Proc. of the Third International Conference on Entertainment Computing (ICEC 2004), Sep. 2004 Eindhoven, The Netherlands, published in Lecture Notes in Computer Science, Matthias Rauterberg (Ed.), vol. 3166, pp. 429-434.

MMOG Player Classification Using Hidden Markov Models

Yoshitaka Matsumoto and Ruck Thawonmas[1]

Intelligent Computer Entertainment Lab., Department of Human and Computer Intelligence, Ritsumeikan University, Kusatsu, Shiga 525-8577, Japan
{matsumoto, ruck}@ice.ci.ritsumei.ac.jp
http://www.ice.ci.ritsumei.ac.jp/

Abstract. In this paper, we describe our work on classification of players in Massively Multiplayer Online Games using Hidden Markov Models based on player action sequences. In our previous work, we have discussed a classification approach using a variant of Memory Based Reasoning based on player action frequencies. That approach, however, does not exploit time structures hidden in action sequences of the players. The experimental results given in this paper show that Hidden Markov Models have higher recognition performance than our previous approach, especially for classification of players of different types but having similar action frequencies.

1 Introduction

The market size of Massively Multiplayer Online Games (MMOGs) continues to grow at a high speed. At the same time, competitions among MMOGs are also becoming very high. To keep players in the games, it is very important that the players' demands are grasped and the players are provided appropriate contents tailored for each player or each specific group of players. This kind of customer relationship management (CRM) [1] for MMOGs is inevitable.

In virtual worlds such as MMOGs, players are typically identified by their characteristics as "killer", "achiever", "explorer", and "socialiser" [2]. Killers just want to kill other players and monsters. Achievers entertain themselves by attempting all possible actions to grow their avatar characters up. Explorers roam around the game world to discover unknown things. Socialisers want to build and maintain social relations with other players. Following this categorization, a typical implementation of CRM for MMOGs can be depicted in Fig. 1. In this figure, players are categorized into pre-defined types based on appropriate selected features from the game logs, and are provided contents according to their favorites. Thereby, the players should enjoy the games more and hence play longer.

---

[1] The author has been supported in part by the Ritsumeikan University's ***Kyoto Art and Entertainment Innovation Research***, a project of the 21st Century Center of Excellence Program funded by the Japan Society for Promotion of Science.

M. Rauterberg (Ed.): ICEC 2004, LNCS 3166, pp. 429–434, 2004.

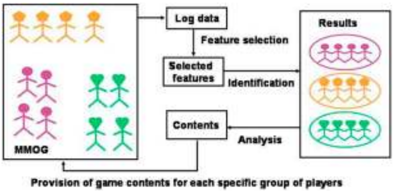

Fig. 1. Typical implementation of CRM for MMOGs

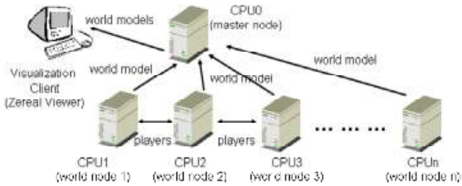

Fig. 2. Architecture of the MMOG simulator

We have reported in [3] a classification approach using Adaptive Memory Based Reasoning (AMBR), a variant of Memory Based Reasoning (MBR), based on the frequencies of player actions. That approach, however, does not exploit time structures hidden in action sequences of the players. Thus it is not suitable for classification of players of different types but having similar action frequencies.

In this paper, we propose an approach using Hidden Markov Models (HMMs) that mines time structures hidden in the action sequences. HMMs have been widely and successfully applied to a large number of problems, such as speech recognition [4], DNA and protein modeling [5], and gesture recognition [6]. We show in this paper that HMMs are also effective in classification of MMOG players, and, in particular, have higher recognition performance than AMBR based on action frequencies.

