Detecting the bank character in MMORPGs by analysis of a clustered network

Eunjo Lee[1] and Jina Lee[2] and Janghwan Kim[3]
[1] Data Information Center, ncsoft
Seoul, Republic of Korea
[e-mail:gimmesilver@ncsoft.com]
[2] Data Information Center, ncsoft
Seoul, Republic of Korea
[e-mail:jinalee@ncsoft.com]
[3] Data Information Center, ncsoft
Seoul, Republic of Korea
[e-mail:janghwan@ncsoft.com]

Abstract

As automatic programs in Massive Multiplayer Online Role Playing Games (MMORPGs) grow rapidly, detecting and blocking the automatic program, namely bots, is an important issue to improve the game user satisfaction. There are various types of bots in MMORPGs: gold farmer, bank character, and broker character. In this paper, we suggest the algorithms of detecting bots with network clustering. By applying the algorithms to a real game, we show that our method achieves practical performance.

Keywords: network clustering, network structure, MMORPGs, bot, bank character

1. Introduction

With increasing popularity in Massive Multiplayer Online Role-Playing Games (MMORPGs), automatic bot programs has become a more critical issue. A bot is a program with artifical intelligence designed to automate certain activities in the game. Some bots are used by general players for their personal greed like leveling up their game characters easily. However many bots are operated by professional organizations with the purpose of gathering game money and items and monetizing them into real currency. This professional group is called "bot nest".

Every action caused by bots violates the agreement betweeen the service comapny and the user. Companies try to reduce these malicious play patterns by giving strong restrictions to the malicious characters that use game bots. In this paper we introduce the bot detection method, especially the bank character detection. The proposed method was applied in Aion, one of the popular games from NCsoft, a major online game company in Korea.

2. Related Work

There are mainly two types of methods in identifying bots: detecting the process patterns of automated programs at client side, and analyzing the action logs at server-side. The former method provides a high accuracy rate, but suffers in detecting patterns of new programs. A large bot nest may run an undisclosed program of their own, which leads to late response in recognizing and identifying the yet unknown pattern.

To overcome this weak point, the server-side method identifies the difference of actions between human players and bot characters. For example, Thawonmas et al. and Kesteren et al. suggested the method that is based on discrepancies in action frequencies and action types[1][2], Varvello and Voelker focused on features of social connection [3], and Ahmad et al. and Mitterhofer et al. suggested to find repeating patterns that is nearly impossible to emerge from human activities [4][5].

While the previous approaches work well in detection of gold farmers who repeatedly accumulate resources in the game, they have limitation in detecting the character that collects money from these bots, called the bank character, which takes the fundamental role in bot nest.

In this paper, we propose a detection of detecting the bank character who collects game money and items from gold farmer characters. The proposed method will efficiently redeem the weak points of the previous methods and discourage the act of bot nest.

3. Bot identification schemes

3.1 Bot Patterns

Unlike bots used by individual players, the bots from bot nest have distinct roles and work in cooperation. First there are dozens of gold farmer bots, who collects game money and items from the game environment. The obtained goods are then delivered to another character who mainly exchanges it into game money. The game money flows into the bank character, and is distributed to the sellers, who sell it for real money to individual players. This is the mechanism how the bot nest makes profit. We also note that there are some broker characters, who buy game money from bot nests and resell it to individuals as depicted in **Fig. 1**.

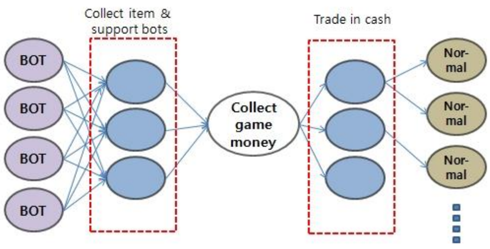

Fig. 1. Trade pattern of bot nest

With this structure, we can identify the bank character by tracking the trade logs of pre-detected bots. Since bot nest characters tend to trade game money between themselves actively, the network analysis on the characters based on the trade log is expected to yield a high detection rate.

