

MMORPG에서 게임 봇 프로그램 탐지를 위한 플레이어 패턴 변화 모델에 관한 연구*

윤태복, 이지형
성균관대학교 정보통신공학부
tbyoon@skku.edu, jhlee@ece.skku.ac.kr

A Study of Player Changed-pattern Model for Game Bots Detection in MMORPG

Taebok Yoon and Jee-Hyong Lee
School of ICE, Sungkyunkwan University

요 약

온라인 게임에서 오토 프로그램 또는 봇 프로그램으로 인하여 다양한 게임 서비스 피해사태가 발생하고 있다. 특히, 게임 머니 및 아이템의 비정상적인 수집은 게임이 가지는 본연의 재미를 잃어버리게 되고, 궁극적으로 게임 생명주기에 결정적 악영향을 미치게 된다. 본 논문은 게임 봇 감지를 위해 게임 행위 변화 패턴을 수집하고 분석하여 봇 탐지 방법에 적용한다. 인간의 게임 행위 변화 정보와 봇의 게임 행위 변화 정보를 이용하여 유사정도를 측정하고, 봇 탐지 기법에 활용하는 것이다. 실험에서는 서비스 중인 온라인 게임을 이용하여 사용자와 봇의 모델을 생성하고 유사성을 판별하였으며 적절한 결과를 확인하였다.

ABSTRACT

In an online-game, the various game service victimized cases are generated by the bots program or auto program. Particularly, the abnormal collection of the game money and item loses the inherent fun of a game. It reaches ultimately the definite bad effect to the game life cycle. This paper collects and analyzes the pattern of game behavior change for the bots detection method. By using the game activity changing information of the human and game activity changing information of the bots, the degree of resemblance was measured. It utilized in the bots detection method. In an experiment, by using the served online-game, the model of a user and bots were generated and similarity was distinguished. And the reasonable result was confirmed.

Keywords : Game Bots Detection, Changed-pattern, Concept Drifting, Game A.I.

접수일자 : 2010년 12월 15일 심사완료 : 2010년 01월 05일

교신저자(Corresponding Author) : 이지형 교수

※ 본 연구는 한국콘텐츠진흥원의 2010년 문화콘텐츠산업기술지원사업과 차세대 게임 전문교육기관지원사업의 연구 결과입니다.

※ 이 논문은 2010년 한국게임학회 추계학술발표대회에서 우수논문으로 선정된 논문입니다.

1. 서론

온라인 게임은 IT기술의 발달을 기반으로 디지털 콘텐츠 분야에서 지식기반 고부가가치 산업으로 많은 관심을 받고 있다. 하지만, 게임의 선정성과 폭력성을 내세운 무분별한 서비스는 게임 중독, 현실 세계와의 혼돈, 게임 머니 악용 등의 문제를 가지고 있다. 특히, 봇(Bot)이나 치트(Cheat)프로그램을 이용하여 비정상적으로 게임을 운영하고 부당한 이득을 취하는 사용자로 인하여 정상적으로 게임을 진행하는 플레이어에게 불쾌감을 주고, 급기야 전체적인 게임 서비스에 악영향을 미치는 경우가 종종 발생한다.

본 논문에서는 플레이어의 과거 게임 행위 정보와 봇을 이용한 게임 플레이 정보를 수집 분석하여 패턴을 추출한다. 정상적인 게임 플레이 행위 모델과 비정상적인 게임 플레이 행위 모델을 만들고, 새로운 게임 로그 정보를 이 두 모델과 비교하여 인간 플레이 인지, 봇 플레이어 인지를 진단하는 방법을 제시한다. 모델 생성을 위하여 사용한 정보로는 게임 캐릭터가 몬스터를 죽이는데 소요된 시간, 마을 체류 시간, 휴식 상태 추출, 이동 거리, 추적 시간을 이용하였고, 이 정보들의 변화 정도를 모델링하여 인간과 봇을 구분하는데 사용하였다.

실험에서는 서비스 중인 온라인 게임을 이용하여 인간 플레이어의 게임 행위 정보를 수집하고, 봇 게임 데이터를 수집하여 비교 분석하였고, 적절한 결과를 확인하였다.

