Investigating the Application of Action Model Learning for Player Modeling

by Abhijeet Krishnan

NC STATE UNIVERSITY

POEM

Principles of Expressive Machines

Outline

- Introduction
- Motivation
- Related Work
- Background
  - Planning Terminology
  - Action Model Learning (AML)
  - Sokoban
- AML for Player Modeling
  - Learning
  - Using
  - Evaluating
- Evaluation
  - Feasibility, Usefulness, Domain-agnosticity
  - Comparing AML Algorithms
- Discussion & Future Work
- Conclusion

Introduction

- This paper explores the application of a technique from the area of planning to player modeling
- The technique is called **action model learning** (AML)
- A framework of how to use it for player modeling
- An evaluation of it as player modeling technique

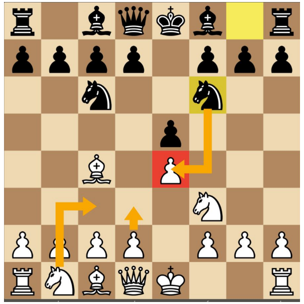
Player Modeling

Opponents

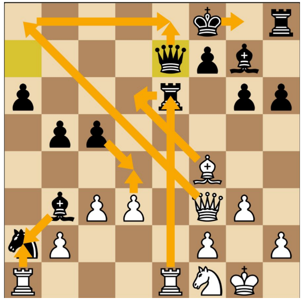
Too easy!

Too difficult!

Puzzles



Too basic!



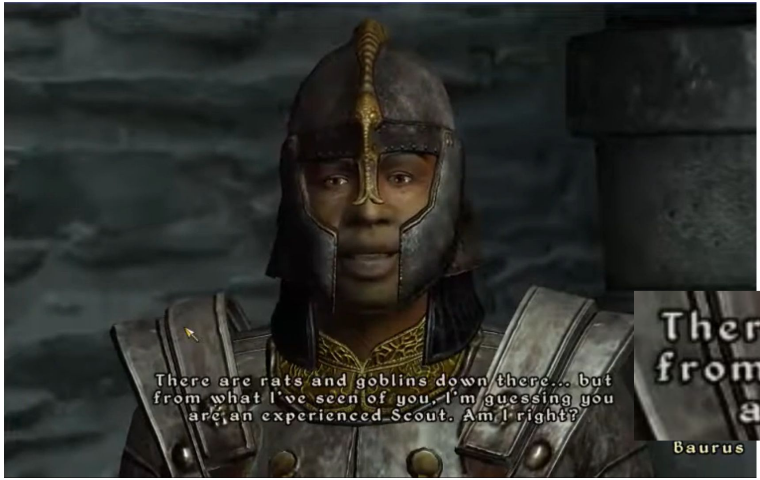
Too advanced!

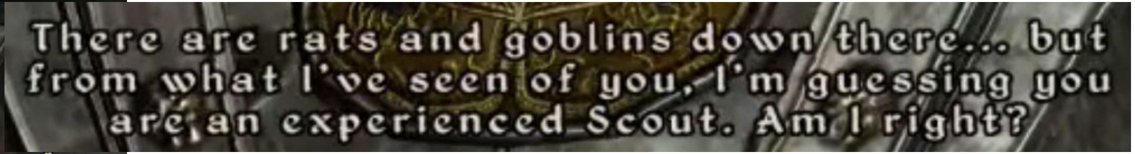
Player Modeling (contd.)

- **What it is**: the study of computational models of players in games (Yannakakis et. al. 2013)
- **Why we do it**: making predictions about the player
- Examples
  - Drivatars in *Forza Motorsport 5* (Turn 10 Studios)
  - Class recommendation in *The Elder Scrolls IV: Oblivion* (Bethesda Softworks)
  - Self-organizing maps in *Tomb Raider: Underworld* (Drachen, Canossa and Yannakakis 2009)
- Player modeling questions -
  - What do we model?
  - How do we model?
  - Why do we model?