3.2 Graph Structure

A graph is a structure with nodes and edges connecting between nodes. The graph is effective in analyzing relationships, so it is to analyze relationships in social networks. In this paper we interchangeably use the term of the graph and the network.

An edge may or may not have a direction from one node to the other, classifying the graph as directed or undirected. For example a friendship relation in real life could be modeled as an undirected graph, while the follower/following relation in Twitter is directed.

The graph can be classified according to their topological structure. A graph is said to be assortative mixing if nodes tend to be connected to other nodes with similar property. A disassortative mixing network is a network where nodes tend to be connected with dissimilar nodes. Most social networks, for example friendship network, are found to be assortative. Internet and neural networks are disassortative mixing [6].

Understanding the topology of the network is crucial in classifying nodes with appropriate algorithm without examining each nodes respectively.

3.3 Graph Clustering

Graph clustering aims to comprise as many as possible within-cluster edges and as few as possible between-cluster edges. Clustering increases the ratio of the number of edges within-cluster over the total edges [7]. This ratio is calculated as shown in Equation (1) and (2).

$$m = \frac{1}{2}\sum_{ij} A_{ij} \qquad (1)$$

$$\frac{\sum_{ij} A_{ij}\delta(c_i, c_j)}{\sum_{ij} A_{ij}} = \frac{1}{2m}\sum_{ij} A_{ij}\delta(c_i, c_j) \qquad (2)$$

Where m is the total number of edges, A_{ij} is the weight between node i and j, $\delta(c_i, c_j)$ returns 1 if node i and j are in the same cluster.

Equation (2) alone leads to conclusion that it is best to enclose the entire graph in a single cluster. A similar condtradictory phenomenon happens on decision trees where placing one record in a single node makes the best classification result. To solve this problem, we adopt the degree of randomness, called a random-walk betweenness in [8].

From a graph structure generated by laying out the edges randomly following the vertex degree, we compute 'the expected rate of the number of within-cluster edges over the total number of edges'. Subtracting this value from the original graph, we get the following Equation (4)

$$. \, k_i = \sum_j A_{ij} \tag{3}$$

$$Q = \frac{1}{2m} \sum_{ij} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j) \tag{4}$$

To increase Q, the graph needs to have many edges inside the cluster and less outside compared to the randomly generated graph. The higher value of Q implies that the graph is more concentrated inside the clusters. The Q value close to 1, the maximum value, indicates that the graph has a strong community structure, while the values for the networks with strong community usually fall in the range between 0.3 and 0.7 in practice [7].

In this paper we used the random-walk betweenness to analyze the topological structure of Aion trade graph and to evaluate the clustering algorithm.

4. Proposed Bot Detection Strategies

4.1 Properties of Trade Graph

In Aion there are many types of trades between the characters, but here we only focus on the actions that mainly happen in the bot nest and generated the trade graph with it. Both directed and undirected graph are used for the analysis, and we used the number of trades between the characters (nodes) as weight of the edge. The amount of the game money or the number of exchanged items was excluded in calculating the weight since it is difficult to compare them fairly.

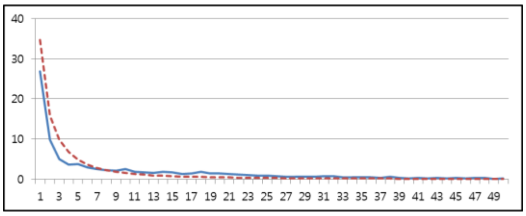

Fig. 2. Distribution of trades, solid line corresponds to bots and dot line to normal characters

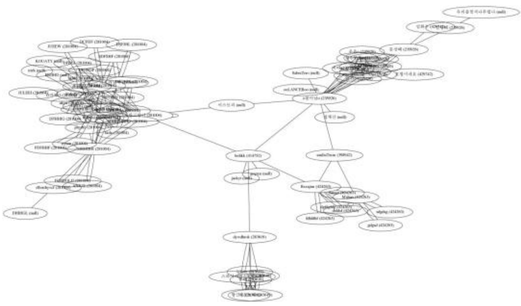

Fig. 3. Trade graph of Aion

Fig. 2 shows an example of the trade network where we can observe the Pareto principle of a scale-free network [9]. In **Fig. 2**, x-axis represents the number of trades, and y-axis is the ratio of characters.

