Personalized adaptive model in games using Machine Learning

Nitish Rangarajan
Graduate Student in Computer Science
College of Computing and Informatics
University of North Carolina, Charlotte
nrangara@uncc.edu

Abstract—With the computer games evolving continuously and rapidly over the years in terms of their complexity and audiovisual qualities, the game AI (Artificial Intelligence) has been largely neglected. The recent study made by the game design companies show that the quality of the game AI sets the game apart from the other mediocre games. There is an essential need to develop a gaming model that can generate an engaging and realistic experience for the player by creating more challenges which can avoid making the player bored if it's too easy or frustrating if it is too hard. [1]The personalized user content generation in games would increase the self-motivation in players as they deeply immerse themselves into the virtual gaming world. An adaptive user model is necessary to capture the skills of the player and make use of an automatic game content altering algorithm to meet the needs of the player by capturing the player's game logs and analyzing them. In this report, I summarize the World of Warcraft Avatar History dataset which is a public asset of the research community, comprising of attributes like level, race, class, location, social guild and understand the gameplay behavior, interactions, predict why the users are unsubscribing from the game using Deep Learning and enhance the game AI on generating user-tailored contents like bonus levels using the results.

Keywords— Artificial Intelligence, Computer Games, User Models, Deep Learning

I. Introduction

The Massively Multiplayer Online Games(MMOG) are one of the popular type of games where users can play online with other users. The reason for its popularity is that users need not use any of their internal memory to play the game. Huge servers in the backend enable such users to play the game without any glitch. Some of the games that are MMOG are DOTA, StarCraft and World of Warcraft.

According to the game designers, players' behavior is one of the key factors they must address when designing games. But the player modelling depends on the genre of the game because the user's behaviour is dependent on the how the game functions. For example, in strategy games players need to organize their army better than the enemies but other RPG games or First-Person Shooter games doesn't need this technique. But some techniques like reacting to the abnormal network condition in games are common to all.

Predicting the number of players who would be interested in the game before launching a game would be difficult since it involves lot of factors like the strategies used to market the game, the date it was launched, cover design and cultures. To predict how many players are likely to stay with the game is a more feasible option than the former since it is related to the interaction of the player with the game world like how quick is his avatar moving to the next levels and the number of days he stays with the game. On predicting the unsubscription decision of the players, we can get the following ideas:

1. Players are likely to unsubscribe from the game because of some aspects like the design that don't meet their requirements. We can find the last level or the character class that the player had played and improve on those features.
2. Using the prediction model, we will be able to find if any of current players are likely to unsubscribe from the game by forecasting the future. We find a similar user from the past users and compare the trends of the users. If they match, we can do something about improving the game or the servers to retain the players.

To understand the gameplay behaviour of the player, we analyze the gameplay time of the users since it provides an idea if the players are going to stop subscribing the game after playing for a period of days. The goal of this study is to predict the player unsubscription that takes the player's logs and predicts how likely are they going to stay with the game.

In this study, the World of Warcraft Avatar History dataset that contains the game play time of 91,065 avatars will be used. [2] Avatars refer to the player's character used to play the game. An avatar belongs to one of the character class like Hunter, Shaman, Warlock, Warrior, Druid, Priest, Mage and Rogue. Each character class has different powers and weapons. The avatar was observed for a period of three years from Jan 2006 to Jan 2009 and the logs of each day were stored. The dataset is now a public asset of the research community.