Player Modeling: Practice

Drivatars in *Forza Motorsport 5*

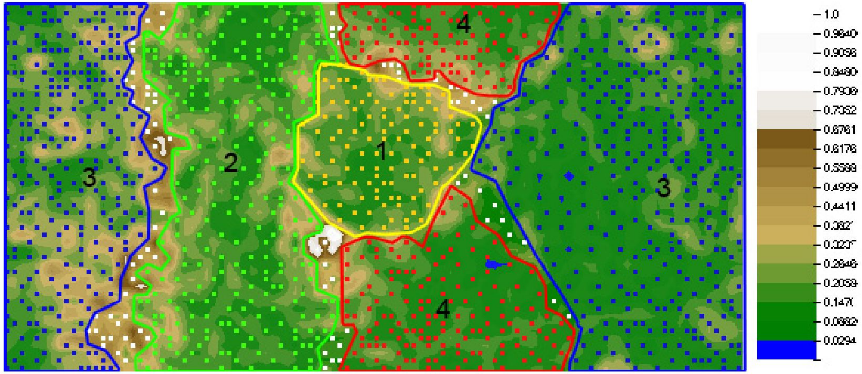




Class recommendation in *The Elder Scrolls IV: Oblivion*

Player Modeling: Research

Screenshot from *Tomb Raider: Underworld*



U-Matrix showing player clusters obtained from highest-performing SOM

Self-organizing maps in *Tomb Raider: Underworld* (Drachen, Canossa and Yannakakis 2009)

Motivation

Lack of **domain-agnostic** player modeling techniques

- Reliance on knowledge-engineering documented in player modeling survey (Hooshyar, Yousefi and Lim 2018)

Need to **explain** players' cognitive processes

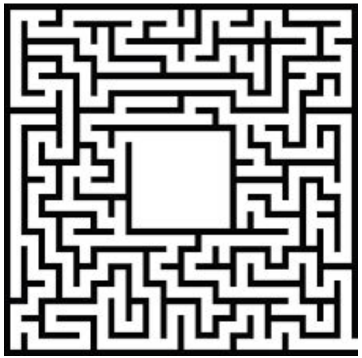
- Players' build mental models of games which change over time (Boyan, McGloin and Wasserman 2018)

Related Work

- Domain-agnostic approaches to player modeling
  - Theoretical model-based
    - Snodgrass, Mohaddesi, and Harteveld (2009)
  - Physiological readings
    - Noguiera et. al. (2014)
  - Deep learning
    - Wang et. al. (2018)
    - Pfau, Smeddinck, and Malaka (2018)
    - Summerville et. al. (2016)
- Action model learning for humans
  - Human-aware planning (Chakraborti et. al. 2017)

Background: Planning Terminology

- *Planning*: finding a sequence of *actions* to reach a desired goal *state* in a specific *domain*
- States modeled as collection of facts about the world (*predicates*)
- Actions modeled as <*preconditions*, *effects*>
- Domain: a description of the “physics” of the world
- Action model: set of actions in the domain
- PDDL is a modeling language commonly in planning



Background: Action Model Learning (AML)

- Motivated by the difficulty of authoring domain models (Kambhampati 2007)
- Used to learn an action model from *plan traces* (*trajectory*)
- Plan trace: sequence of state-action transitions

```
(:action move
   :parameters (?p - player ?from ?to - location
?dir - direction)
   :precondition (and (at ?p ?from)
                      (clear ?to)
                      (MOVE-DIR ?from ?to ?dir)
                      )
   :effect       (and (not (at ?p ?from))
                      (not (clear ?to))
                      (at ?p ?to)
                      (clear ?from)
                      )
   )
```

Sample action model (with one action)

```
(trajectory

(:objects dir-down -  direction [...])

(:init (is-goal pos-01-01) [...])

(:action (push-to-nongoal player-01 stone-01
pos-01-01 pos-01-02 pos-01-03 dir-down))

(:state (is-goal pos-01-01) [...])

)
```

Sample trajectory (truncated for brevity)

Background: Sokoban

- Tile-based puzzle game
- Objective is to push all blocks onto goal tiles
- Frequently used to test automated planners (Coles et. al. 2012)

[Sokoban Demo Link]