We can see that while most players trade once or twice a week, only a few players have more than thousands of trades. The number of characters dealing with and the distribution of the average number of trades also show that the trade network in Aion follows the Pareto principle.

On the other hand, the trade network has an extreme hub node. The existence of hub node is the property of the scale-free network. The extreme hub node is the node that connects lots of characters but has low closeness. It has a 'weak link'. In other words, the extreme hub node has large number of edges but the weight of edge is low.

The distributions of bot characters are not far off from this principle either. However, bots have narrower ranges in number of transactions and trade connections compared to those of normal players. The average weight of bots is also higher, and extreme hub-nodes are hardly found.

The scale-free network has a weak tie property which means that some nodes strongly connected with each other generate a community and there is a weak link among different communities. The Aion trade network is also expected to have such property and some sample graphs are found to build communities with nodes that have heavy weighted edges as shown in **Fig. 3.**

Table 1. Q values due to the removing level of edges

w	Q	*W*	Q
1	0.69	6	0.81
2	0.89	7	0.78
3	0.89	8	0.76
4	0.86	9	0.74
5	0.83	10	0.72

The weak tie suggests that removing light weighted edges leads to clustering in some level. **Table 1** shows Q values of Equation (4) according to the level of light weighted edges to be removed.

However, we should consider the special characteristic of bot nest where the bank character collects game money and items from other bots. The bank character, which does not trade frequently with the same character, can have a light weighted edge. Removing light weighted edge globally can also remove bank characters. Thus we need to deal with this special case in the algorithm as well.

4.2 Clustering Algorithms

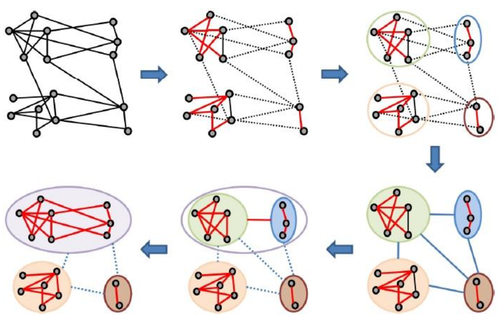

Fig. 4. Algorithms of graph clustering

a. Generate a trade graph from log data where a character is a node and a trade is an edge with the weight reflecting the number of trades between two nodes.

b. Build a cluster with nodes connected by edges with weights greater than or equal to internal weight of the cluster (node). The internal weight is the predefined boundary weight initially.

c. Generate a cluster graph where the average of the weights of edges inside the cluster represents the internal weight of the cluster, and the sum of the weights of the edges that connect nodes in a cluster and nodes in another cluster represents the weight of the edge between two clusters.

d. Merge two clusters if the weight of edge between clusters is higher than the internal weights of the two.

e. Repeat the steps c and step d until clustering becomes stable, or the internal weight of the cluster becomes lower than the initial boundary.

While most clustering algorithms deal with all nodes, the suggested algorithm filters out the nodes with light weights that have one or two trades. Since the purpose of the proposed system is to detect the bot nest, and not to cluster every node, this will suffice. The main targets of this algorithm are the characters of the bot nest that actively trade game money and items with each other.

The initial boundary weight is important because it affect the quality of the clustering. Q values from various boundary weights are shown in **Table 2**. We found that the clustering has the best performance when the boundary is set at 3. However, the further analysis with real data samples showed that with boundary weight of 3, the size of the cluster became too large and could contain characters that are actually not from bot nest. After several tweaking, we landed on the initial boundary weight of 5.

