Predicting Time of Purchase in Games with Microtransactions

Ondřej Pluskal[1]

[1]Dept. of Computer Science, Czech Technical University, Karlovo náměstí 13, 121 35 Praha, Czech Republic

pluskond@fel.cvut.cz

Abstract. *This paper focuses on predicting player behaviour in two-player games with microtransactions. Typically the games are for free and companies generate their revenue by selling in-game goods. We focus on modeling the recommendation system techniques in a novel way, predicting the time of purchases rather than the most likely product to be purchased. The player model is based on in-game signals, such as players success, curiosity, social interactions etc. We had access to a Pool Live Tour game dataset made by Geewa. We created a model based on regressing the time before purchase using machine learning. We report promising results in predicting the time of purchase events.*

Keywords

machine learning, regression, online games, data mining

1. INTRODUCTION

Today's popular past time is playing computer games. In this paper we specifically focus on how to improve the monetization of two players online games. The monetization works in three different schemas. You purchase a game for a set price and you can use it indefinitely. You can purchase a game for a monthly subscription. The third option is to have the game for free but introduce small payments to the game for different in-game items. Players then can unlock different cosmetic outfits for their characters or gain items that give them certain advantage over the other player. The goods are purchased in small payments, microtransactions, for real-currency. The number of paying users, conversion rate is the *Key Performance Indicator* (KPI). To increase the KPI users are bombarded with ads [1]. The problem is that very frequent advertisements do not lead to conversions but they lead to advertisement fatigue and advertisement wear out.

Both phenomenons are recognized in marketing models [2]. Advertisement fatigue is the event of a customer no longer liking or buying goods from the advertisement, because the advertisement is bothering them too much. Advertisement wear out means that the customer will ignore the advertisement and it would have no effect.

These two negative effects can be reduced by advertising the in-game goods only when the player is likely to make the purchase. When the ad's timing is correct the likelihood of converting through the advertisement is increased. This improves the revenue. Building such a system by an expert is impossible, simply because we are analysing a large-scale dataset of thousands of players.

We proved our novel idea on a *Pool Live Tour* (PLT) game made by Geewa. PLT is a virtual pool game, that can be played in a web browser or on a tablet. This game is played by 2.5 million daily active users[1] all over the world. The in-game good are better cues giving a slight advantage. Certainly, buying a better cue does not guarantee a win it only increases player's chances. To design the model Geewa provided us with two datasets, dataset D_1 with a sample of monthly users activity, totalling 272k unique user ids, and dataset D_2 with monthly activity of a subset of users, that registered that month and converted, totalling 11.5k unique user ids. The datasets contain client actions collected from the clients as well as events generated by the server. The datasets in their raw form have over 30 GB.

The paper is structured as follows. In Section 2 we describe the state of the art. In Section 3 we state the problem at hand and propose a machine learning approach using regression. In Section 4 we describe the properties of the dataset. In Section 5 we present promising results of our experiments. Finally in Section 6 we conclude the paper and propose future steps of this work.

2. RELATED WORK

In the literature there are a few studies concerning player modelling using machine learning and data mining methods. We have identified a few trends in the literature, from methods supporting AI, finding similar players to support the developers, to cheater detection. Up to our knowledge, nobody published any work about building recommen-

[1]Collected from http://corporate.geewa.com/game/pool-live-tour/ on 2nd Feb 2014

dation systems in games. This section therefore covers not only player modelling, but also recommendation systems.

In cheater detection studies there are several studies using machine learning and data mining methods. The studies define the problem as a classification problem. They use derived rules [3], SVM [4] and Hidden Markov Models [5] with a set threshold. The features were consisting of playtime lengths[3] or action frequencies [4]. Bot detection tasks are studied in several game settings. These are mainly MMOGs[2] [3, 4] and FPS[3] games [5].

More descriptive models were made in the case of segmentation [6, 7, 8], a problem of finding distinct groups of users. Segmentation is viewed in all of these articles as unsupervised machine learning problem. Two papers consider player segmentation task [6, 7], the third segments organised groups of players called guilds [8]. Player segmentation is done using cumulative features including playtimes, game specific estimates of success (e.g. number of trials per level, number of kills etc.). The approaches used for this task vary from NMF[8], k-means, Simplex Volume Maximization[6], to SOM[7]. In all researches the main challenge is the number of users and therefore the problem of large-scale datasets. All the interesting information are extracted from game logs, that the producers store in databases acquired by telemetry. The games used in player segmentation are both single player games [7] as well as multiplayer games [6, 8]. The

There are more papers[9, 10] using supervised machine learning. Another paper tries to predict the players maximal progress in a single player game [9]. This is done using machine learning methods. By monitoring gameplay features in the early levels of the game and solving this problems both as classification problem as well as regression problem. Both problem abstractions show promising results. Another paper [10] studies the prediction of strategy that a competitive RTS[4] player would play from his actions done in game at certain time point. This study aims on creating an AI that by predicting the strategy probabilities would infer correct counter strategy.

