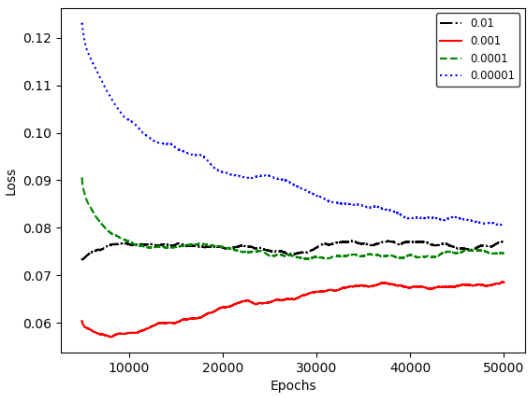

그림 6. 학습률에 따른 이동 평균 손실
Fig. 6. The moving average loss of each learning rate

인하면, 모든 학습률에 대해 손실 값들이 충분히 작은 값을 달성하며, 학습이 진행됨에 따라 모든 학습률의 손실 값들이 점차 수렴한다. 이 실험을 통해 강화학습을 이용한 디지털 롤모델 트윈 시스템의 수렴성을 확인할 수 있다.

V. 결 론

본 논문에서는 강화학습 기법을 이용하여 디지털 롤모델 트윈을 생성하고, 이를 사용자가 자신의 가치 판단에 활용할 수 있는 시스템을 구현하였다. 그리고 실험을 통해 디지털 롤모델 트윈을 선호도에 기반을 둔 뉴스 추천시스템에서 활용할 수 있음을 보였다. 제안한 디지털 롤모델 트윈을 사용하면 주어진 의사결정 문제에 대한 롤모델의 선호도 정보를 예측하고 롤모델의 선호도 정보를 얻고 싶은 사람에게 제공할 수 있다.

하지만 제안한 디지털 롤모델 트윈은 롤모델의 선호도 정보를 학습하기 위해 롤모델이 직접 선택지를 고르는 행위를 여러 번 반복해야 하고, 특정한 서비스 및 시나리오에 적용할 수 있도록 구현된 단점이 있다. 따라서 향후 연구 방향으로 적은 롤모델의 선택 데이터로도 롤모델의 선호도를 학습하고, 범용적인 환경 및 서비스에서 활용될 수 있는 디지털 롤모델 트윈 연구가 고려될 수 있다. 구체적으로 전이학습과 메타러닝 등의 기법을 활용하면 롤모델에 관한 다른 분야에서의 정보 혹은 롤모델과 비슷한 다른 사람들의 선호도 데이터를 사용하여 매우 적은 선택 데이터로도 디지털 롤모델 트윈이 롤모델의 선호도 정보를 학습하여 다양한 분야에 대한 디지털 롤모델 트윈을 생성할 수 있을 것으로 기대된다.

References

[1] J. J. You, J. G. Lee, and W. Choi, "Digital twin technology and standardization trends," *Inf. and Commun. Mag.*, vol. 38, no. 9, pp. 40-47, Aug. 2021.

[2] D. H. Shin, "Metaverse and its future," *Inf. and Commun. Mag.*, vol. 39, no. 4, pp. 55-60, Mar. 2022.

[3] D. H. Kim, Y. W Kim, and K. S. Lee, "Medical digital twin-based dynamic virtual body capture system," *J. KIICE*, vol. 24, no. 10, pp. 1,398-1,401, Oct. 2020.

[4] T. Erol, A. F. Mendi, and D. Doğan, "The digital twin revolution in healthcare," *2020 4th ISMSIT*, pp. 1-7, Oct. 2020. (https://doi.org/10.1109/ISMSIT50672.2020.9255249)

[5] S. Boschert and R. Rosen, "Digital twin—the simulation aspect," *Mechatronic Futures*, pp. 59-74, Jun. 2016. (https://doi.org/10.1007/978-3-319-32156-1_5)

[6] E. VanDerHorn and S. Mahadevan, "Digital twin: Generalization, characterization and implementation," *Decision Support Syst.*, vol. 145, Jun. 2021. (https://doi.org/10.1016/j.dss.2021.113524)

[7] V. Mnih, et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529-533, Feb. 2015. (https://doi.org/10.1038/nature14236)

[8] R. S. Sutton and A. G. Barto. "Reinforcement learning: An introduction," vol. 1, MIT Press Cambridge, 1998.