Table 2. Q values according to *w values*

w	Q	*w*	Q
1	0.69	6	0.81
2	0.88	7	0.79
3	0.89	8	0.76
4	0.86	9	0.75
5	0.83	10	0.73

The second step is to mark bot nests in clusters. We define a bot nest cluster as the cluster that has some pre-detected bot characters, which are detected at client-side.

Last point to be considered is the case of brokers. Brokers buy game money from bot nests, and resell it to normal players. This type of character is difficult to bind to one specific cluster. We design a rule for this. We search for a character that receives game money from various bot nest where suspected clusters found, and tag it as the broker character. The proposed system tags the character that receives many times from those clusters as the broker character and merges those clusters into one.

5. Implementation and Evaluation

We defined a bot nest cluster as the cluster that has one or more pre-detected bot characters and detected the bank characters with the proposed algorithm.

Our system has been applied 10 times with the log data from Oct. 22, 2010 to Dec. 30. The results are displayed in **Table 3**, showing the number of blocked account and the recovered game money.

As shown in the table, the amount of recovered game money per blocked bank character is significantly higher compared to the performance of the current bot detecting method of Aion (see BOT items part in **Table 3**).

Table 3. Blocked results by detecting system

	Number of accounts		Game money (unit :million)	
	BOT	Bank	BOT	Bank
1st	25,695	1383	737,000	1,600,000
2nd	6,322	446	10,200	333,400
3rd	1,217	119	21,600	162,700
4th	974	106	14,000	28,100
5th	3,763	171	348,700	81,100
6th	1,031	140	15,500	97,900
7th	1,053	137	12,700	167,400
8th	467	71	13,400	12,300
9th	13,742	512	807,800	870,000
10th	704	72	54,400	27,300
total	54,968	3,157	2,035,300	3,380,200

While the amount of recovered game money through the current system is 37 million Kinas (the unit of game money in Aion) per bot account, the new system recovered 1,070 million Kinas per account. This result implies that the analysis system has effectively given an economic hit to the bot nest by detecting the bank and the broker characters.

6. Future Work

The proposed method uses pre-detected bot characters to identify related characters, and hence it has the limitation of depending on the quality of the current bot pattern detection system in client side. To overcome this limitation, we need to come up with a novel method to detect bot characters.

We suggest that by 1) a direct method analyzing the behavior patterns of bots and 2) tracking down the bot characters from bank/broker characters through network analysis as a novel bot detection method. The latter is expected to yield a large pool of malicious accounts through the mutual reinforcement analysis between the bot character and related characters.

References

[1] R. Thawonmas, Y. Kashifuji, K.-T. Chen, Detection of MMORPG Bots Based on Behavior Analysis, *In Proc. of the International Conference on Advances in Computer Entertainment Technology*, 2008

[2] M. Kesteren, J. Langevoort, F. Grootjen, A step in the right direction: Bot Detection in MMORPGs using movement analysis, *In Proc. of the 21th Belgian-Dutch Conference on Artificial Intelligence*, 2009

[3] M. Varvello and G.M. Voelker, Second Life: a Social Network of Humans and Bots, In Proc. ACM NOSSDAV, 2010

[4] M. A. Ahmad, B. Keegan, J. Srivastava, D. Williams, N. Contractor, Mining for Gold Farmers: Automatic Detection of Deviant Players in MMORPGs, *In Proc. of IEEE Social Computing*, 2009

[5] S. Mitterhofer, C. Platzer, C. Kruegel, E. Kirda, Server-side Bot Detection in Massive Multiplayer Online Games, *IEEE Security and Privacy, 7(3): pp. 29-36*, 2009

[6] M.E.J. Newman, Mixing patterns in networks, *Phys. Rev. E 67, 026126*, 2003

[7] M.E.J. Newman and M. Girvan, Finding and evaluating community structure in networks. *Phys. Rev. E 69, 026113*, 2004

[8] M.E.J. Newman, Analysis of weighted networks, *Phys. Rev. E 70, 056131*, 2004

[9] Albert-Lâaszlâo Barabâasi, 『Linked: The New Science of Networks』, *Basic Books*, 2002