Predicting player's behaviour and departure in MMORPGs using World of Warcraft Avatar History Dataset

Shubh Vachher
Department of Computer Science
Tsinghua University
Beijing, China
vachhers11@mails.tsinghua.edu.cn

Davide Liu
Department of Computer Science
Tsinghua University
Beijing, China
liud20@mails.tsinghua.edu.cn

Abstract

MMORPGs (Massive Multiplayer Online Role Play Games)are a kind of videogames where each user controls a digital avatar and interacts with other online users in a digital world. The player's main task is the one of growing his own avatar by raising his level, enhancing his skills, improving his equipment, completing quests and so on. On the other side, also the social part of the game plays a fundamental role. Often, players are given the chance to form teams or alliances, talk, discuss, exchange items, cooperating to complete some quests or even competing in some kind of arenas (PvP maps) in both one vs one and team vs team matches. This makes it very useful for the MMORPG industry to model the behaviour of players: to predict their actions, study their behaviour as well as to predict their loss of interest in the game. In this project we will analyze the data of the avatars of World of Warcraft [8] and build some models in order to predict player' behaviours and their departure.

1 World of Warcraft dataset

The game World of Warcraft has been one of the most popular MMORPG with over 8.3 million active players since its release in 2003 by Blizzard [2]. The game is divided into two factions, the Horde and the Alliance. Each faction is divided into 5 races and 10 general classes, plus 229 zones or regions. However, the data collected for this study comprises only Horde players. The dataset, also known as wowah dataset [7], includes the records of 37,354 avatar sampled every 10 minutes during the year 2008. The data includes the avatars' game play time and a number of attributes, such as their race, class, guild, current level, and in-game visited locations. During the monitored period 10,826,734 sessions associated with the avatars were observed. Moreover, to protect players' privacy, the avatars' name and guild have been randomly mapped as positive integers with a consistent mapping (the same names were always mapped to the same integers).

2 MMORPG's Business model

Nowadays, MMORPGs are so popular that their market is worth over a billion dollars in 2019 and it is expected to keep growing during the following years [5]. They can count over 10 millions active monthly players, and, as expected, their huge base of players is one the keys of their success. In fact, these kind of games belong to the (FTP) Free-To-Play category, which means that everyone can play them for free, but it is still possible to pay some micro-transactions to get additional contents such as new skins for its own avatar, new items or pieces of equipment, exclusive maps or in-game virtual coins. These micro-transactions usually have a price that can range from one dollar to hundreds of

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

Query Time	Seq. #	Avatar ID	Guild	Level	Race	Class	Zone
01/01/06 23:59:39	1	467		1	Orc	Warrior	Orgrimmar
01/01/06 23:59:39	1	921	19	1	Orc	Shaman	Orgrimmar
01/02/06 00:03:31	45	1367	8	60	Undead	Warrior	Arashi Mountain

Figure 1: Example of some rows of the dataset.

dollars and are often the only source of revenue of these FTP games since the advertisements are often absent because they would severely impact on the user's experience. Despite their huge amount of players, only a small part of them regularly engage in these optional purchases, but it's still enough to maintain these games very profitable and successful. That's why having a lot of players is not enough but it is also important to keep them in the game longer, creating addiction, and convince them to make some in-game purchases so that the company of the game can keep running.

3 Importance of predicting players' departure

Considering the company's reliance on these in-game purchases, the cessation of play from part of its users will lead to a negative impact on the company's revenue. The business strategist Fred Reichheld [6] stated that within the financial services industry "A 5% increase in customer retention produces more than a 25% increase in profit". That's why it is very important to predict when players are about to leave the game (churn prediction) [1] and immediately take actions to remedy this problem dissuading them from leaving it. In this post we are going to use the term churn to refer to the process of a player leaving the game indefinitely and discontinuing to be a customer. Many players might be intentioned to leave a game for many different reasons such as lack of contents, monotonous game-play or game imbalances (too easy or excessively hard). Sometimes, player may just leave the game because they realized it was a waste of time or because they had more important things to do such important exams or work. A research conducted by SuperData [5] found that gamers tend to abandon games in groups, with 34% of churned players indicating that they had left a game simply because their friends stopped playing. Thus, player departure should indicate low user satisfaction and if we can predict it and find the problem, we may have a chance to stop them from leaving and make further improvements to keep the game interesting. Following, we have listed (in frequency order) the most frequent reasons for players leaving a game [3]:

- They had more important things to do, such as obligatory military service or school entrance exams;
- They become bored with it;
- Their friends left;
- They realized that it is waste of time playing MMORPG after all;
- Their accounts were hacked;
- They turned to other newer games;
- They had no more money to spend on entertainment.

4 Importance of predicting player's behaviour

Predicting players' behaviours can have many benefits on the development of MMORPGs, since it helps developers to create contents to satisfy each kind of players, that is, more maps to satisfy the explorers, more pvp arenas to allow the killers fight and so on. This task is known as player modeling and it helps game developers or game designers to create experiences with appeals specifically devoted to each behavior making sure that each class of players may feel equally satisfied [4].

5 Different behaviours in MMORPGs

In MMORPGs are globally recognized 4 different types of behaviours [2]:

- **Socializer**: This kind of gamers prefer to play online just for social pleasure, to interact with other players and to naturally evolve their character. They often crowd the game chat

Figure 2: Example of gameplay of World of Warcraft.

by talking about their personal life and use many emoticons. Finding a lot of friends is more important than complete hordes of missions for them, hence, they spend most of their time in neutral maps like the various cities located around the map and rarely deep into the wild lands;

- **Killer**: They prefer to develop their character, but not with the intention of obtaining merits. In constrast, they are interested in competing against other players or against NPC enemies that are more powerful and complex. They almost never talk and when they do it is only to write a bunch of specific words. Their natural habitat are the PvP arenas and they love using their powerful avatar to smash whoever has the misfortune of walking on their way;
- **Explorer**: They prefer to know the whole game environment, to discover secret areas, to find out 3D modeling errors or programming errors, easter eggs and know all the possible items, such as monsters and maps. They are not eager of evolving their avatar as fast as possible but they are willing of just acquiring experience by exploring the world following the natural process of their character's evolution;
- **Achiever**: They prefers to earn points, to evolve their character, to acquire equipment and other concrete measures of success in the game. Every action they performs is focused to improve their character's skills and for no reason in the world they would spend their time in tasks which may not make their avatar more powerful.

Following, we are going to describe a method to analyze the World of Warcraft dataset and predict the behaviour of each player classifying it into one of the four above mentioned classes: killer, socializer, achiever and explorer. This method can also be extended to other games, not necessarily MMORPG ones, simply by processing the data in the appropriate way.

6 Data preprocessing for behaviour prediction

Before working on the data, are required some pre-processing steps in order to prepare it and fitting it for our goals. First of all, our new dataset will have only one entry for each different player. This reduces the size of the new dataset down to 37,354 rows. Furthermore, the players whose total playing time was less than 2 hours have been filtered since they would probably consist in new players who just tried the game for a couple of hours before realizing the game itself wasn't suitable for them and then they quit from it. This process further reduced the numbers of total rows down to 15,740.
Of the original dataset, have been keept only the attributes ID (char in wowah dataset), class (charclass in wowah dataset), race and guild. Other attributes have been removed or combined to form new useful ones. In particular, the level attribute have been replaced with lvl_start and lvl_end which represent the initial and final level of the player respectively in the considered time frame. These two attributes have then been used to calculate the evolution, that is the number of level a character grew during the analyzed window of time, and the lvl_speed which is gives an idea of the speed of the increase of a player's level. The different timestamps have been used to calculate the player's playing time in hours, time_hours, and have been also counted the number of guilds a player joined

(n_guilds) and the number of visited maps (n_maps).
Moreover, have also been added a few attributes about players' location such as n_city and n_pvp which is the number of hours spent in zone of type city (neutral) and PvP such as arenas. Lastly, have been added the numbers four more features relative to the of hours spent in different areas divided by difficulty (map_beginner, map_low, map_medium and map_advanced) based on the raccomended required level to enter in a specific zone.