2. 관련 연구

본 논문에서 다루는 패턴 변화 정보를 이용한 봇 검출 방법은 패턴 변화 모델을 위한 연구와 봇 검출을 위한 연구로 구분할 수 있다. 패턴 변화 정보는 앞서 설명한 바와 같이 인간과 봇을 구분하는 결정적인 정보로 활용한다. 먼저 봇 진단을 위한 연구로, Chen등은 게임 속 캐릭터의 이동에 관

한 총 4가지 속성을 정의하여 게임 봇과 플레이어의 행동 차이를 모델링했으며 3가지 게임 봇과 플레이어의 비교 실험을 통해 제안하는 모델을 검증하였다[1]. 하지만 1인칭 슈팅 게임 분석을 기반으로 모델링 하여 게임 봇의 피해가 가장 심각하게 드러나고 있는 MMORPG에 대한 적용 가능성 여부는 알 수가 없다. Kesteren은 게임 속에서 캐릭터가 가지는 이동 경로에서 방향 정보를 기반으로 하여 게임 봇과 플레이어의 차이를 모델링 하였다[2]. 하지만 방향 정보와 같은 단순한 정보만 가지고는 게임 봇 제작자들이 쉽게 회피 가능한 단점을 가지고 있다.

패턴의 변화를 분석하기 위한 연구로, Aggarwal은 패턴 변화 데이터를 분석하기 위한 다양한 방법을 소개하고 있다. 그는 패턴 변화 데이터를 분석하기 위하여 군집(clustering), 분류(classification), 빈도 패턴 추출, 패턴 변화, 인덱싱 기법, 슬라이딩 기법 등 여러 가지 유용한 방법을 조사하였다[3]. Babcock 등은 패턴 변화 데이터 환경에서 모델 생성과 이슈에 대하여 소개하였고[4], Jain은 다양한 환경에서의 패턴 변화 데이터 발생을 소개하고, 통계적 방법을 이용한 패턴 변화 데이터 분석을 제안하였다[5]. 김진화와 민진영은 연속 발생 데이터의 실시간 분석을 위해 슬라이딩 윈도우(sliding window)를 이용하여 분할하고 의사결정나무 방법을 이용하여 분석하였다[6]. 하지만, 각각의 슬라이딩 윈도우에서 분석을 통하여 만들어진 규칙을 통합하는 과정이 명확하지 않고 단순하게 처리하여 효율성이 낮다. Golab 등은 패턴 변화 데이터의 저장/관리를 위한 다양한 방법을 제시하였다[7]. 이처럼 패턴 변화를 분석하기 위한 연구는 다양하게 시도 되고 있으나 변화 정보를 이용한 모델 생성 방법을 소개한 사례는 찾아 보기 어렵다. 특히, 변화 정보를 모델링하고 이 모델 정보를 이용한 유사도 측정은 그룹생성, 침입탐지 등의 다양한 분야에서 활용가능하다.

3. 게임 행위 패턴 모델 기반 봇 검출

3.1 게임 행위 패턴

- 몬스터 제거 시간(monster killing time)

MMORPG에서 캐릭터가 몬스터를 사냥하는 경우, 몬스터를 죽이는데 걸리는 시간은 얼마나 게임 캐릭터가 공격적으로 행동했는가에 대한 지표로서 정의할 수 있다. 만약 캐릭터가 강력한 기술을 자주 사용할 경우 몬스터를 죽이는 시간은 매우 짧아질 것이며 반대로 일반적인 약한 공격만 자주 사용할 경우 죽이는 시간은 길어질 것이다. 일반적으로 봇은 설정된 공격 방식대로 행동하므로 동일한 종류의 몬스터를 사냥할 경우 죽이는데 걸리는 시간이 비슷하게 나올 것이다. 하지만 사람의 경우 자신의 캐릭터가 위험한 경우 같이 게임 내 적으로 나타나는 다양한 상황에 맞춰 공격적인 상태 변화가 다양하게 나타나므로 몬스터를 공격하기 시작한 시점으로부터 죽이는 시간까지 걸리는 시간 변화가 다양하게 나타난다. 각각의 몬스터 제거 소요시간을 몬스터 공격횟수로 나눈 값을 공격횟수의 오차 범위로 이용하였다. 이 오차 범위는 다시 표준 편차와 평균값을 이용하여 몬스터 제거 시간 변동계수로 활용하였다.



