User Assistance for Serious Games Using Hidden Markov Model

Vivek Yadav[1], Alexander Streicher[2], and Ajinkya Prabhune[1]

[1] SRH Hochschule Heidelberg, Germany
[2] Fraunhofer IOSB, Karlsruhe, Germany

Abstract. Serious Games, i.e., games not just for pure entertainment and with characterizing goals, are gaining huge popularity for the purpose of education and training. To further increase the learning outcome of serious games, assistance functionalities like adaptive systems observe the users and try to guide them to achieve their learning objectives. The research question is how to model the user's behavior, their progress, and how to determine the best adaptation strategies to motivate the users and provide assistance whenever required. Using experience-data in a serious game is one approach to develop and train models for adaptivity. In this paper, we present SeGaAdapt, an adaptive framework that is based on a Hidden Markov Model (HMM) algorithm for providing dynamic user-assistance and learning analytics for a serious game. For the development and training of the HMM, we use activity streams or user interaction data gained from an Experience API (xAPI) tracker. The adaptivity mechanism uses the HMM to analyze the current state of the user (player) in order to predict the best feasible activity for future states. Technical verification of this work-in-progress implementation shows the feasibility of the approach and hints at future research directions.

Keywords: serious games · xapi · adaptivity · hidden markov model.

1 Introduction

Educational Serious Games (SG) are designed for the purpose of education and training, and they are among the widely used instruments for its respective purpose. Adaptive and personalized SGs could lead to better outcomes from the playing exercise [1] supported by Machine Learning (ML) but there is still a large potential to enhance the adaptivity through the implementation of AI [1]. Activities in the SG are not like levels or stages in entertainment games, wherein the players (users) are driven by the challenges associated with pure entertainment rather, the activities are intended to teach, educate, train, or guide, without being less engaging and without the users losing interest. Activities are a very important part of the learning process in SGs, thus they must be monitored carefully and provided with assistance during the engagement process to the player whenever required.

[2] European Conference on Technology Enhanced Learning
EC-TEL 2020: Addressing Global Challenges and Quality Education pp 380-385

In this paper, we present our work in progress aiming towards an adaptive framework (SeGaAdapt) for SG based on Hidden Markov Model (HMM) for assistance that accepts *Experience API* (xAPI) statements [2] as input and processes them, and if assistance is required then it provides the assistance related to activities as output. The part of the SeGaAdapt that processes the input is equipped with (1) a performance tracker to decide when to adapt, (2) trained HMM to devise effective assistance providing aid to perform activities, (3) a compass for helping the player to perform the activities correctly. SeGaAdapt uses the performance tracker (see section 2.1) to monitor the player's performance through xAPI. If the performance of the player drops below a certain threshold, then the SeGaAdapt initiates the operation to assist the player to perform the activities correctly until the performance level goes above the threshold. To compose the assistance strategy the framework uses xAPI Statements and HMM. The xAPI statements portray the player's activity, it is used as input for the HMM. The framework uses the inference capability of HMM to check for the most feasible activity given at a particular state. Using the Markov chain's ability to predict future states, the compass assists the player to perform the activities in the correct order.

Implementation of the HMM-based assistance mechanism for adaptivity being used in SeGaAdapt is new from the perspective of SGs, but HMM has been previously used in SGs for different purpose. Bunian et al. [3] used HMM to model the individual behavior patterns of the players and to classify players' characteristics. Derbali et al. [4] used HMM to identify physiological patterns to access motivational strategies.

2 An Adaptive Framework Using Hidden Markov Model

First, we train the HMM by using the player data (from multiple players) of SG comprising of activities performed while playing. Player data can be previously recorded xAPI statements or activity records in a database. To construct the most feasible strategy, the activities with the highest score or point gain from the player data is used for training HMM. A trained HMM comprises (a) transition probabilities of the states and (b) emission probabilities of the observables from the states. After training the HMM, the framework can start receiving the xAPI statements in the 'actor-verb-object' format which contains information related to the activities that take place within the Serious Game (SG) at a given state. The framework uses the xAPI statements and conducts (1) performance tracking of the player, (2) use HMM to discover misfit activities and flag feasible activity to the state in which misfit activities was performed, (3) prediction of future states using Markov chain property and assist (using feasible activity flagged from step 2) in advance.