II. DATA DESCRIPTION

The raw dataset was collected as a log from the servers. The log has two types of arrays namely Persistent_Storage and RoundInfo. [3] The avatar's history is stored in the Persistent_Storage. Each element in this array contains the information about the avatar during the observed timeframe. They contain 11 fields which are separated by commas. They are dummy, query_time, query_sequence_number, avatar_ID, guild, level, race, character_class, zone, dummy and dummy. The meaning of each of these fields and sample observations are listed below

Fields	Meaning	Values
query_time	The time the element was recorded	From Jan 2006 - Jan 2009
query_sequence_number	Sequence log information	Integer >=1
Avatar_ID	Unique ID assigned to each player	Integer >= 1
Guild	Social guilds for players	Between [1, 513]
Level	The levels in the game	Between [1, 80]
Race	Looks of the character	Blood Elf, Orc, Tauren, Troll, Undead
Character_Class	Types of characters in the game	Death Knight, Druid, Hunter, Mage, Paladin, Priest, Rogue, Shaman, Warlock and Warrior
Zone	Zones in the game world	One of the 229 zones

TABLE 1: FIELD DESCRIPTION

Sample Observations
[1] = "0, 12/31/05 23:59:46, 1,0, , 5, Orc, Warrior, Durotar, no, 0"
[2] = "0, 12/31/05 23:59:46, 1,1, , 9, Orc, Shaman, Durotar, yes, 0",

TABLE 2: EXAMPLE RECORDS

To process these data, a regular expression in Python was written that finds each of the fields from the 138,094 files and stores it in a csv format[1]. There were 36,306,188 observations for the three years in the csv file and data was roughly around 2.23 GB.

A. Initial Observations:

For the preliminary analysis, the number of characters at each level and number of logins at each level were checked.

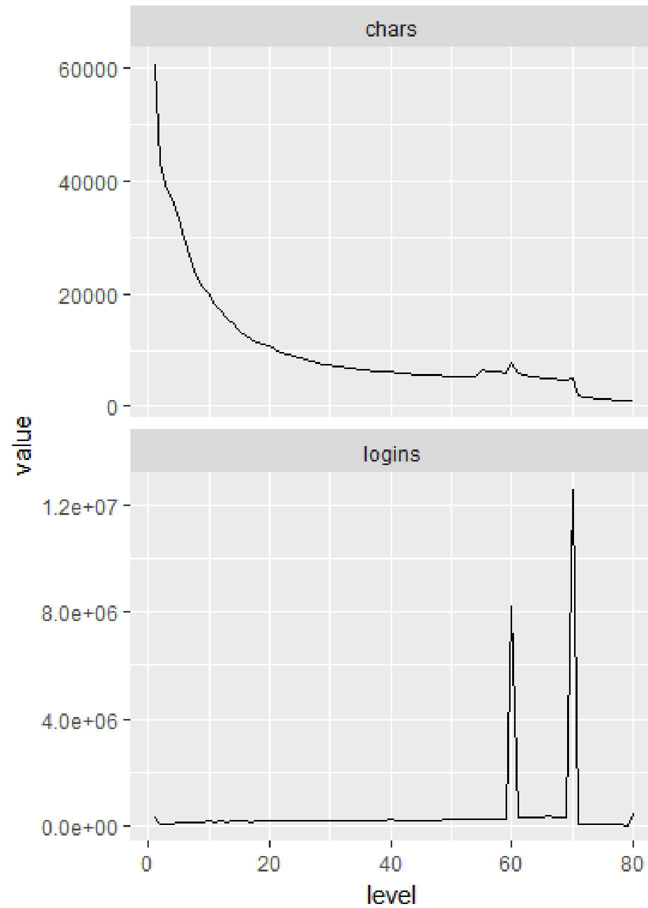

Fig. 1. Characters and Logins by level

The results of the number of characters at each level wasn't unsurprising as there were lots of people in the first few levels, but the results show that there lots of people who frequently login when they are at the level ~70.[2]

[1] *The detailed code to this pre-process can be found at https://github.com/nitishr12/Personalized-adaptive-model-in-games-using-Machine-Learning/blob/master/data_preprocess.Rmd*

[2] *The detailed code to these analyses can be found at https://github.com/nitishr12/Personalized-adaptive-model-in-games-using-Machine-Learning/blob/master/analysis_report.Rmd*

The playing patterns of players over the years can be found by converting the number of observations of each player to hours by dividing the count for each user by 6 since we know that the observations were collected every 10 minutes.