2 MMOG Simulator and Player Modeling

To acquire game logs in MMOGs, we use the version[2] of Zereal [7] released to us on June 14, 2003. Zereal is a Python-based multiple agent simulation system running on a PC cluster system. The architecture of Zereal and a screen shot of a game world are shown in Figs. 2 and 3, respectively.

In this study, we focus on two types of player agents, Killer and MarkovKiller, provided with Zereal, because they somehow behave like "killer" and "achiever", respectively. Each player agent has nine actions, i.e., 'walk', 'attack', 'chat', 'pickuppotion', 'pickupfood', 'pickupkey', 'leaveworld', 'enterworld', and 'removed'. The action 'removed' is outputted to the game logs when a player agent (or a monster) dies due to its hit point having reached zero. Killer and MarkovKiller have different characteristics as described below.

- **Killer (K)** has no sociability and will pursue the closest player or monster and kill it.
- **MarkovKiller (MK)** selects the next action from multiple candidates using a given Markov matrix. By manipulating the Markov matrix, we implement two types of MarkovKiller, InexperiencedMarkovKiller and ExperiencedMarkovKiller, described as follows:
- **InexperiencedMarkovKiller (IMK)** who equally attempts all possible actions in a given situation; all elements of the Markov matrix equal.

---

[2] This version of Zereal is different from the version that we used in our earlier reports in [1][3][8].

- **ExperiencedMarkovKiller (EMK)** who prefers particular actions over others in a given situation; the elements of the Markov matrix are not uniform.

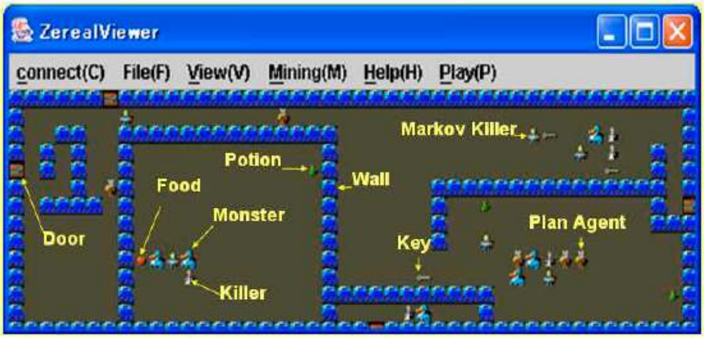

Fig. 3. Screenshot of a game world

3 Hidden Markov Models

HMMs [4] are a tool to statistically model a process that varies in time. From the set of observations, time structures hidden in the data are derived. An HMM can be specified by (1) the set of the possible hidden states $S = \{s_1,\ldots,s_N\}$; (2) the transition matrix A whose elements a_{ij} represent the probability to go from state s_i to state s_j; (3) the set of the observation symbols $V = \{v_1,\ldots,v_M\}$; (4) the emission matrix B whose elements b_{jk} indicate the probability of emission of symbol v_k when the system state is s_j; (5) the set of initial state probability distribution $\Pi = \{\pi_1,\ldots, \pi_N\}$ whose elements π_i represent the probability for s_i to be the initial state. For convenience, we denote an HMM as a compact notation $\lambda = (A, B, \Pi)$.

Table 1. Initial emission matrix having 8 states and 9 symbols (w: walk, a: attack, c: chat, p: pickuppotion, f: pickupfood, k: pickupkey, e: enterworld, l: leaveworld, r: removed)

	w	a	c	p	f	k	e	l	r
fight	0.00	0.75	0.00	0.00	0.00	0.00	0.00	0.00	0.25
talk	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00
hunt	0.70	0.00	0.00	0.10	0.10	0.10	0.00	0.00	0.00
transit	0.75	0.00	0.00	0.00	0.00	0.25	0.00	0.00	0.00
go for powerup	0.00	0.00	0.00	0.50	0.50	0.00	0.00	0.00	0.00
flee	0.75	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.25
bored	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
transported	0.00	0.00	0.00	0.00	0.00	0.00	0.50	0.50	0.00

The Baum-Welch algorithm was used to train HMMs, one for each player type, using the training data set. The training data set consists of both the action sequence and the type of each known agent player. To identify the type of an unknown player agent, the Viterbi algorithm was used. This algorithm computes the log probabilities using the trained HMMs, inputted by the action sequence of the unknown player agent. The unknown player agent will be assigned the type of the HMM with the highest log probability.