[그림 1] 몬스터 제거 시간 변동량

- 마을 체류 시간(staying in town)

대부분의 MMORPG에서, 게임 캐릭터는 한 가지 이상의 특정한 마을을 저장 포인트로서 지정하는 특징을 가지고 있다. 이렇게 지정된 마을은 그 캐릭터에 대해 인식처로서 작용할 뿐만 아니라 각종 활동

에 대한 준비의 공간으로서의 역할도 수행한다. 따라서 일반적으로 캐릭터는 대부분의 게임 활동을 시작하기 이전에 지정된 마을에서 준비의 시간을 가진다. 봇은 설정된 행동에 따라 지정된 마을에서 매번 같은 행동을 반복하는데 반해 사람은 지정된 마을에서 매번 행동의 변화를 가진다. 따라서 지정된 마을에서 체류하는 시간의 변화 수치를 통해, 봇과 사람과의 차이를 비교할 수 있다. 속성을 위한 마을 체류 시간의 표준 편차와 평균을 이용하여 마을 체류 시간 변동 계수를 생성하였다.



[그림 2] 마을 체류 시간 변동량

- 휴식 상태 추출(rest state extraction)

게임 속에서 게임 캐릭터는 자신의 상태에 문제와 같은 상황이 발생할 경우, 예를 들어 체력 혹은 마력 수치가 낮을 경우 회복을 위해 휴식 행동을 취한다. 일반적으로 봇은 휴식 상태를 위한 조건이 사용자에게 의해 정해져 있다. 따라서 휴식 행동 조건이 만족되는 상황이 발생할 경우 반드시 휴식 행동을 취하게 된다. 반대로 말하면 휴식 상태가 발생했을 경우의 봇 사용자의 캐릭터 상태 값을 살펴보면 유사한 수치를 확인할 수 있다. 플레이어는 플레이어가 자신이 캐릭터에게 휴식이 필요함을 인식하였을 경우에 휴식 행동을 취하므로 휴식 행동이 발생하였을 경우의 상태 값이 다양하게 나타난다. 속성을 휴식 상태 수치에 대한 표준 편차와 평균을 이용하여 휴식 상태 변동계수를 생성하였다.



[그림 3] 휴식 상태 변동량

- 이동 거리(moving distance)

이동 거리 패턴은 게임 캐릭터가 이동 시 어느 정도의 거리의 간격을 보이며 이동하는가에 대한 행동 속성으로 이동 시 발생하는 두 좌표 점과의 거리 변화량을 측정한다. 게임 속에서 이동 시 하나하나의 수치 계산을 가지며 좌표점 이동 명령으로만 구성된 게임 붓 이동과는 다르게 언제든 변화 가능한 사람의 직관에 따르는 플레이어의 이동 패턴과의 차이를 나타내는 속성이다.

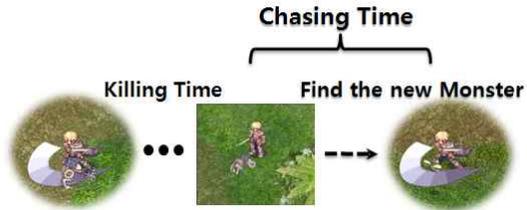


[그림 4] 이동 거리 변동량

- 추적 시간(chase time)