Since up to our knowledge in the research of player modelling there is our research[15] we have tried to find resources in the setting of web recommendation systems aimed on recommending relevant goods, entertainment or web pages to specific users based on their preferences or browsing history.

In recommendation systems there are many ways from which data we should predict what a user might like. This can be done via contextual information, where in advertisement we try to find ads with similar context to the site. Also user based recommendations, where we look on similar users and recommend goods to them that others purchases (Collaborative filtering). And the last approach is that we use sequential data and try to find sequential patterns that might help us what the user might want to see next. We try to find out similarities with web recommendation, which is in active research for more than ten years[11]. Also hybrid approaches [12] using combinations of different approaches can be seen mainly in the Netflix Prize competition, where researchers accomplished to improve accuracy of the Netflix recommendation system by 10%.

Other approach is using sequential patterns [13, 14]. The idea comes from web-page recommendation, where a users browsing in some domain might share the same browsing sequential patterns as other users. Finding the most common patterns and creating a patricia trees from them is the core idea. Then assuming the Markov property (dependence of next state only on previous state, here states are web pages) the recommendation system can show the most probable web pages by traversing the patricia tree and suggesting the most probable pages.

3. TASK DEFINITION

The main motivation of this paper is to increase the revenue generated by games using microtransactions by reducing the amount of advertisement sent to the user. In the ideal case we would like to suggest to the user to purchase in-game items only in the times he would really buy the in-game item.

We will be explaining our approach on the PLT game. In this game there are two currencies. The first one are coins, that can be acquired by playing, daily rewards and earning trophies. The second one are gold coins, that can be only acquired by purchase using real currency. The task at hand is predicting the buy of a cue for gold coins. We divided the task into two stages, first we try to predict that the player would buy a cue of any kind, then we decided to only predict the gold cue buys. Both stages use only users of which we have information from their registration. For the second stage we use a segment of paying users as a subset of all users whose behaviour we would like to model. From the log data, described in section 4, we can identify these events for each player. We propose that the event of purchase depends on the player in-game behaviour.

In our previous paper we decided to model the time to the purchase unlike the event that after a game the user will make a purchase before the next game. In this paper we compare this approach to modeling the time before the purchase. The time will be measured in hours.

It is clear, that the purchase of the in-game item might not be dependent on what happened in his early career, if even the player does not remember it. This can be captured in the feature vector $\mathbf{x}_i$. This proposed task definition is robust enough that the design of the feature vector is not dependent on the game design. The only demand from the game design is that the games have clear beginnings and endings

[2]Massive Multiplayer Online Games
[3]First Person Shooter
[4]Real Time Strategy

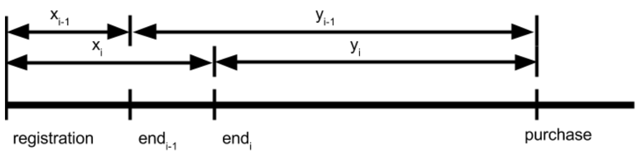

Fig. 1. Feature vector $\mathbf{x}_i$ and the corresponding target variable.

(segments) and the game offers an in-game item as the target of the classification.

There are two different cost functions we would want to minimize when modeling the time before purchase. The first one is the squared error. This kind of modeling might not be appropriate since for the advertiser it is much more important to know whether the user is likely to purchase in 1 hour versus 2 hours rather than 1 week or two weeks. Because of this we introduce the logarithm of squared error. This way we can distinguish between small and large numbers and penalize it accordingly to intuition. The logarithm base was chosen to be 10.

4. DATA COLLECTION AND EXTRACTION

The dataset of raw data in form of logs were provided by Geewa from their game PLT. The data come from their internal reporting system, that is used by their analytics in order to better understand how the players are playing the game.

These are live data from real players in their homes captured by telemetry and stored on Geewas side. The datasets contain logs from the client residing in player's computers as well as actions generated by the server. The dataset D is a set of all users that registered and bought some cue in one moth (1st Nov 2013 - 30th Nov 2013). The size of raw data is 6.6 GB.