[9] C. W. Ahn, "Metabus and digital twin," *ie Mag.*, vol. 28, no. 4, pp. 23-27, Dec. 2021.

[10] X. Wang, et al., "Dynamic attention deep model for article recommendation by learning human editors' demonstration," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, pp. 2051-2059, Aug. 2017. (https://doi.org/10.1145/3097983.3098096)

[11] Z. Lu and Q. Yang, "Partially observable markov decision process for recommender systems," *arXiv preprint arXiv:1608.07793*, Aug. 2016. (https://doi.org/10.48550/arXiv.1608.07793)

[12] T. Mahmood and F. Ricci, "Learning and adaptivity in interactive recommender systems," *ICEC '07*, pp. 75-84, Aug. 2007. (https://doi.org/10.1145/1282100.1282114)

[13] G. Shani, D. Heckerman, and R. I Brafman, "An MDP-based recommender system," *J. Mach. Learn. Res.*, pp. 1265-1295, Sep. 2005. (https://doi.org/10.48550/arXiv.1301.0600)

[14] G. Zheng, et al., "DRN: A deep reinforcement learning framework for news

recommendation," in *Proc. 2018 World Wide Web Conf.*, pp. 167-176, Apr. 2018. (https://doi.org/10.1145/3178876.3185994)

[15] H.-T. Cheng, et al., "Wide & deep learning for recommender systems," in *Proc. 1st Wkshp. Deep Learn. for Recommender Syst.*, pp. 7-10, Sep. 2016. (https://doi.org/10.1145/2988450.2988454)

[16] X. Chen, et al., "Generative adversarial user model for reinforcement learning based recommendation system," in *Proc. 36th Int. Conf. on Mach. Learn.*, PMLR pp. 1052-1061, 2019. (https://doi.org/10.48550/arXiv.1812.10613)

[17] Z. Zhao, Y. Liang, and X. Jin, "Handling large-scale action space in deep Q network," *2018 IEEE ICAIBD*, pp. 93-96, 2018. (https://doi.org/10.1109/ICAIBD.2018.8396173)

[18] H. van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," in *Proc. AAAI Conf. Artificial Intell.*. vol. 30, no. 1, Mar. 2016. (https://doi.org/10.1609/aaai.v30i1.10295)

김 지 완 (Ji-Wan Kim)

2017년 3월~현재 : 세종대학교 스마트기기공학과 학사과정
<관심분야> 통신망 자원 할당, 최적화, 인공지능

제 갈 홍 (Hong Je-Gal)

2017년 3월~현재 : 세종대학교 무인이동체공학과 학사과정
<관심분야> 통신망 자원 할당, 최적화, 인공지능

이 승 진 (Seung-Jin Lee)

2017년 3월~현재 : 세종대학교 무인이동체공학과 학사과정
<관심분야> 통신망 자원 할당, 최적화, 인공지능

이 현 석 (Hyun-Suk Lee)

2012년 2월 : 연세대학교 전기전자공학과 학사
2018년 2월 : 연세대학교 전기전자공학과 박사
2018년 3월~2020년 8월 : 연세대학교 전기전자공학과 박사후연구원
2019년 9월~2020년 8월 : Department of Applied Mathematics and Theoretical Physics, University of Cambridge 박사후연구원
2020년 9월~현재 : 세종대학교 지능기전공학부 조교수
<관심분야> 통신망 자원 할당, 인공지능, 최적화
[ORCID:0000-0001-5885-1711]