2.1 Performance Tracking

Player's performance can typically be measured based on player score or multiple failures or activity report. The performance tracker processes the received xAPI statements and triggers the adaptivity requirement signal whenever player's performance reaches the minimum threshold, thus initiating assistance.

2.2 Use of Hidden Markov Model

A Markov chain describes a temporal stochastic process, where the current state is only dependent on the previous state (first-order Markov chain), and not on all the states before it [5]. The HMM consists of a Markov chain, but it cannot be observed directly, so the states in the chain are called hidden states. Observables are used to identify the underlying states [6].

The information within the xAPI statement is used to discover the observables (player's activity). The HMM is capable to infer the state based on the observables only in case if the observable is not misleading. To explain this let us consider an example wherein a person is wearing casuals implying that the person is in a hot environment. But what if the condition is opposite and the environment is cold, this would lead to an inaccurate assumption. In the same way, an activity (observation) which does not suit to the given state, the HMM would end up making incorrect inference about the current state. Now the challenge is how to gain the knowledge of the unknown state, simple, by requesting the information regarding the state. If acquiring information of the state is non-viable then having multiple observables is adequate to comprehend the accurate state. To elaborate on this approach lets extend the condition from the former example by adding other elements in the scenario like multiple persons in the background wearing winter apparels and also there is snow in the background. This solution acts like a Kalman filter [7] empowering the comprehension capability to derive state. Once the system knows the current state, using the HMM it can discover the activity with the highest emission probability at a given state and match it with the player's activity. If there is no match then the player's activity is treated as a misfit in that given state. The most feasible activity is flagged to the given state by the compass to be later used for the purpose of assistance whenever the player approaches the given state next time. There is an open discussion at our end related to the data utilization of incorrect activities.

2.3 Usage of Markov Chain

Using the Markov chain property it is possible to predict the future state based on current state [5]. The Markov chain predicts the future state from the current state, and if there is an activity flagged to the upcoming future state then the compass notifies the SG in the current state in advance so that the appropriate customization measures can be conducted by the SG.

3 Application and Result

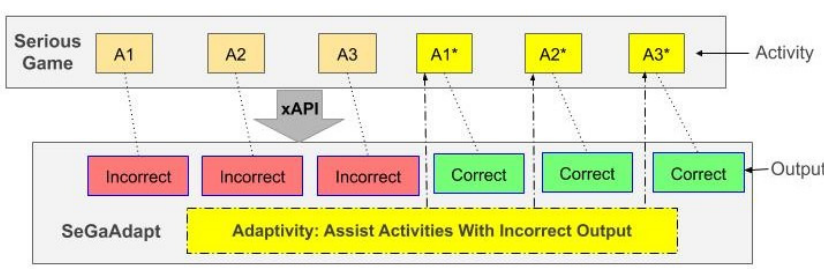

Fig. 3-1. Activity Customization of Serious Game by SeGaAdapt

Fig. 3-1, depicts the implementation of the SG with SeGaAdapt tracking the activities with incorrect output and providing assistance related to the activity. SeGaAdapt constantly receives xAPI statements from SG whenever an activity is performed. In Fig. 3-1 activities A_i non-assisted while A_i^* are provided assistance.

3.1 Command Line Tool Integrated With Framework Prototype

We have developed a command line tool that mimics activities inside the Lost Earth Serious Game (LESG). We integrated it within the framework prototype for technical verification. We considered only two aspects from LESG: (1) time of day (states - no observable) and (2) sensor (observables).

Time of Day: Denotes the current time of day within the game (changes after each round); possible values are dawn, day, dusk and night.

Sensors: Imagery sensors are one of the game's producible artifacts which are deployed to collect image data. Any sensor can be deployed at any given time of day, but the type of sensor must be carefully selected due to atmospheric variations and deployment costs. The 3 available sensor types are (1) electro-optical (EO), cheapest but efficient only during the day; (2) radar (SAR), costly and efficient in all times of day unless distorted; (3) infrared (IR), costly and efficient in all times of day.

HMM: The HMM is $\lambda = (A, B, \pi)$. A is transition probability (check Table 1) and B is emission probabilities (check Table 2). The initial probability π has 25% probability for each time of day.