추적 시간 패턴은 게임 캐릭터가 몬스터를 죽인 이후부터 새로운 몬스터를 찾는 데까지 소비한 시간을 측정 후 이것의 변화량을 분석한다. 이 속성이 가지는 의미는 게임 플레이어가 가지는 사냥 혹은 게임 플레이의 몰입도 변화 정도를 나타낸다. 예를 들어 사냥에 집중을 하고 있는 플레이어의 경우 몬스터를 죽인 이후 또 다른 몬스터를 찾기 위한 빠른 행동패턴을 보일 것이나 사냥 집중도가 떨어지는 플레이어의 경우 상대적으로 반응 및 행동 소비시간에서 느린 패턴을 보일 것이다. 다음과 같이 플레이어

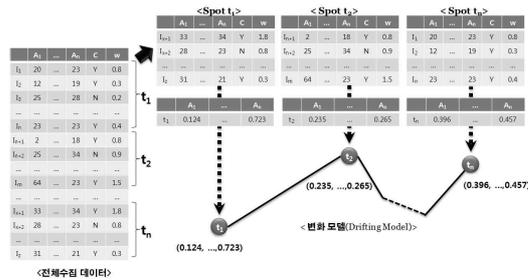
는 집중도에 따라 추적 시간이 변화하므로 큰 변화량을 보이나 게임 붓의 경우 미리 정의되어 있는 행동 및 행동 소비시간을 보이므로 추적 시간의 변화 수치가 낮을 것이다.



[그림 5] 추적 시간 변동량

3.2 패턴 변화 모델링

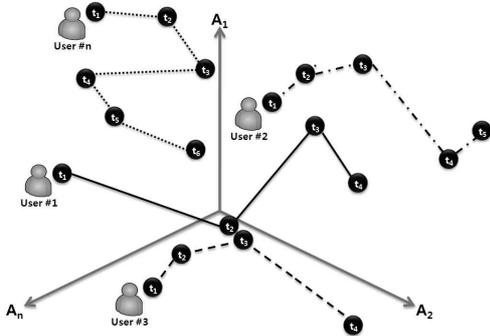
패턴 변화 모델은 수집된 데이터의 변화 흐름을 모델링한 것이다. 수집 데이터를 시간 또는 크기에 따라 분할하고(spot), 각각 분할된 데이터의 knnWeight 고려하여 엔트로피를 계산한다. 각 spot은 속성에 따라 엔트로피 값을 계산하고, 이 값의 변화를 n차원 공간에 사상시킨다. n차원 공간에 사상된 엔트로피 변화 정보를 패턴 변화 모델이라고 정의한다.



[그림 6] 패턴 변화 모델 생성을 위한 데이터 분할

[그림 6]과 같은 방법을 이용하여 전체 수집 데이터를 분할하여 시간 t에 따른 엔트로피 테이블을 생성한다. 생성된 테이블은 n차원 공간에 사상하여 패턴 변화 모델로 나타낸다. [그림 7]는 [그림 6]과 같은 방법을 이용하여 여러 가지 형태의 패턴 변화 모델들을 나타내고 있다. 이와 같이 공간상에 사상된

패턴 변화 모델은 그 흐름이 얼마나 유사한지 비교하여 유사 모델로 선정할 수 있다. 즉, 나와 비슷한 패턴 변화 모델이 있다면, 나의 모델에서 부족한 부분을 그 유사한 모델로부터 보충 할 수 있다. [그림 2]에서 만약 User #3의 t4 이후를 예측한다고 가정할 때, 가장 유사한 모델이 User #2 이므로, User #2의 t5가 사용 될 것이다. 여기서 필요한 기술은 패턴 변화 모델간의 유사한 정도를 측정하는 방법이다. 이에 대해서는 다음 장에서 구체적으로 설명하겠다.



[그림 7] n차원 공간에서의 다양한 패턴 변화 모델 표현 예

3.3 패턴 변화 유사도에 따른 봇 검출

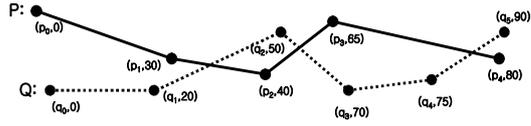
하나의 패턴 변화 모델은 공간상에 사상되었으며, 시간의 흐름에 따라 이동하는 하나의 경로(path)와 유사한 의미를 가지고 있다. 즉, 패턴 변화모델은 안에 있는 지점의 위치와 그곳을 통과한 시각의 쌍의 집합으로 정의할 수 있다. 예를 들면 사용자가 p0 지점을 시간 0에 시작하여 p1을 시간 30에 지났고 p2를 40에, p3을 65에 통과하여 p4에 80에 도달했다면 모델 P는

$$P = \{(p_0, 0), (p_1, 30), (p_2, 40), (p_3, 65), (p_4, 80)\}$$

으로 나타낼 수 있으며, 마찬가지로 임의의 다른 모델 Q는 아래와 같이 나타낼 수 있다.