Table 2. Average appearance frequency of each action, and the average length of the action sequences for each agent type

	w	a	c	p	f	k	e	l	r	length
K	157.79	29.08	0.00	0.46	1.96	0.78	0.12	0.12	0.06	190
IMK	223.69	2.52	22.09	1.96	1.91	0.33	0.11	0.11	0.01	253
EMK	219.27	6.62	19.83	4.31	4.09	0.40	0.10	0.10	0.00	255

The performance of HMMs is in general dependent on the model structure and the initial parameters of λ. Here, we based on the model of MarkovKiller in Zereal. Namely, each HMM was constructed by 8 states ($N = 8$) and 9 symbols ($M = 9$), and the initial value of each element in B was shown in Table 1. An equivalent probability, i.e., $1/N = 0.125$, was assigned to each element of Π and A.

4 Results

In our experiments, we generated game logs by running 10 independent Zereal games with 300 simulation-time cycles. Each game consisted of 16 worlds, in each of which there were 50 player agents of each type, 50 monsters, and 50 items for each game object (food, potion, and key). The total number of each agent per game was thus 800. Next we transformed these raw game logs into sequences of actions for being used by HMMs. For performance comparisons, the game logs were also preprocessed by the feature selection algorithm originally proposed in [8] for being used by AMBR.

Table 2 shows the average appearance frequency of each action, and the average length of the action sequences for each agent type. It should be noted that InexperiencedMarkovKiller and ExperiencedMarkovKiller are relatively similar to one another, in terms of action frequencies, which would cause low recognition rates for any classifier that uses only this type of information.

We conducted experiments on hierarchical classification of players. First, upper-level classification was performed. Namely, we classified all agents into two agent types, Killer and MarkovKiller. Second, we conducted lower-level classification. MarkovKiller-type agents were classified into InexperiencedMarkovKiller and ExperiencedMarkovKiller. To reliably measure the recognition rates of the two classifiers, we used the leave-one-out method discussed in [9].

4.1 Upper-Level Classification

Figures 4 and 5 show the recognition rates of both HMMs and AMBR for classification of the player agents into Killer and MarkovKiller, respectively. Both classifiers give high recognition rates though HMMs slightly outperform its counter part for all games.