Table 1. A: Transition Probability

	Dawn	Day	Dusk	Night
Dawn	0.0	1.0	0.0	0.0
Day	0.0	0.0	1.0	0.0
Dusk	0.0	0.0	0.0	1.0
Night	1.0	0.0	0.0	0.0

Table 2. B: Emission Probability

	EO	IR	SAR
Dawn	0.2	0.5	0.3
Day	0.7	0.2	0.1
Dusk	0.15	0.4	0.45
Night	0.1	0.4	0.5

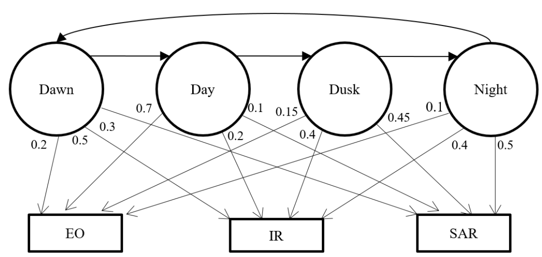

Fig. 3-2. HMM Probabilities from Table 1 & 2

3.2 Adaptivity Manifestation

When the game starts, the information about the initial state is sent to the framework. Using the initial state and the transition probability from Table 1, the system keeps track of the current state and is able to predict the future state using Markov chain, and meanwhile the HMM checks for the incorrect activity. Two adaptivity manifestions have been implemented for the prototype:

Assistance using hints - Instance 1: While performing the sensor deployment activity in round x, the EO sensor is deployed during dusk. As per the emission probabilities from Table 2, EO sensor is not feasible for dusk and the most feasible one is IR sensor, thus IR sensor is flagged to 'day'. Using transition probabilities (Table 1), a deduction can be made that the round $x+3$ will be day and the round $x+4$ will be dusk. Upon reaching the round $x+3$ the compass checks for the flagged sensor to the upcoming state i.e., day and sends message to SG. The SG displays message stating "IR sensor would be more feasible for dusk". This gives the chance to the player to deploy IR sensor in the next round.

Assistance using interactive elements - Instance 2: The SG and framework behave exactly same as Instance 1 with one difference: the SG locks the non-feasible game element, hence no chance of its utilization. The player used the IR sensor during day. As per the emission probability from Table 2 the EO sensor is much more feasible for day, therefore during day all sensor types except EO are being disabled and enabled again only after completion of a "day"-round.

4 Conclusion and Future Work

We were able to implement the HMM-based assistance in a prototype that mimics an activity engagement use case from our serious game. The results from a real use case like scenario provided the desired result and expectation of an empirical assistance system as a contrivance for the smart user assistance. HMM was able to identify the misfit activity at a given state in the game and point out the feasible activity which allowed to assist when the game reached the same given state next time where a mistake was made. Feedback from the framework provided the chance for the player to be well prepared in advance and gather resources or right tool in order to perform the upcoming activity more efficiently. In order to implement the assistance mechanism using SeGaAdapt very few alterations are required into the pre-existing structure of the SG, given that the SG follows the xAPI specification. The development of the SeGaAdapt system along with its integration in operational serious games is ongoing.

References

[1] Streicher, A., Smeddinck, J.D.: Personalized and adaptive serious games. In: Entertainment Computing and Serious Games, pp. 332–377. Springer (2016)

[2] ADL.net: xAPI Specification 1.0.1, https://github.com/adlnet/xAPI-Spec/blob/a752217060b83a2e15dfab69f8c257cd86a888e6/xAPI.md

[3] Bunian, S., Canossa, A., Colvin, R., El-Nasr, M.S.: Modeling individual differences in game behavior using hmm. In: Thirteenth Artificial Intelligence and Interactive Digital Entertainment Conference (2017)

[4] Derbali, L., Ghali, R., Frasson, C.: Assessing motivational strategies in serious games using hidden markov models. In: The Twenty-Sixth International FLAIRS Conference (2013)

[5] Gagniuc, P.A.: Markov chains: from theory to implementation and experimentation. John Wiley & Sons (2017)

[6] Rabiner, L., Juang, B.: An introduction to hidden markov models. IEEE ASSP magazine **3**(1), 4–16 (1986)

[7] Musoff, H., Zarchan, P.: Fundamentals of Kalman filtering: a practical approach. American Institute of Aeronautics and Astronautics (2009)