7 Automatic label annotation

The next step consists in automatically labeling some players' behavious based on some fixed conditions:

- **killer**: lvl_end>60 and n_pvp>40 and n_maps<50;
- **socializer**: lvl_end<15 and lvl_speed<1 and evolution<10 and n_city>10.
- **achiever**: time_hours>100 and evolution>25 and lvl_speed>8.
- **explorer**: time_hours>150 and n_maps>50 and lvl_speed<4 and evolution<25 and n_pvp<10.

For example, we label the behaviour of a player as killer if the associated avatar has a level higher than 60, has spent at least 40 hours in a pvp competitive maps and didn't visit many maps but just focused on a few of them (n_maps<50). This is reasonable since those kind of players own high level avatars and play the game with the purpose of competing with other players. On the other hand, socializers are not too much interested in evolving their avatar too much (evolution<20 and lvl_speed<1) but they are rather focused in making new friends and interact with other avatars. They are usualy low level players (lvl_end<15) and spend most of their time in maps where they can avoid dangers and where they easily find new players (n_city>10).
Tuning the above parameters, thus fixing the right criterias to decide the player's behviour, is the most crucial part of this project, in fact, setting different conditions can lead to different outcomes in the next phase. Given the nature of the problem, it can be very hard to test the correctness of the obtained results, thus, there is no evidence that our conditions are the perfect ones but they have been fixed experimentally.
This process should also label a small fraction of the dataset's entries. In fact, we are going to use this automatically labeled entries to predict the behaviours of the other players.

8 Behaviour prediction

In this phase we are first going to train a random forest classifier on the portion of data with the labeled behaviours. Next, we are going to use this model to predict and fill the missing values. It is important to say that, because of the nature of the task, we don't have a test set to check the accuracy of the result but most of the result depend on the previous annotated labels. In a future work we may replace the random forest with a semi-supervised algorithm such as a semi-supervised version of K-neightbours to spread the annotated labels across the whole dataset.

9 Data preprocessing for player's departure prediction

The second task regards predicting the number of players who are about to leave the game (departure prediction). The idea is the one to split the data of each player into a different dataset and keep one entry for each week of the year (from week 1 to week 52) by grouping the sessions with same ID and in the same week, therefore, the attribute timestamp has been transformed such that it indicates the appropriate week rather than the date and the time. Next, we dropped all the original attributes except timestamp and ID and replaced them with new features calculated as a function of the original columns:

- **evolution**: the level of the avatar at the beginning of the week minus the level reached at the end of the same week;
- **lvl avg**: it means "average level" and is calculated as the mean between the level of the avatar at the beginning of the week minus the level reached at the end of the same week;

- **time hours**: the number of hours a player played during the week. Note that play times of than 15 minutes have been smoothed to be 0. For example, a player who is about to leave the game might play less hours than the daily mean hours of normal players and could play less frequently because less interested in the game.

The following next two attributes involve also the player's history to be computed (previous rows). Since both are based on the feature time_hours, if a player played for less than 15 minutes in a week, that week would be considered as the player didn't play at all [4].

- **current absence**: Number of weeks the since the player last played;
- **weeks present ratio**: Number of weeks the player has been active divided by the total number of weeks since he first registered to the game.