[4] We can see below that most number of people log in for 5 hours and there were some hardcore players who log in for 24 hours a day.

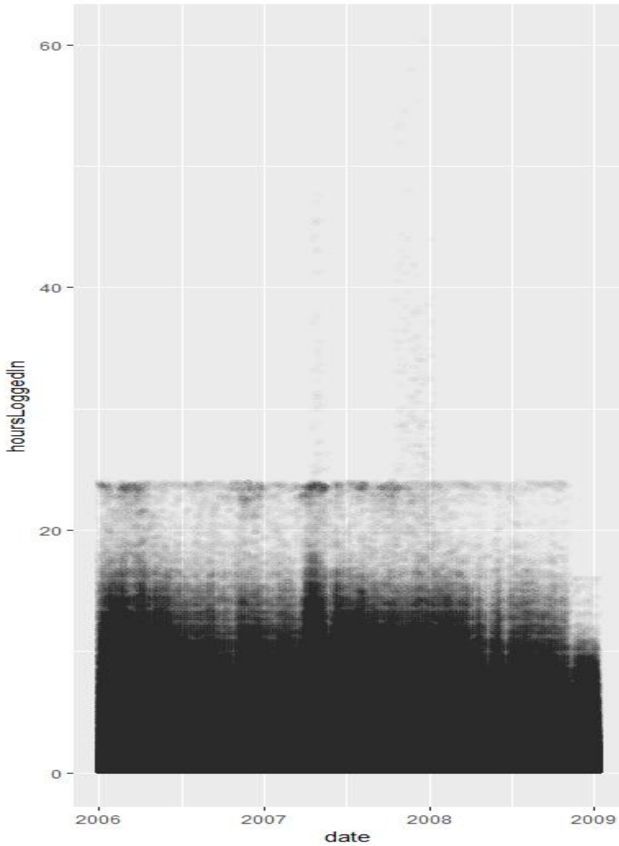

Fig. 2. Number of hours logged by users in the years

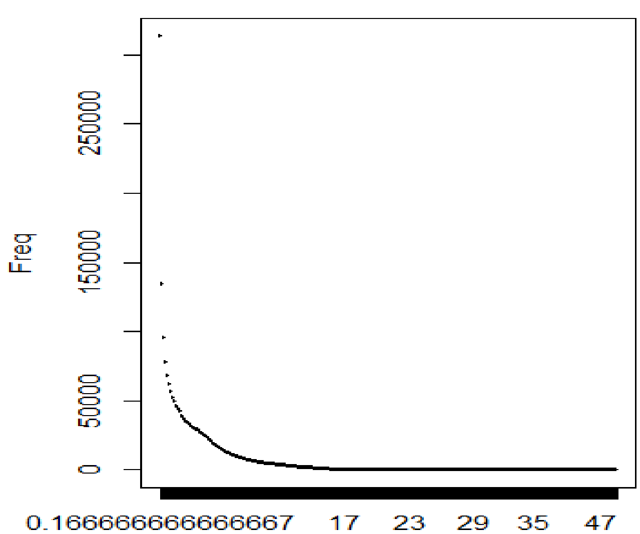

Fig. 3. Average Player Login Hours In a Day

On checking how many average players are logged in each day on a hourly basis to check if there are actually hardcore players, we can see a gradual dropoff as expected in the number of players. Login time of 47 hours might be because of duplicate observations.

B. Predicting the Unsubscription Rate

After learning about the data, the Deep Learning Network was used with the Keras library in python. [5] Keras is a high level neural networks API that is written in Python and developed with a focus on enabling fast experimentation". The sequential deep learning model in Keras was used since that is best suited to classify human actions with limited knowledge.

Before building the training and the test data from the actual dataset, we need to ensure that there are no leakages in the training data. By leakages, I mean the information in the training data about the player activity in 2008-2009. Therefore, we need to first separate the player logs before and after 01/01/2008. We come up with a first dataset on player activity before 2008 and another dataset on player activity after 2008. We merge both these datasets and partition 80% for the training and 20% for the test. This method would help the model train better since there might be two entries for same user if the user was active during 2008.