$$Q = \{(q_0, 0), (q_1, 20), (q_2, 50), (q_3, 70), (q_4, 75), (q_5, 90)\}$$

또한, P와 Q의 지점에 대한 위치를 각각 점으로 표시하여 [그림 8]과 같이 나타낼 수 있다.



[그림 8] 경로 P와 Q의 유사도

이때 시공간에서 발생한 두 패턴 변화 모델 P와 Q는 상호간에 얼마나 유사하다고 해야 할 것인가? 이 질문에 답을 찾기 위해 다시 다음과 같은 세부적인 질문에 대답 할 수 있어야 할 것이다. “두 변화 모델이 얼마나 비슷한 시간을 가지고 이동하였는가?”, “시간에 따른 이동 변화가 얼마나 비슷한가?”, 그리고 “이동 변화의 유사정도와 함께 얼마나 방향이 일치 하는가?” 이 질문은 시간, 거리 그리고 방향 요소를 고려한다면 대답할 수 있을 것이다. 그 이유는 이 세 가지 요소가 시공간 정보를 비교하는 가장 기본이 되는 요소라고 판단되기 때문이다. 본 논문에서는 시간, 거리 그리고 방향 요소를 반영하여 모델 간에 유사정도를 측정한다.

이를 위해 P의 임의의 지점과 Q의 임의의 지점간의 유사도를 정의하였다. 그 이유는 모델의 유사도를 두 모델이 얼마나 유사한 지점(노드, 모델)들이 포함되어 있는가를 이용하여 정의 할 수 있기 때문이다. 예를 들어 P의 한 지점 (pi, ti)와 Q의 한 지점 (qj, tj)의 유사도는 아래와 같이 정의할 수 있다.

$$nodeSim(p_i, q_j) = C_1^D \left[\cos \frac{\theta}{2} \right]^{C_2} C_3^T$$

C₁, C₂, C₃ : 상수

T : |t_i - t_j|

D : 두 지점 p_i, q_j간의 거리

θ : p_i와 q_j의 사잇각

$nodeSim(p_i, q_j)$ 에서 C는 상수 값을 의미하며, 두 지점의 거리(C_1)와 시간(C_3)의 차이가 작을수록 큰 값이 나오도록 하였고, 같은 방향(C_2)을 향할수록 더 큰 가중치 결과를 나타내도록 설계하였다.

$$ModelSim(P, Q) = \max_{\substack{i=0, \\ j=0}}^{m, n} nodeSim(p_i, q_j)$$

$ModelSim(P, Q)$ 는 두 변화 패턴 모델의 유사 정도를 비교하여 최적의 유사도 값을 얻기 위한 수식이다. 앞서 설명한 $nodeSim$ 을 이용하게 되는데, 패턴 변화 모델의 크기에 따라 무수하게 많은 경우의 조합 발생하는 문제를 가지고 있다.

$$ModelSim(P, Q) = g(p_i, q_j)$$

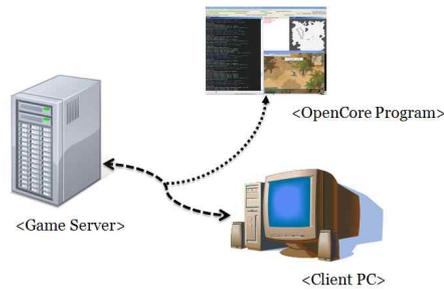
이것은 동적 정합법(dynamic time warping : DTW)과 유사한 방법으로, 위의 수식은 동적 프로그래밍 (Dynamic programming) 기법으로 해결할 수 있다. 동적 정합법을 이용하여, 두 경로에 속한 각 지점에 대해 이동 순서를 고려하여 매칭을 실시한다.