Since ours is a supervised approach, we also need to set the appropriate label for each week of each player. For semplicity, we refer to the week the player has been active as ING (in-game), the weeks where he was about to churn as ATC and the weeks during the which he churned as CHR [4]. Next, we define a churn window (C) such that if there is no data for a player during at least a C weeks period, then that player is considered to have churned during that time, thus the relative weeks have been labeled as CHR. Note that this means that it is also possible for a player to churn for a period of time and then return. Moreover, the C weeks before a player churned have been labeled as ATC and the rest of the weeks have been labeld as ING. It is worth noting that ATC sequences might be shorter than C weeks long, this can happen if, for example, an ATC period is surrounded by CHR periods and doesn't have an activity period of at least C weeks in a row.
Secondly, we extract sequences of week of lenght L from each player (history) where the label of each sequence is the same of the label of the last week of the sequence, for example, a sequence composed of ING, ING, ATC, ATC weeks would have label ATC. Anyway, we are going to discard the CHR labeled sequences since they are not useful for our goal. Following, have been discarded as much ING sequences such to have an equal number of ING and ATC sequences, they have been standardized, mixed and splitted in training and test set with a test set size of 0.2.

10 Departure prediction

In our first attempt we used a SVM classifier with C=4 and L=1. L=1 means that the classifier takes in input only a single week and C=4 means that the 4 weeks before a player churned have been labeled as ATC. We will call this classifier as SVM_4_1.
In a second try we used again a SVM classifier but this time with parameters C=4 and L=7. In this way we don't only have the current week but also the history of the 6 previous weeks to use for our prediction. This model will be called as SVM_4_7.
Next, we used a LSTM deep recurrent model with recurrent layers of 50 neurons, and a final single neuron layer output with softmax activation. We trained it for 30 epochs with a batch size of 64, binary crossentropy as loss function and adam as optimizer algorithm. In addition, we setted a value of C=4 and L=10 and we call this network as LSTM_4_10.
Finally, we trained again the SVM_4_7 and LSTM_4_10 classifiers but this time setting the parameter C=1. It means that only the week before a player churned have been labeled as ATC. We are going to call these last two classifiers as SVM_1_7 and LSTM_1_10 respectively.

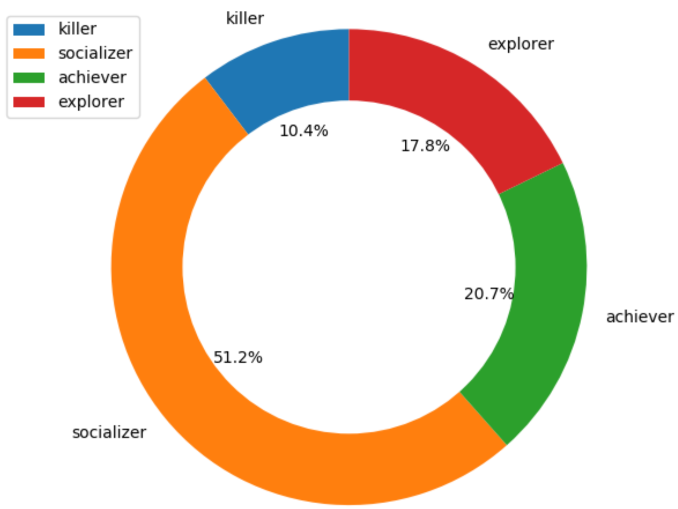

Figure 3: Pie chart representing the distribution of the four kinds of behaviour

11 Results

In the last section, we are going to show and comment the results we obtained in both tasks and to discuss possible alternative solutions which could further improve the quality of the obtained results.

