Detection of MMORPG Bots Based on Behavior Analysis*

Ruck Thawonmas
Intelligent Computer Entertainment Lab
ISE, Ritsumeikan University
Kusatsu 525-8577, Japan
ruck@ci.ritsumei.ac.jp

Yoshitaka Kashifuji
Intelligent Computer Entertainment Lab
ISE, Ritsumeikan University
Kusatsu 525-8577, Japan
ci003053@is.ritsumei.ac.jp

Kuan-Ta Chen
Multimedia Networking and Systems Lab
IIS, Academia Sinica
Taipei 115, Taiwan
ktchen@iis.sinica.edu.tw

ABSTRACT

Game bots, i.e., autoplaying game clients, are currently causing troubles to both game publishers and bona fide players of Massively Multiplayer Online Role-Playing Games (MMORPGs). Use of game bots leads to collapse of game balance, decrease of player satisfaction, and even retirement from game. To prevent this, in-game polices, played by actual human players or game masters, often roam around game zones and individually question suspicious players, which is obviously laborious and ineffective task. In contrast to other work on automatic detection of MMORPG game bots based on the window events such as keystrokes, the game traffic, and the CAPTCHA test, our research focuses on log typically recorded by game publishers for database rollback. In particular, our research is based on discrepancies in action frequencies and action types in the log between human and bot characters. We propose the bot-detection methodology consisting of two stages. In the first stage an unknown character will be classified as "bot" if its frequencies of particular actions are much higher than those of known human characters. In the second stage, the rest of characters will be classified by the support vector machine classifier based on their action types. We evaluate the proposed methodology using game log of a Korean MMORPG titled Cabal Online and confirm its effectiveness.

Keywords

Game Bot, Online Game, Action Frequency, Action Type

1. INTRODUCTION

Nowadays online gaming has become an important subculture among Internet users. Also, it has been a profitable Internet business because online games are so attractive and addictive that numerous players are willing to pay for a ticket for venturing in the game's virtual world. Most users play games for purposes such as killing time, participating the game narration, or enjoying the ambience created by group sharing and collaboration. However, at the same time, some users like to achieve their in-game goals, such as gathering certain amount of wealth or possessing certain powers, without spending as much effort as other players. These users would turn to "cheat" by using game bots to acquire the desired benefits.

Game bots are automated programs with or without artificial intelligence that help players enhance, accelerate, or bypass some routines in the game. For example, in Massively Multiplayer Online Role-Playing Games (MMORPGs), players can save a great deal of time by using bots to perform repetitive tasks, such as slashing low-level monsters, or fishing in a river to master the fishing skills. In this way, players can acquire great power and gather a considerable deal of wealth in a relatively short time by the help of game bots. This may erode the balance of power and economics in the game world and therefore make bona fide players feel frustrated and even stop playing the game.

Due to the above reasons, the gaming community generally disapproves of the use of game bots, as bot users obtain rewards with disproportionate efforts. However, game bots are hard to detect because they are designed to simulate human game playing behavior and they follow game rules exactly. Therefore, the use of game bots generally cannot be prevented or detected by the techniques designed for fighting game cheating. To cope with the increasingly use of game bots, researchers have proposed various techniques for preventing [1, 2] with CAPTCHA or detecting game bots at the game level based on movement patterns [3, 4, 5, 6], operating system level based on window events [7], and traffic level [8], but none of them have been proved perfect. One of the primary reasons is that bot developers may employ countermeasures to evade the bot detection methods, and thus bot detection has become an indefinite war between researchers and bot developers, just like the war between antivirus companies and virus writers. Due to this reason, we believe that bot detection algorithms that rely on more fundamental behavior of game bots will be more robust against bot developers' counterattacks, as changing such behavior may make bots less competent in fulfilling users' needs efficiently.

*This work was supported in part by Japan Society for Promotion of Science (JSPS) under the Grant-in-Aid for Scientific Research (C) 20500146 as well as by Taiwan Information Security Center (TWISC), National Science Council under the grants NSC 97-2219-E-001-001 and NSC 97-2219-E-011-006.