$$g(p, q) = \begin{cases} \max \begin{cases} nodeSim(p, q) + g(p-1, q) \\ nodeSim(p, q) + g(p, q-1) \\ nodeSim(p, q) + g(p-1, q-1) \end{cases} & , \text{if } i \geq 0 \& j \geq 0 \\ 0 & , \text{otherwise} \end{cases}$$

이와 같은 방법을 이용하여 각 패턴 변화 모델간의 유사도를 측정한다. 측정된 유사도는 유사한 패턴 변화를 가지는 그룹(group)을 생성하거나, 나와 유사한 성향을 가지는 파트너를 매칭(matching)하기 위한 기반 기술로 활용가능하다.

4. 실험 및 검증

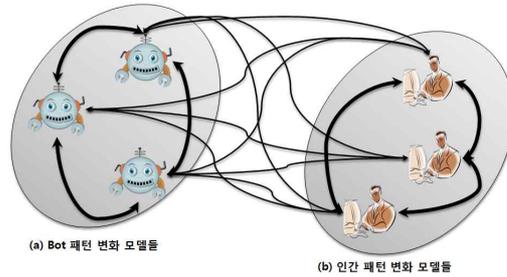
행동 패턴 모델의 검증을 위해 [그림 9]와 같이 국내 G게임사의 R게임을 게임 기반 봇 프로그램 'OpenCore'를 사용하여 게임 플레이 로그를 수집하였다. 이 봇 프로그램의 특징은 무료로 배포되며 지금까지 오랜 시간 동안 개발자들이 온라인 포럼에서 연구의 목적으로 공동으로 개발 했고, 거의 대부분의 게임 및 A.I. 세부요소 설정이 가능하며 사용자가 직접 NPC의 A.I.를 구축할 수 있는 장점을 가지고 있다.



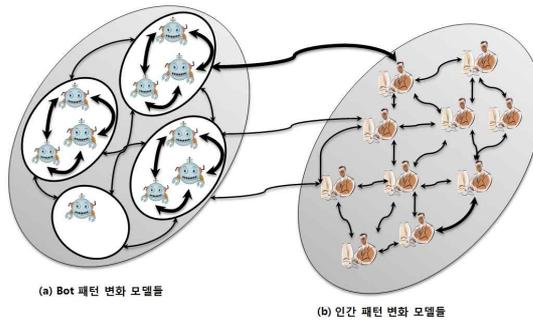
[그림 9] 실험 환경 소개

실험 데이터 수집을 위해 S대학의 학생 15명을 공개 모집하였다. 실험에 참여한 사람은 10시간 동안(1회 접속 시 최소 3시간 유지) 게임을 진행하게 하였고, 수집된 데이터를 사람의 게임 행위 데이터로 이용하였다. 봇 게임 데이터의 경우, 다양한 봇 형태를 수집하기 위해 10가지 다른 환경에 각각 다른 설정을 주었다. 또한, 게임 진행 과정에 따라 패턴 변화를 조절하기 위해 1가지 형태의 봇은 그 안에서 네 가지 변하는 모습을 보이도록 설정하였다. 하나의 패턴 변화는 30시간 동안 유지되며, 그 동안의 봇이 생성한 로그 정보를 수집하였다. 다시 정리하면, 10가지의 다양한 봇과, 1개의 봇이 네 가지 패턴 변화를 가지고 있으며, 1개의 패턴 변화를 위해 30시간 동안 데이터를 수집하였다. 전체 1,200시간 동안의 봇 게임 데이터를 수집하였다. 실험의 목적은 봇 탐지를 위한 모델을 생성하는 것이다. 봇 탐지를 위해 두 가지 모델을 생성하였