The log data contain information about user logins, their in-game actions, from starting a game to shooting with a cue. The dataset was provided in a single csv file. We had to split the single file into a file for each player and those needed to be sorted according to the time when they were generated. Both datasets are exported from relational databases in form of one large table with all events in a structured form.

The dataset D consists of a subset of players who bought a gold cue. This dataset was extracted from Kontagent DataMine product used by Geewa to store the actions of their customers. We ran into problems with the Kontagent DataMine, such as a limit on the exported csv files to 1.1GB and not valid CSV files. Also the infrastructure of the databases allows us to extract events from server and from client separately. This process unfortunately yielded only 4545 players who registered and played at least 10 games.

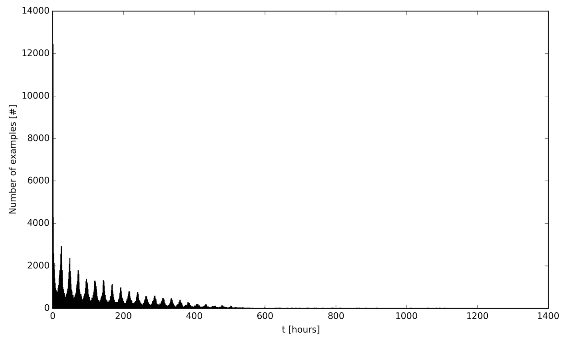

Fig. 2. Histogram of how many games happened each our before purchase.

These users played at least 10 games and bought a cue. Each player played about 64 games on average. The smallest amount of time before purchase is 16 seconds and the longest is more than 46 days. In Figure 2 we can see the histogram of examples, where each bin contains the number of games, that happened t hours before the purchase.

It is an interesting observation, that the distribution of the purchases happen most often in 24 hour intervals. The explanation is that people return to the game in a 24 hour periodical interval, copying their daily habits.

The dataset was processed and each feature x_i was extracted the same way as in our previous paper [15]. The target variable time before purchase was computed as the difference between the time of purchase and the end of a match i.

5. EXPERIMENTS

As we mentioned in Section 3 we will be using two different measures for the performance of classifiers. One will be the mean squared error (MSE) and the other one the mean squared logarithmic error (MSLE). In order to create a valid training protocol, we divided the dataset into two parts, training set and testing set, where training set is 20% and testing set is 80%. If parameter tuning is needed in order to find the best model we use 5-fold cross-validation. We have chosen scikit[5] library in python for the training and evaluation of the regressors.

As the classifier to test the performance we have chosen Random Forest [16] a simple Regression Tree, Linear Regression and a Support Vector Regression (SVR). The choice of these models is based on a good performance of Random Forests with comparison to more classical linear models. The Random Forest is a well performing classifier in various different tasks and linear SVR and Linear Regression represents a simple but well grounded model. We also compare all the results to a Dummy Regressor, where we used mean

[5] Version 0.17 http://scikit-learn.org/stable/

	MSE	MSLE
Linear Regression	12830.25	0.4854
Random Forest	**12825.24**	0.4712
SVR	13509.99	0.4294
Tree Regression	13591.91	0.4879
Dummy Regression	16318.59	0.4745
Log Linear Regression	16059.22	0.3683
Log Random Forest	16202.42	**0.3654**
Log SVR	24756.59	0.4822
Log Tree Regression	16486.99	0.3848
Log Dummy Regression	16318.59	0.4745

Table 1. Performance of regressors on MSE and MSLE

of the training set. For the regressors that are sensitive to the scale of the input variables we use the Z-normalization.

In order to properly minimize the MSLE we should redefine the task and not learn the function $f(\mathbf{x}) = y$ but learn $f(\mathbf{x}) = log(y) = \hat{y}$. We can then directly use all the optimization methods with evaluation. We can get back the target variable using simple transformation $y = 10^{\hat{y}}$. Classifiers that are constructed this was will have a prefix Log in every table of figure. Also for each value that the regressor predicted to be below 0 we set the result to 0, since negative time does not make sense in this task.

In Table 1 we can see the results. The Random Forest and Log Random Forest won in their categories. But for both MSE and MSLE the Linear Regression achieves comparable results. In order to better understand the results we created a 2D histogram, where on x axis we plot the predicted y and on y axis we plot the true y. In Figure 3 we see that the Random Forest makes many mistakes but predicts rather lower times before purchase. For the logarithmic version in Figure 4 we can see that the Random Forest behaves similarly.