11.1 Behaviour prediction results

In the above pie chart in fig3, representing the distribution of the four kinds of behaviour, we note that the socializer is the most frequent behaviour in World of Warcraft (at least during the year 2008) accounting for half of the total players. Killers are usually high level and expert players, hence, they are the less frequent behaviour since it is not easy finding many of them in the game. Finally, achievers and explorers almost equally divide the behaviour of the remaining players accounting for about 40%. An interesting observation is that many players end up changing behavior when they reach certain characteristics. An achiever, for example can become a killer or an explorer when it reaches the highest levels or the quests become scarce. Therefore, one possibility is that some explorers and killers can be classes as arising from achievers [2]. We can observe in the top-left chart of fig2 that the explorers and the killers are the players visiting more maps while the achievers travelling on a fewer number of zones since they are more focused on playing on that maps which can guarantee a faster growth of their avatar. Despite both visiting many maps, killers and explorer won't meet very often. In fact, the first spend more time in competing with other players in PvP zones while the latter focus mainly on exploring world map areas. Conversely, the socializers can't afford to visit high level maps. Firstly, they are not enough skilled to face strong hostile NPCs. Secondly, both killers and achiever won't show a friendly attitude toward them. Thus, socializers prefer staying in beginner or neutral maps such as cities where they can meet other socializers and make new friends while avoiding the wild and PvP maps.
Next, we are going to analyze the top-right chart in fig2 representing the relation between the behaviours and final level reached by the players belonging to each behaviour. As expected, killers, who are very competitive players, are amassed around the 70-80 level ranged and only a few lower level players play the game showing this kind of behaviour. Among the socializers, although they are gathered aroung lower levels, there are still some higher level players assuming this kind of behaviour. Instead, Achievers spreads along a wide range of levels where explorers are more concentrated toward higher level than achivers but lower than killers. We can observe that a natural way of behavior changing is that a new player, once creates a new avatar and starts playing, at the beginning may act like a socializer in order to get familiar with the game mechanisms and design. Later, he discovers he really likes this game so he turns into an achiever in order to enchance his character, complete as many quests and possible and rush toward the end-game. Finally, he can decide whether becoming a

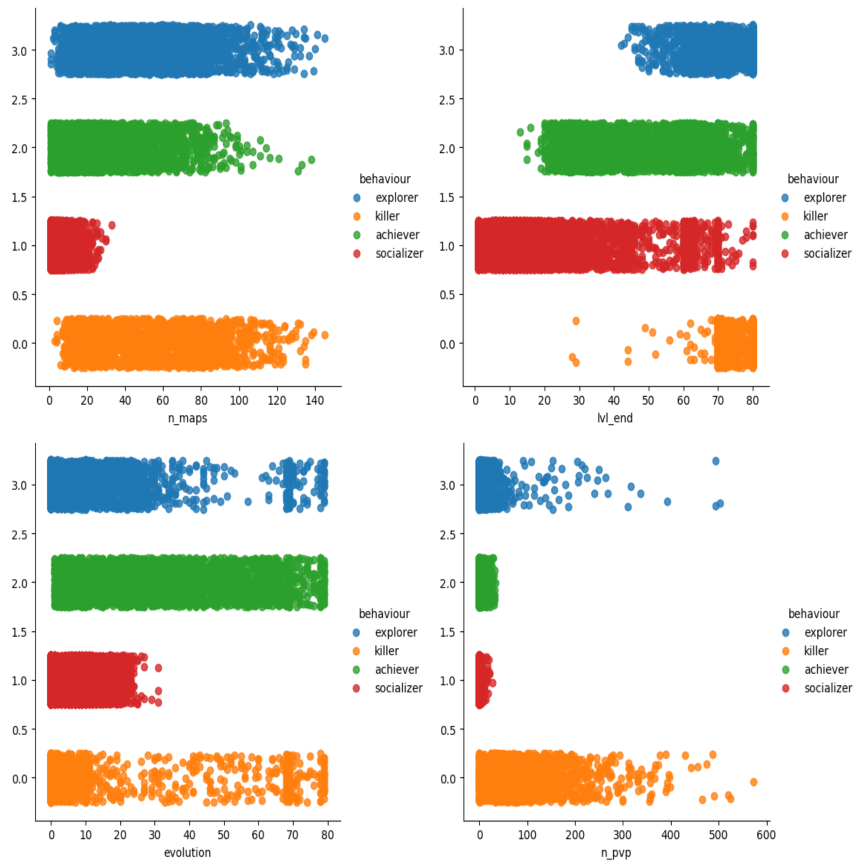

Figure 4: Relationship between players' behaviors and various features such as number of maps visited, highest level reached, evolution and time spent in pvp arena.