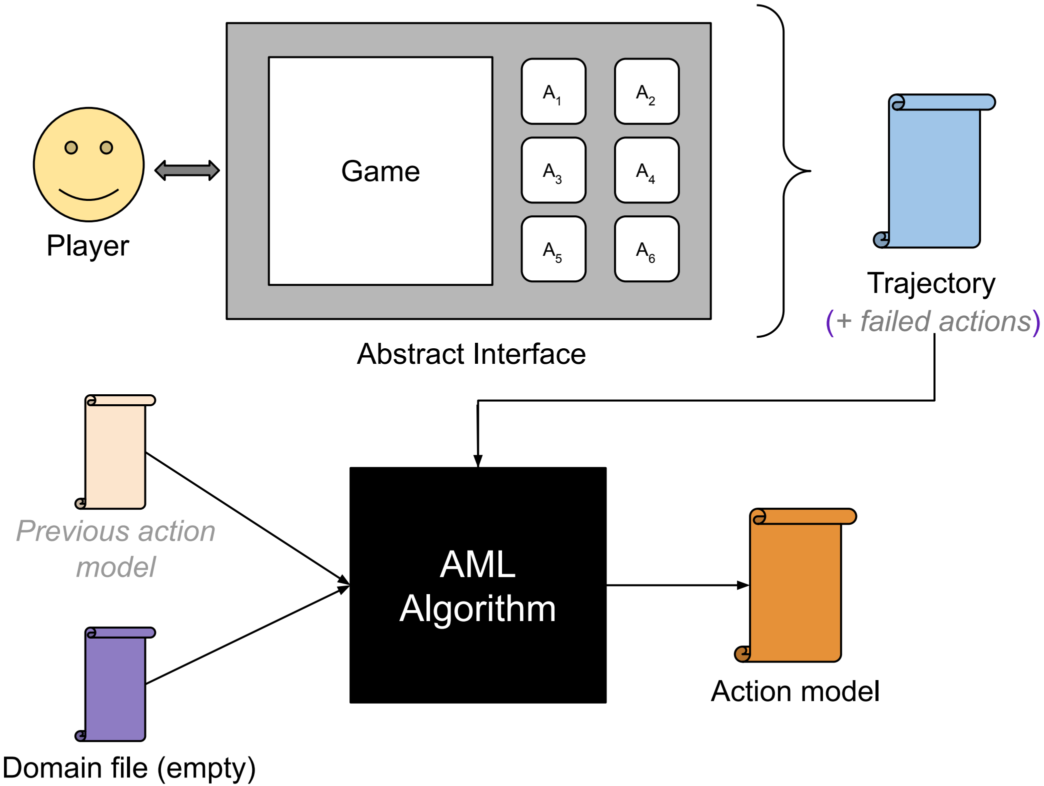
Outline

- Introduction
- Motivation
- Related Work
- Background
  - Planning Terminology
  - Action Model Learning (AML)
  - Sokoban
- AML for Player Modeling
  - Learning
  - Using
  - Evaluating
- Evaluation
  - Feasibility, Usefulness, Domain-agnosticity
  - Comparing AML Algorithms
- Discussion & Future Work
- Conclusion

AML for Player Modeling

- Learning a player model using AML
- Using the player model to predict something useful about the player
- Evaluating the player model for correlation with the player’s mental model

Learning the model



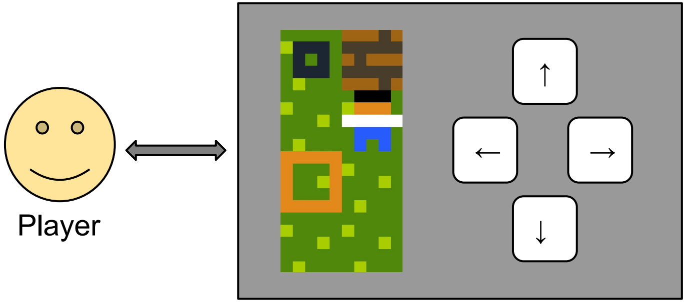
Failed actions: Actions chosen to be taken by the player, but are prohibited due to its preconditions not being met

$\mathbf{A_n}$: actions possible in the domain (with objects they act upon)

Learned action model treated as player model

Action model represents player's knowledge of game mechanics

Learning the model (contd.)