Figure 1: Screenshot of Cabal Online

In this paper, we deal with MMORPG bots with a completely different perspective—their resource gathering and trading behavior. Our reasons are that, first, bot detection based on event timing, as done in [7], can easily be evaded by a random arrangement of the trigger time of bot's actions. Also, detection schemes based on characters' movement behavior, as done in [3, 4], is less effective in MMORPGs than in FPS games, because players control the exact movement of characters, i.e., "how" to move, in a FPS game, while players control only the moving destination of characters, i.e., "where" to move, in MMORPGs. At the same time, as resource accumulation is often one of the most important goals to achieve in an MMORPG, the game bots will unavoidably collect loots and trade them with non-player characters or other player characters, which we believe can be a basis to distinguish game bots and human players. To the best of our knowledge, our research is the first to detect MMORPG bots based on their resource accumulation and trading behavior.

The contribution of this paper is two-fold. 1) We propose to detect game bots based on their in-game behavior, especially those related to the designed purposes of bots, such as resource accumulation, which are less likely significantly changed by bot developers as a countermeasure to bot detection methods. 2) Through the real-life game log of Cabal Online, a Korean online game, we justify our proposal by classifying of the action log of characters which are invoked by game bots and human players respectively. The results show that our methodology yields a recognition rate above 0.91 with the minimum detection time of 15 minutes. As resource accumulation is a relatively fundamental behavior of MMORPG bots, we believe this approach has the potential to distinguish between human players and automated programs and thus merits further investigation.

2. DATA SET

The data set of Cabal Online (Fig. 1) in use was provided to us by Gamepot Inc, the local publisher in Japan. The data set consists of three days log of seven human players of the Blader character class and seven detected bots of the same class, where character levels are evenly distributed between humans and bots.

Table 1: The mean and standard deviation of action types by bots and humans

Character Type	MEAN	STD
Bot	32.71	8.99
Human	54.29	13.52

The information available in each line of the log includes

- character ID modified accordingly by Gamepot before given to us in order to prevent leak of players' personal information;
- action related to item trading, item usages, inventory item processing, warehouse processing, etc.; and
- time stamp indicating the time that the action was invoked.

This type of log is in general recorded by the game publisher for database rollback when data loss occurs. There are in total 92 action types in this log.

3. METHODOLOGY

According to information from Gamepot, most detected bots are those who used illegal macro tools in order to acquire particular skills, such as extraordinarily continuous use of skills or items, and the ability to enter prohibited zones. From this information, we speculate that for particular actions, their frequencies invoked by bots should be distinguishably higher than by human players. In addition, bots should concentrate on particular action types. We confirm these two speculations in Fig. 2 and Table 1, respectively.

Figure. 2 is the scatter plot of action frequencies; where the horizontal axis is the frequency of a selected action invoked by the human player who most frequently invoked that action, and the vertical axis is the frequency of the same action by a bot of interest. An action is selected for this figure if a bot of interest invoked this action with the frequency beyond

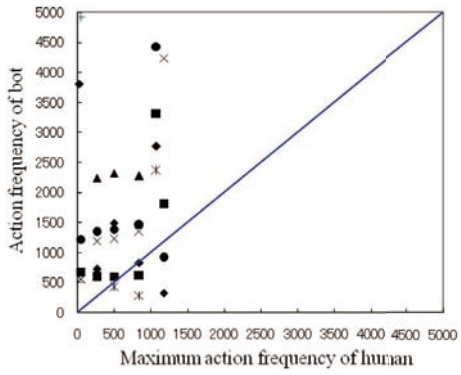

Figure 2: Scatter plot of action frequencies of bots and maximum action frequencies of humans

its mean of action frequencies. This figure well confirms our first speculation, with few exceptions.

Table 1 shows the mean and the standard deviation of the numbers of action types by bots and humans. It can be seen from this table that both statistics of bots are lower than those of humans, confirming well our second speculation. In addition, we further found that among 92 action types, there are 58 action types invoked by both humans and bots, 33 action types invoked only by humans, and 1 action type invoked by only bots.

The proposed methodology for detecting bots consists of two stages. The first stage classifies characters into "bot" or "pending" based on the first speculation, i.e., the information on discrepancy in action frequencies is exploited. For those classified as "pending", they are proceeded to the second stage. The second stage is based on our second speculation, and thus the information on discrepancy in action types is exploited, where the support vector machine classifier SVM [9] is applied to further classify those "pending" characters into "bot" or "human". Below we describe these two stages in detail, respectively.