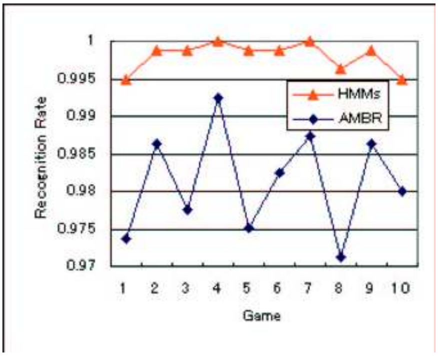

Fig. 4. Recognition rates for Killer

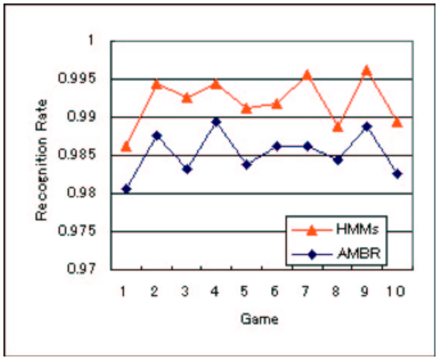

Fig. 5. Recognition rates for MarkovKiller

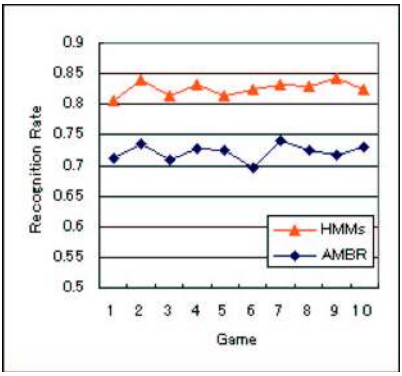

Fig. 6. Recognition rates for Inexperienced MarkovKiller

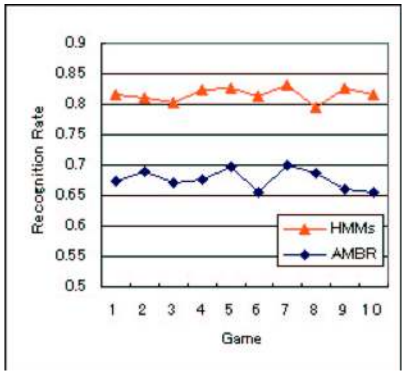

Fig. 7. Recognition rates for Experienced MarkovKiller

4.2 Lower-Level Classification

Figures 6 and 7 show the recognition rates of both HMMs and AMBR for classification of the MarkovKiller-type agents into InexperiencedMarkovKiller and ExperiencedMarkovKiller, respectively. For this task, the performance of HMMs is significantly superior to that of AMBR.

5 Conclusions

In this paper, we focused on time structures hidden in players' game logs of MMOGs, and proposed an effective approach for player classification using Hidden Markov Models. From the experiments, the recognition performance of HMMs is superior to that of the previously proposed AMBR based on the frequencies of player actions. As our future work, we will be researching on automatic generation of the optimal model structure of HMMs and the initial parameters. Moreover, we will be developing feature extraction techniques that preserve time structures. Eventually, we plan to apply our findings to real MMOG data.

References

1. Ho, J.Y., Matsumoto, Y. and Thawonmas, R., "MMOG Player Identification: A Step toward CRM of MMOGs", Proc. the 6th Pacific Rim International Workshop on Multi-Agents (PRIMA2003), Seoul, Korea, pp. 81-92, Nov. 2003.
2. Bartle, R., "Hearts, Clubs, Diamonds, Spades: Players Who Suit MUDs", The Journal of Virtual Environments, 1(1), May. 1996.
3. Ho, J.Y., and Thawonmas, R., "Episode Detection with Vector Space model in Agent Behavior Sequences of MMOGs", Proc. Future Business Technology Conference 2004 (FUBUTEC'2004), Fontainebleau, France, pp. 47-54, Mar. 2004.
4. Rabiner L.R., "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition", Proceedings of the IEEE, Vol. 77, No. 2, pp. 257-286, Feb. 1989.
5. Hughey, R., Krogh, A., "Hidden Markov Model for sequence analysis: extension and analysis of the basis method", Comp. Appl. in the Bioscience, 12, pp. 95-107, 1996.
6. Eickeler, S., Kosmala, A., Rigoll, G., "Hidden Markov Model based online gesture recognition", Proc. Int. Conf. on Pattern Recognition (ICPR), pp. 1755-1757, 1998.
7. Tveit, A., Rein, O., Jorgen, V.I., and Matskin, M., "Scalable Agent-Based Simulation of Players in Massively Multiplayer Online Games", Proc. the 8th Scandinavian Conference on Artificial Intelligence (SCAI2003), Bergen, Norway, Nov. 2003.
8. Thawonmas, R., Ho, J.Y., and Matsumoto, Y., "Identification of Player Types in Massively Multiplayer Online Games", Proc. the 34th Annual conference of International Simulation And Gaming Association (ISAGA2003), Chiba, Japan, pp. 893-900, Aug. 2003.
9. Weiss, S.M. and Kulikowski, C.A., Computer Systems That Learn, Morgan Kaufmann Publishers, San Mateo, CA, 1991.