killer or an explorer based on his own intentions and personality. Of course, it is also possible that a player never changes his behaviour or changes it by following a different way.
From the bottom-left chart always in fig2, which shows for each behaviors, how fast those players tend to increase the level of their avatar, we can observe that the explorers tend to have a medium to slow progress, but we have already seen that their level is above the average, therefore, they may walk freely in the world without being killed by monsters considered aggressive. The achievers in turn have rapid development, as they try to develop their character doing quests and using the appropriate maps for training. Their speed of evolution is given by the large amount of experience won through quests in the game and also the efficient management of resources by the player, and the correct use of the world (to know where to play to evolve faster) to obtain quick experience. The socializers usually have subdued to a certain level and when they reach a comfortable level, where they can use all the items considered fashions offered by the game. As the socializers, killers and explorers are not interested in growing their level since they already have high-level avatars.
Finally, from the bottom-right chart in fig2, showing the relationship between player's behaviour and number of hours spent in pvp arenas, we can note how the PvP maps are dominated by the killers with most of them spending more than 100 hours in a year playing on them with only a few players belonging to the other three classes daring face them.

11.2 Departure prediction results

For this task, we grouped our results in two tables, the first one showing the metrics values obtained predicting the the ING (active) weeks and the second one predicting the ATC (about to churn) weeks which is the main goal of this project. Those predictions have been computed by the 5 different models explained in section 10. We can conclude saying that predicting player's departure is a

Classifier	Precision	Recall	F1
SVM4-1	0.819	0.738	0.777
SVM4-7	0.842	0.740	0.788
SVM1-7	0.886	0.804	0.843
LSTM4-10	0.855	0.749	0.798
LSTM1-10	0.880	0.828	0.853

Table 1: Classification results regarding the prediction of the active weeks (ING sequences).

Classifier	Precision	Recall	F1
SVM4-1	0.710	0.797	0.751
SVM4-7	0.704	0.817	0.757
SVM1-7	0.793	0.878	0.833
LSTM4-10	0.710	0.829	0.765
LSTM1-10	0.821	0.875	0.847

Table 2: Classification results regarding the prediction of the about to churn weeks (ATC sequences).

quite challenging task and the best we can get is estimating the number of players who are about to churn and take measures to counter the problem. We can also see how LSTM, which is a network specialized in temporal series, produces slight better results than SVM and the value of predictions with C=1 outperforms the ones with C=4. Of course, a dataset collecting more statistics about players activities such as quest completed and enemy killed would have surely had a good impact on the overall result. Moreover, we know that each player plays according to a specific behaviour. Therefore, one possible solution would be dividing the players in four datasets, one for each behaviour and make predictions separately taking advantage of their characteristic features. For example, if an explorer, who is a player who enjoys exploring as many maps as possible, should suddenly decrease its number of weekly maps visited, it might be a signal that the player could feel tired about the game, thus he is going to churn.

References

[1] Simon Demediuk & Alexandra Murrin & David Bulger (2018) *Player Retention In League of Legends: A Study Using Survival Analysis.*

[2] Marcelo Souza Nery & Victor do Nascimento Silva & Roque Anderson S. Teixeira & Adriano Alonso Veloso (2017) *Setting Players' Behaviors in World of Warcraft through Semi-Supervised Learning.*

[3] Yun Tarng & Kuan-Ta Chen & Polly Huang (2009) *On Prophesying Online Gamer Departure.*

[4] Marco Tamassia & William Raffe & Rafet Sifa & Anders Drachen (2016) *Predicting Player Churn in Destiny: A Hidden Markov Models Approach to Predicting Player Departure in a Major Online Game.*

[5] Superdata: https://www.superdataresearch.com/

[6] Fred Reicheld: https://en.wikipedia.org/wiki/Fred_Reichheld/

[7] Wowah dataset: https://www.kaggle.com/mylesoneill/warcraft-avatar-history

[8] World of Warcraft: https://worldofwarcraft.com/en-us/