To predict the unsubscription, a deep learning model has been built that will be trained on the data from 2005 to 2007 to predict the unsubscription for the year 2008-2009. So, the player by total number of entries, number the days online from 2005 to 2007 with the last level, the minimum and the maximum guild were grouped. These are my x variables. For the y variable in the network, a binary variable was created that denotes if the user was online during 2008-2009 or not. There were totally 91,045 avatar records. The training data that contained 80% of the avatars and test data that had 20% of the avatars was created.

C. Building the Sequential Model

The initial sequential model created using Keras had a lesser accuracy around 60% because of two hidden layers and there were less than 10 units in each layer. Hence a sequential model was created with 5 hidden layers with the first hidden layer containing 512 units, second hidden layer containing 256 units, third hidden layer containing 128 units, fourth hidden layer containing 64 units and used the "relu" activation function. The last hidden layer contained 2 units because of the size of y and used the "softmax" activation function.[3]

[3] *The detailed code to these models can be found at https://github.com/nitishr12/Personalized-adaptive-model-in-games-using-Machine-Learning/blob/master/model.Rmd.*

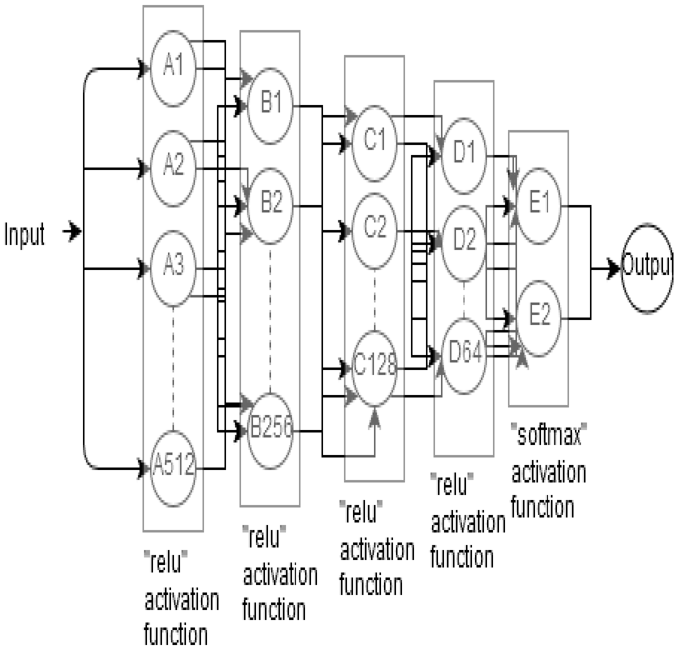

Fig. 4. Deep Learning Model

The model was compiled using categorical cross entropy loss function, RMSprop optimization algorithm and accuracy metrics. To fit the model using the training data, 50 epochs were used with a validation split of 0.2 to prevent overfitting.

Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 512)	3584
dropout_1 (Dropout)	(None, 512)	0
dense_2 (Dense)	(None, 256)	131328
dropout_2 (Dropout)	(None, 256)	0
dense_3 (Dense)	(None, 128)	32896
dropout_3 (Dropout)	(None, 128)	0
dense_4 (Dense)	(None, 64)	8256
dropout_4 (Dropout)	(None, 64)	0
dense_5 (Dense)	(None, 2)	130