In the first stage, first let $freq(x, y)$, $mean\_freq(y)$, and $max\_human\_freq(x)$ denote the frequency of action x by character y, the mean of action frequencies by character y, and the action frequency of the human player in the training data who most frequently invoked action x, respectively. A character of interest c is classified as "bot" if at least one action a exists with
$freq(a, c) > mean\_freq(c)$ and
$freq(a, c) > \rho \cdot max\_human\_freq(a)$ and
$max\_human\_freq(a) > 0$,
otherwise this character is classified as "pending". The only parameter here threshold ρ is determined from the training data as follows:

- For each bot b, select an action a as a candidate action if $freq(a, b) > mean\_freq(b)$ and $\frac{freq(a,b)}{max\_human\_freq(a)} \geq 1$ and $max\_human\_freq(a) > 0$;
  and, among all candidate actions, consider only action a^* whose ratio $\frac{freq(a^*,b)}{max\_human\_freq(a^*)}$ is minimal and add its ratio value as a new element into list L.

- If list L is not null, set ρ to the median of the elements in L; otherwise, set ρ to a very large value to ensure that the condition $freq(a, c) > \rho \cdot max\_human\_freq(a)$ is false for any character c.

In the second stage, for a "pending" character, its action vector is input to the SVM classifier, trained by the training data, to further classifier if the character is a bot or a human. Here, the action vector is defined as a vector of n binary elements, where n is the number of action types and thus $n = 92$ in our case, and an element of interest is 1 if the corresponding action was invoked at least once by this character, and is 0 otherwise.

4. EVALUATION

Table 2: The number of chunks of bots and humans for each detection time

Detection Time (mins)	Bot	Human
15	2513	1005
30	1285	579
45	874	419
60	663	339

In practice, it is preferable to detect bots within a short time. Henceforth, the detection time is used to refer to the time used to determine if a character of interest is a bot or a human. To evaluate the proposed methodology under different detection times, we partitioned the three-days log described in Sec. 2 of each human and bot into multiple chunks of a same interval, where a pair of consecutive chunks are overlapped with the ratio of 0.5. In our evaluation, each chunk was considered as an individual character, and four interval lengths were considered, i.e., 15, 30, 45, and 60 mins, each having the total numbers of chunks for humans and bots shown in Table 2.

In addition, in order to evaluate the detection performance for unknown characters not seen in the training data, the commonly used k-fold cross validation was employed, by which the recognition rate $= \frac{\#\text{correctly classified chunks}}{\#\text{tested chunks}}$, was derived. In each fold, the test data consist of all chunks of a pair of one human and one bot, and the train data consist of all chunks of the remaining six humans and six bots. Because there are 49 possible pairs of a human and a bot, k was set to 49. Each recognition rate discussed below is the mean of the recognition rates of 49 folds by a method of interest. A SVM tool called SVM-Light Support Vector Machine (svmlight.joachims.org) was used in the second stage, where the linear kernel was selected.

Figures 3,4,5, and 6 show the recognition rates of our methodology (AUTO), our methodology but with threshold ρ varied manually (MANUAL), and the SVM classifier only (only the second stage in our methodology was applied). From these figures, it can be seen that the proposed methodology outperforms the SVM classifier for all interval lengths of detection time. In addition, for the detection time of 30, 45, and 60 mins, its recognition rates are beyond 0.92 and comparable to the best recognition rate achieved by varying threshold ρ manually.

The contribution of the first stage in our methodology can be expressed by the recall and the precision ratios, where the former is defined as the ratio that a bot chunk is classified as "bot", and the latter is defined as the ratio that a chunk classified as "bot" is actually a bot chunk. Table 3 summarizes these two ratios for each detection time.

5. CONCLUSIONS

Unlike existing work, our research uses log typically recorded by game publishers and thus requires no addition cost for data acquisition. This fact indicates that our methodology is generalizable to other MMORPGS having game bots with similar behaviors to those in our work. Discrepancies from human characters in action frequencies and action types are

Table 3: The recall and precision ratios at the first stage for each detection time

Detection Time (mins)	Recall	Precision
15	0.24	0.95
30	0.33	0.95
45	0.43	0.95
60	0.43	0.94

utilized in the proposed methodology. The first stage of our methodology alone contributes to game-bot detection in average with 0.36 recall ratio and the precision of 0.95. In addition, its only parameter, the threshold ρ, is determined automatically from the training data, and its value leads to comparable performance with the best value determined manually for most interval lengths of detection time. The information on discrepancies in action types is exploited in the second stage as the input vector to the SVM classifier. The total recognition rate of the methodology, combining both stages, is superior to the SVM classifier alone for all cases attempted. As our future research, we plan to investigate possible, if any, counter measures from bot developers any and make the proposed methodology robust to them.