Sokoban

```
(trajectory

(:objects dir-down -  direction [...])

(:init (is-goal pos-01-01) [...])

(:state (is-goal pos-01-01) (is-nongoal pos-02-01)
[...]))
```

Trajectory (PDDL-like syntax)

```
define (domain
sokoban-sequential)
  (:requirements :typing)
  (:types thing location
direction - object
            player stone - thing)
  (:predicates (clear ?l -
location) [...]
```

Empty Sokoban domain file (in PDDL)

FAMA

```
(:action move
   :parameters (?p - player ?from ?to -
location ?dir - direction)
   :precondition (and (at ?p ?from)
                      (clear ?to)
                      (MOVE-DIR ?from ?to
?dir)
                      )
   :effect       (and (not (at ?p ?from))
                      (not (clear ?to))
                      (at ?p ?to)
                      (clear ?from)
                      )
   )
```

Action model (in PDDL)

Using the model

Learned	Reference	Category
(clear ?to)	(clear ?to)	correct
(at ?p ?from)	(at ?p ?from)	correct
(is-nongoal ?from)		extra
	(move-dir ?from ?to ?dir)	missing
(clear ?from)	(clear ?from)	correct
(at ?p ?to)	(at ?p ?to)	correct
(move-dir ?from ?to ?dir)		extra
(not (at ?p ?from))	(not (at ?p ?from))	correct
(not (clear ?to))	(not (clear ?to))	correct

- Outputs a quantitative measure of a player's understanding of game mechanics
- Method based on Aineto, Celorrio and Onaindia (2019)
- Algorithm is as follows -
    - Compare learned action model with ground truth action model
    - For each action, count the number of predicates in preconditions and effects in the learned model which are
        - Correct (true positive)
        - Extra (false positive)
        - Missing (false negative)
    - Compute F-1 score for each action
- F-1 score represents a player's understanding of the game mechanics

Evaluating the model

- Goal: compare learned action model with player's mental model (of game mechanics)
- No documented method of eliciting action model from player
- Hypothetical evaluation method, not yet used in actual user study -
  - Test knowledge of mechanics through prediction of action success (preconditions) and validity of post-state (effects)
  - Present questions of 2 types -
    i. State + Action -> is the action applicable in this state?
    ii. State + Action = New state -> is the new state what we actually get?
  - Compare player predictions with predictions made using learned action model to measure accuracy

Evaluation: Feasibility, Usefulness, Domain-agnosticity

- Feasibility: can this method be used to learn a player model?
  - Yes, we successfully learn player models using FAMA with the Sokoban domain
- Usefulness: does the learned player model have some functionality?
  - Yes, we make predictions regarding player's mechanical knowledge using the learned player model
- Domain-agnosticity: can this method be used to learn player models across multiple domains?
  - Yes, we successfully learn player models across two different domains (N-puzzle, Hanoi) using the same method

Evaluation: Comparing AML Algorithms

- Goal: comparing various AML algorithms for their suitability to player modeling
- Metrics
    - Time taken
    - Memory consumed
    - Ability to use previous action models
    - Ability to use failed actions
    - ~~Accuracy~~
- Dataset: manually generated trajectories from 3 Sokoban levels of increasing complexity (L1, L2, L3)

Evaluation: Comparing AML Algorithms (contd.)

Algorithm	Time (s)			Memory (MB)			$f_{AML}(\tau, \alpha)$	$\langle s - a_f - s \rangle \in \tau$
	L_1	L_2	L_3	L_1	L_2	L_3		
ARMS	0.05	0.01	0.67	17.06	16.66	91.75	✗	✗
LOCM	0.21	0.19	0.15	21.50	34.35	74.64	✗	✗
LOCM2	0.25	0.20	0.10	20.93	33.72	70.97	✗	✗
LOUGA	0.92	0.77	3.28	25.06	37.74	64.04	✗	✗
FAMA	10.59	1572.60	—	21.19	4632.52	—	✗	✗

Discussion & Future Work

- Evaluation of player models learned using AML
  - Eliciting action models from players directly
  - Mismatch between representation and implementation
    - Predicates in domain might not be presented to the player
    - Actions in the domain might not correspond to the interface provided to the player
- Improving AML algorithms
  - Incorporating cognitive theories of mental model formation
    - Failed actions
    - Previously learned models
  - New AML algorithm for player modeling - Blackout (Krishnan, Williams and Martens 2020) at AIIDE 2020
- Domain-agnosticity of AML
  - Pros: easily applicable to multiple games
  - Cons: requires games to be representable in PDDL