Total params: 176,194
Trainable params: 176,194
Non-trainable params: 0

Fig. 5. Summary of the sequential Deep Learning model created using Keras

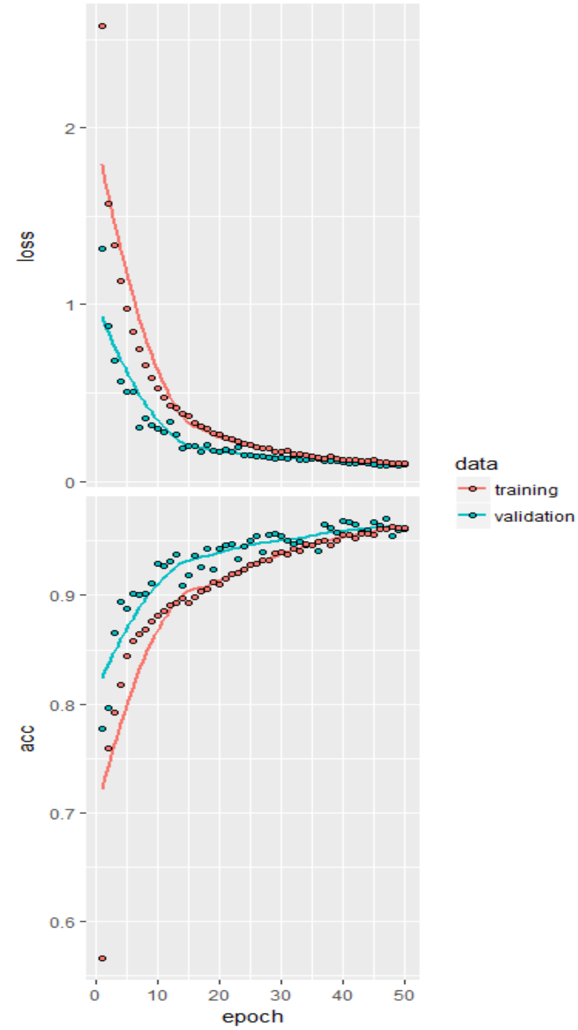

Fig. 6. Accuracy And Loss of the Deep Learning Model

[6] The “ReLU” is an activation function defined by

f(x) = ln(1+ex)

where x is an input to the neuron. It is known as Rectified Linear Unit because that unit uses a rectifier as an activation function. These units are mostly used in the deep neural nets.

[7] Softmax activation function is used as a final layer of the deep learning since it provides a probability distirbution of k possible outcomes moslty to denote a catagorical distribution. In our case, the y variable has 2 possible outcomes. Hence we use softmax activation for the two units in the final layer.

[8] The RMSprop is an adaptive learning rate algorithm developed by Geoff Hinton in his Coursera class. RMSprop increases the step rates, keep the learning rate constant by exponentially decaying the average of squared gradients.

The model had an accuracy of 95.88% and loss value of 0.092 when it was evaluated with the test data.

D. Predicting the Unsubscription rate with SVM

The same training data with the labelled X and Y variables is fed to the [9] Support Vector Machine as an input and then predict the players who are likely to unsubscribe using the test data. A summary of the SVM model created using the training data is shown below.

```
Call:
svm(formula = y_train ~ ., data = cbind(x_train, y_train))


Parameters:
   SVM-Type:  eps-regression
 SVM-Kernel:  radial
       cost:  1
      gamma:  0.1666667
    epsilon:  0.1


Number of Support Vectors:  14102
```

Fig. 7. Summary of the SVM Model created using the training data

We predict the test data using the SVM model created. We can see below that the model produces a 90% accuracy with a precision of 82% and recall of 59%.

```
> accuracy(Prediction, y_test)
[1] 0.8998245
> precision(Prediction, y_test)
[1] 0.8196532
> recall(Prediction, y_test)
[1] 0.5861926
```

Fig. 8. Accuracy, Precision and Recall of the SVM model

On looking at the results, one can notice that there were lots of players who left the game (True Negatives) and only a few players stayed with the game (True Positives) using the Precision and Recall scores.

[10]Precision refers to the fraction of the retrieved instances that were correct to the total retrieved instances.

[10]Recall refers to the fraction of the retrieved instances that were correct to the total number of correct instances in the corpus.

III. Future Work

Since the dataset comprises of 36 million logs of 91,045 users, there could be many different predictions based on it. Some work that could be done has been listed below.