다. 하나는 봇 데이터를 이용하여 생성한 봇 모델이고, 다른 하나는 인간 플레이어의 게임 정보를 이용하여 생성한 인간 모델이다. [그림 10]은 봇과 인간의 패턴 변화 모델 유사도에 따라 예측되는 그룹화 모습이다. 아마도 봇 모델들은 봇 모델들과 더 유사하고, 인간 모델들은 인간 모델들과 더 유사하다고 나타날 것이다. [그림 10]과 [그림 11]에서 선의 두께는 앞에서 제시한 $ModelSim(P, Q)$ 를 이용한 유사 정도를 나타내며, 두꺼울수록 유사 정도가 높다는 것을 의미한다. 상대적으로 봇과 인간은 낮은 유사도를 보일 것이다. [표 1]은 생성된 패턴 변화 모델을 이용하여 봇과 봇, 인간과 인간, 봇과 인간의 모델 유사도를 측정된 결과이다. 봇과 봇은 어느 정도 유사한 결과를 보이고 있으나(굵은 글씨), 인간과 인간의 유사도는 봇과 인간의 유사도 보다 낮은 결과를 보이고 있다. 이는 인간이 가지는 게임 패턴이 다양하기 때문에 쉽게 그룹화가 되지 않기 때문이다. 따라서 [그림 10]에서 예측했던 인간 그룹은 [그림 11]과 같이, 그룹을 찾지 못하고 개별적인 모델 모습을 보이고 있다. 반면 봇의 경우, 3가지 타입이 다른 봇에 비하여 상대적으로 유사도가 높은 모습을 보여 주었다.



[그림 10] 실험 전 예측 상황



[그림 11] 봇과 인간 플레이어 패턴 변화 구분 실험 결과

[표 1] 인간 10명, 봇 10개체를 이용한 유사도 측정 결과

		Bot										Human											
		Type1		Type2			Type3			Type4													
		#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10		
Bot	#1	1.00	0.58	0.61	0.41	0.38	0.33	0.38	0.37	0.35	0.32	0.43	0.30	0.14	0.16	0.32	0.49	0.24	0.24	0.25	0.11		
	#2		1.00	0.52	0.38	0.33	0.41	0.37	0.31	0.34	0.33	0.25	0.41	0.16	0.17	0.33	0.51	0.26	0.15	0.32	0.12		
	#3			1.00	0.39	0.38	0.38	0.39	0.44	0.33	0.37	0.24	0.31	0.24	0.19	0.35	0.53	0.22	0.19	0.33	0.11		
	#4				1.00	0.67	0.59	0.41	0.42	0.31	0.26	0.34	0.18	0.34	0.11	0.02	0.45	0.35	0.17	0.39	0.19		
	#5					1.00	0.52	0.42	0.49	0.25	0.28	0.29	0.06	0.37	0.31	0.03	0.24	0.31	0.16	0.25	0.17		
	#6						1.00	0.38	0.26	0.34	0.21	0.27	0.12	0.39	0.24	0.04	0.27	0.41	0.11	0.28	0.16		
	#7							1.00	0.58	0.53	0.15	0.85	0.24	0.38	0.28	0.41	0.36	0.37	0.15	0.26	0.21		
	#8								1.00	0.59	0.12	0.24	0.31	0.33	0.45	0.42	0.35	0.34	0.15	0.27	0.21		
	#9									1.00	0.19	0.21	0.29	0.21	0.24	0.39	0.15	0.31	0.17	0.22	0.14		
	#10										1.00	0.16	0.31	0.25	0.27	0.25	0.19	0.21	0.16	0.23	0.19		
Human	#1											1.00	0.32	0.25	0.29	0.24	0.25	0.22	0.21	0.32	0.26		
	#2												1.00	0.29	0.27	0.29	0.24	0.29	0.57	0.34	0.34		
	#3													1.00	0.25	0.27	0.24	0.34	0.41	0.37	0.25		
	#4														1.00	0.24	0.37	0.29	0.21	0.56	0.27		
	#5															1.00	0.28	0.34	0.31	0.36	0.38		
	#6																1.00	0.33	0.37	0.31	0.34		
	#7																	1.00	0.39	0.35	0.25		
	#8																		1.00	0.34	0.17		
	#9																			1.00	0.11		
	#10																				1.00		