Conclusion

- We find that AML is a viable technique for domain-agnostic player modeling
- We present a method to use it to quantify player’s understanding of game mechanics
- We found existing AML algorithms to be performant for player modeling
- We suggested improvements to make AML algorithms better player modeling techniques
- We propose the evaluation of action model-based player models as a useful challenge to solve

References

- Yannakakis, G. N.; Spronck, P.; Loiacono, D.; and André, E.2013. Player modeling
- A. Drachen, A. Canossa and G. N. Yannakakis, "Player modeling using self-organization in Tomb Raider: Underworld," 2009 IEEE Symposium on Computational Intelligence and Games, Milano, 2009, pp. 1-8, doi: 10.1109/CIG.2009.5286500.
- Hooshyar, D.; Yousefi, M.; and Lim, H. 2018. Data-driven approaches to game player modeling: A systematic literature review. ACM Comput. Surv. 50(6).
- S. Kambhampati, Model-lite planning for the web age masses: the challenges of planning with incomplete and evolving domain models, in: National Conference on Artificial Intelligence, (AAAI-07), 2007.
- Snodgrass, S.; Mohaddesi, O.; and Harteveld, C. 2019. Towards a generalized player model through the peas framework. InProceedings of the 14th International Conference on the Foundations of Digital Games, FDG '19. New York,NY, USA: Association for Computing Machinery.
- Nogueira, P. A.; Aguiar, R.; Rodrigues, R. A.; Oliveira,E. C.; and Nacke, L. 2014. Fuzzy affective player models:A physiology-based hierarchical clustering method. In Tenth Artificial Intelligence and Interactive Digital Entertainment Conference.
- Wang, P.; Rowe, J.; Min, W.; Mott, B.; and Lester, J.2018. High-fidelity simulated players for interactive narrative planning. InProceedings of the 27th International Joint Conference on Artificial Intelligence, IJCAI '18, 3884–3890. AAAI Press.
- Pfau, J.; Smeddinck, J. D.; and Malaka, R. 2018. Towards deep player behavior models in mmorpgs. In Proceedings of the 2018 Annual Symposium on Computer-Human Interaction in Play, CHI PLAY '18, 381–392. New York, NY,USA: Association for Computing Machinery.
- Summerville, A.; Guzdial, M.; Mateas, M.; and Riedl, M. O.2016. Learning player tailored content from observation:Platformer level generation from video traces using lstms. In Twelfth Artificial Intelligence and Interactive Digital Entertainment Conference.
- Chakraborti, T.; Kambhampati, S.; Scheutz, M.; and Zhang,Y. 2017. AI challenges in human-robot cognitive teaming.CoRRabs/1707.04775.
- Serafini, L., and Traverso, P. 2019. Incremental learning of discrete planning domains from continuous perceptions.arXiv preprint arXiv:1903.05937.
- Matsuda, N.; Cohen, W. W.; and Koedinger, K. R. 2015.Teaching the teacher: Tutoring simstudent leads to more effective cognitive tutor authoring.International Journal of Artificial Intelligence in Education 25(1):1–34.
- Coles, A.; Coles, A.; Olaya, A. G.; Jim ́enez, S.; L ́opez, C. L.;Sanner, S.; and Yoon, S. 2012. A survey of the seventh international planning competition.AI Magazine 33(1):83–88. Copyright - Copyright Association for the Advancement of Artificial Intelligence Spring 2012; Document feature - ;Diagrams; Last updated - 2018-10-06.
- Aineto, D.; Celorrio, S. J.; and Onaindia, E. 2019. Learning action models with minimal observability.Artificial Intelligence 275:104 – 137.
- Reynolds, M. 2019. How forza 5's cloud-based drivatar's work. https://www.digitalspy.com/videogames/xbox-one/a530468/forza-motorsport-5-how-the-xbox-one-racers-drivatar-system-works. Accessed: January 23, 2020.

Questions?