Acknowledgments

The authors are much indebted to Hiroshi Kobayashi at Gamepot Inc, who helped us gather the game log.

6. REFERENCES

[1] P. Golle and N. Ducheneaut. Preventing bots from playing online games. Computers in Entertainment, 3(3):3–3, 2005.

[2] L. von Ahn, M. Blum, N. J. Hopper, and J. Langford. CAPTCHA: Using hard AI problems for security. In Proceedings of Eurocrypt, pages 294–311, 2003.

[3] C. Thurau, C. Bauckhage, and G. Sagerer. Learning human-like movement behavior for computer games. In In Proc. 8th Int. Conf. on the Simulation of Adaptive Behavior (SAB'04), pages 315–323. IEEE Press, 2004.

[4] C. Thurau and C. Bauckhage. Tactical waypoint maps: Towards imitating tactics in fps games. In M. Merabti, N. Lee, and M. Overmars, editors, Proc. 3rd International Game Design and Technology Workshop and Conference (GDTW'05), pages 140–144, 2005.

[5] K.-T. Chen, A. Liao, H.-K. K. Pao, H.-H. Chu. Game Bot Detection Based on Avatar Trajetory. In Proceedings of IFIP ICEC 2008, 2008.

[6] K.-T. Chen, H.-K. K. Pao, and H.-C. Chang. Game Bot Identification based on Manifold Learning. In Proceedings of ACM NetGames 2008, 2008.

[7] H. Kim, S. Hong, and J. Kim. Detection of auto programs for MMORPGs. In Proceedings of AI 2005: Advances in Artificial Intelligence, pages 1281–1284, 2005.

[8] K.-T. Chen, J.-W. Jiang, P. Huang, H.-H. Chu, C.-L. Lei, and W.-C. Chen. Identifying MMORPG bots: A traffic analysis approach. In Proceedings of ACM SIGCHI ACE'06, Los Angeles, USA, Jun 2006.

[9] B. Schölkopf and A.J. Smola. Learning with Kernels. MIT Press, Cambridge, MA, 2002.

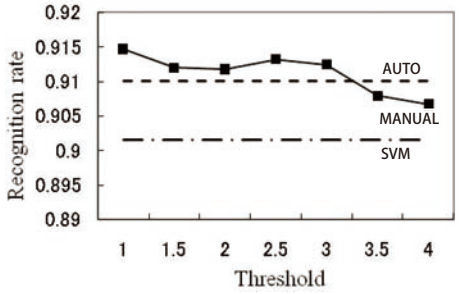

Figure 3: Comparison of recognition rates for the detection time of 15 min

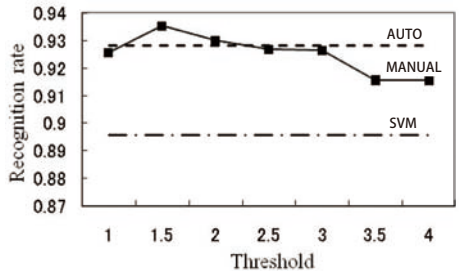

Figure 4: Comparison of recognition rates for the detection time of 30 min

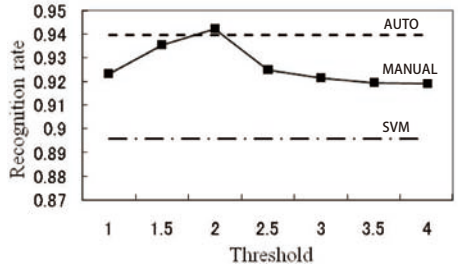

Figure 5: Comparison of recognition rates for the detection time of 45 min

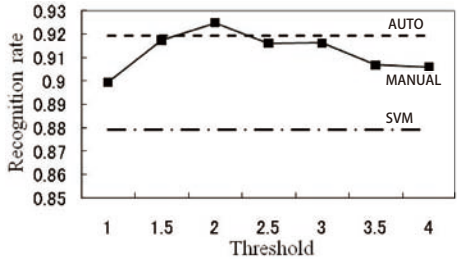

Figure 6: Comparison of recognition rates for the detection time of 60 min