A. Improve the efficiency of the SVM mode

To improve the efficiency and accuracy of the SVM model, we can set some tuning parameters like the cost and gamma value ranges to match the efficiency of the Deep Learning model.

B. Classifying the users based on the Playing History

On checking the playing hours of six random players, it was seen that every player fits into the two types of playing pattern. [12] The first pattern is called the “fade-out” pattern where the playing time of the user reduces gradually along the days. The reason for the reduce in playing time is because of the boring design or complexity of the level. The second type of pattern is the “sudden-out” pattern where the playing time stops suddenly. They might play 12 hours every day in a week to make up for the disappearance in the next week.

When there is a need to reduce the unsubscription rates, it might not be a good idea to concentrate on the “sudden-out” users since they might leave the game because of assignments, work or any personal reasons. Hence there is no way to bring them back to the game. But “fade-out” users leave the game because they feel the game to be boring or might have found a better game. To bring these users back to the game, we can improve on the design or give them some bonus credits to make them stay with the game.

We can classify these users based on the game time and work on the “fade-out” users specifically so that unsubscription rate can be reduced as well as predict any kind of such users in the future before they unsubscribe and make them stay with the game. With this technique, we can train the Deep Learning model better with more classes.

IV. Conclusion

In this study, we study the player’s game logs for the World of Warcraft Avatar dataset during a 3-year period and a Deep Learning model to predict gaming departures has been proposed. In addition, we also proposed the need to classify the gamers into two categories. These predictions would benefit the gaming industry by enabling the companies to keep the subscribers to stay motivated, engaged with the game and also help to improve the game design at certain levels.

Appendix

A. Appendix A

The entire World of Warcraft avatar history dataset is a public asset of the research community since it is believed that it could be used for various research purposes. It is available for free download at http://mmnet.iis.sinica.edu.tw/dl/wowah/wowah.rar.

B. Appendix B

The regular expression used to process the logs is given below.

regex=
re.compile(r'^.*"[\d+],\s(.*),\s?(\d+),\s?(\d+),\s?(\d*),\s?(\d*),\s?([A-Z].*),\s?([A-Z].*),\s?([A-Z].*)".*$').

The first field is for the dummy integer, the second field is for the query processing time, the third is for the sequence number, the fourth is for the character ID, fifth for the guild value, sixth is for the level number, seventh for the race, eighth is for character class and ninth is for the zone.

REFERENCES

[1] Georgiou, T. & Demiris, Y. User Model User-Adap Inter (2017) 27: 267. https://doi.org/10.1007/s11257-017-9192-3

[2] Blizzard Entertainment Inc, World of Warcraft https://worldofwarcraft.com/en-gb/game/new-players-guide.

[3] P.-Y. Tarng, K.-T. Chen, and P. Huang. An Analysis of WoW Players Game Hours. In *NetGames '08: Proceedings of the 7th ACM SIGCOMM Workshop on Network and System Support for Games*, pages 47-52. ACM, 2008.

[4] Y.T. Lee, K.T. Chen, Y.M. Cheng, and C.L. Lei, "World of Warcraft Avatar History Dataset" Proceedings of ACM Multimedia Systems, Feb 2011.

[5] JJ Allaire, F. Chollet, RStudio, Google, "Keras" https://keras.rstudio.com/.

[6] Wikipedia Inc, Rectifier(neural networks) https://en.wikipedia.org/wiki/Rectifier_(neural_networks).

[7] Wikipedia Inc, Softmax Function https://en.wikipedia.org/wiki/Softmax_function.

[8] Climin Developers, rmsprop - climin 0.1 documentation http://climin.readthedocs.io/en/latest/rmsprop.html

[9] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, 2000.

[10] Wikipedia Inc, Precision and Recall. https://en.wikipedia.org/wiki/Precision_and_recall.

[11] P.-Y. Tarng, K.-T. Chen, and P. Huang, "On Prophesying Online Gamer Departure," Proceedings of ACM NetGames 2009 (poster), 2009.