즉 인간은 같은 인간과 변화하는 패턴이 유사할 것이고, 봇 역시 같은 봇과 변화 패턴이 유사할 것이라는 가정([그림 10])으로, 이 두 부류를 구분하는 정보로 패턴을 활용하려 했지만, 인간 패턴 변화의 다양성으로 인하여 인간의 변화 패턴을 찾는 데 실패하였다. 하지만, 봇의 변화 패턴은 일관된 모습([그림 11])을 확인할 수 있었으며 이 정보를 이용하여 봇 유무를 판단하는 기술로 활용가능하다.

5. 결론 및 향후 연구

본 논문에서는 게임 플레이어의 과거 게임 행위 정보와 봇을 이용한 게임 플레이 정보를 수집 분석하여 변화 패턴 모델을 생성하는 방법을 제안하였다. 변화 패턴 모델 생성을 위하여 사용한 정보로는 게임 캐릭터가 몬스터를 죽이는데 소요된 시간, 마을 체류 시간, 휴식 상태 추출, 이동 거리, 추적 시간을 이용하였고, 이 정보들의 변화 정도를 모델링하여 인간과 봇을 구분하는데 사용하였다. 실험에서는 인간 플레이어로부터 수집한 게임 데이터와 봇 플레이어로부터 수집한 게임 데이터를 분석하여 변화 패턴 모델을 만들고 비교 하였다. 인간 변화 패턴의 다양성으로 인하여 모델 생성이 어려웠지만, 봇 변화 패턴 모델은 적절하게 생성되었다. 이 정보는 모델 유사도를 이용하여 새로운 게임 패턴이 수집되었을 때 봇인지 아닌지를 판단하는 유용한 정보로 활용 가능하다.

향후에는 실험 환경을 확대하여 보다 많은 인간 플레이어로부터 데이터를 수집하고 분석하여 인간의 변화 패턴을 모델링하기 위한 연구가 필요하겠다.

참고문헌

- [1] K. Chen, A. Liao, H. K. Pao and H. H. Chu, "Game Bot Detection Based on Avatar Trajectory," Entertainment Computing (ICEC), pp. 94-105, 2008.
- [2] M. Kesteren, J. Langevoort, F. Grootjen. "A step in the right direction: Botdetection in MMORPGs using movement analysis," In Proceedings of the 21th Belgian-Dutch Conference on Artificial Intelligence (BNAIC2009), 2009.
- [3] C. C. Aggarwal, "Data Streams Models and Algorithms", Springer US, pp. 1-7, 2007.
- [4] B. Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom, "Models and issues in data stream systems", ACM SIGMOD-SIGACT-SIGART Symposium on principles of database systems, 2002.
- [5] A. Jain, "Statistical Mining in Data Streams", Ph.D. Dissertation, University of California, Santa Barbara, 2006.
- [6] 김진화, 민진영, "연속발생 데이터를 위한 실시간 데이터 마이닝 기법", 한국경영과학회지, 제 29권, 제4호, pp. 41-60, 2004.
- [7] L. Golab, and M. T. Oszu, "Issues in Data Stream Management", SIGMOD Record, Vol. 32, No. 2, 2003.



윤 태 복 (Yoon, Taebok)

2001년 공주대학교 전자계산학과(이학사)
2005년 성균관대학교 컴퓨터공학(공학석사)
2010년 성균관대학교 컴퓨터공학(공학박사)
2010년-현재 성균관대학교 BK21 연구원

관심분야 : 사용자 모델링(User Modeling), 게임 인공
지능(Game A.I.), 지능시스템



이 지 형 (Lee, Jee-Hyong)

1993년 한국과학기술원 전산학과(학사)
1995년 한국과학기술원 전산학과(석사)
1999년 한국과학기술원 전산학과(박사)
2002년-현재 성균관대학교 정보통신공학부 부교수
2008년-현재 한국게임학회 정회원

관심분야 : 지능시스템, 기계학습, 온톨로지