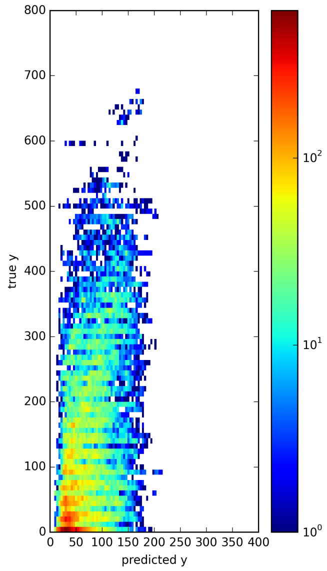

Fig. 3. 2D histogram of Random Forest Regressor with true versus predicted targets.

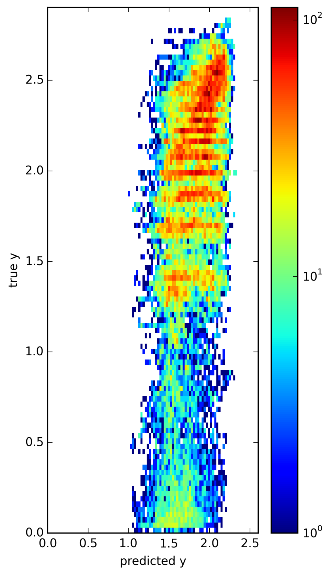

Fig. 4. 2D histogram of Random Forest Regressor with logarithms of true versus predicted targets.

6. CONCLUSION AND FUTURE WORK

We presented a new recommendation system applicable on games with microtransactions. Unlike a typical recommendation system we are not solving the problem of what we should recommend to the player, but when is the right time to recommend purchase of in-game goods. This approach allows dynamic advertisement placements to achieve higher revenue, higher conversion rates.

The task is defined as a regression problem where we want to predict the time of the next purchase measured from the end of a game. This would be then used as an input to a recommendation engine, that would present a pop-up advertisement for an in-game advertisement.

We tested the newly developed algorithm on the Pool Live Tour game with large player base. We created models predicting in-game items purchases and in-game items purchases for hard currency leading to real revenue. Our experiments have shown that the Random Forest algorithm was performing the best. We achieved a 21.4% improvement compared to a Dummy Regressor in normal scale and 23% improvement in log scale.

This system is yet to be tested in practice, because the only way how to measure the real impact on revenue is by A/B testing. We optimistically hope that the correct advertisement placement timing will make the players buy the in-game items for hard currency more frequently or earlier than today. This will bring higher revenue to the company.

We hope that we are bringing the game developers a game design pattern leading to higher in-game purchases, helping them to focus on improving the game design, game appearance in order to improve gameplay using machine learning methods.

As future steps we would like to create a testing protocol that would compare this approach to our previous approach using classification[15]. Also we would like to generalize this problem to other domains than just Computer Games, but also Web Usage Mining and other related topics.

Acknowledgements

I would like to thank my supervisor Jiří Kléma who helped me through various consultations. This work was supported by the Grant Agency of the Czech Technical University in Prague, grant No. SGS16/093/OHK3/1T/13.

References

[1] C. A. El-Nasr Magy Seif, Drachen Anders, *Game Analytics: Maximizing the Value of Player Data*. Springer, 2012.

[2] M. S. Lillien G., Kotler P., *Marketing Models*. Prentice Hall, 1992.

[3] A. R. Kang, J. Woo, J. Park, and H. K. Kim, "Online game bot detection based on party-play log analysis," *Computers & Mathematics with Applications*, vol. 65, no. 9, pp. 1384 – 1395, 2013. Advanced Information Security.

[4] R. Thawonmas, Y. Kashifuji, and K.-T. Chen, "Detection of mmorpg bots based on behavior analysis," in *Proceedings of the 2008 International Conference on Advances in Computer Entertainment Technology*, ACE '08, (New York, NY, USA), pp. 91–94, ACM, 2008.

[5] S. Yeung, J.-S. Lui, J. Liu, and J. Yan, "Detecting cheaters for multiplayer games: theory, design and implementation[1]," in *Consumer Communications and Networking Conference, 2006. CCNC 2006. 3rd IEEE*, vol. 2, pp. 1178–1182, 2006.

[6] A. Drachen, R. Sifa, C. Bauckhage, and C. Thurau, "Guns, swords and data: Clustering of player behavior in computer games in the wild," in *Computational Intelligence and Games (CIG), 2012 IEEE Conference on*, pp. 163–170, 2012.

[7] A. Drachen, A. Canossa, and G. N. Yannakakis, "Player modeling using self-organization in tomb raider: underworld," in *Proceedings of the 5th international conference on Computational Intelligence and Games*, CIG'09, (Piscataway, NJ, USA), pp. 1–8, IEEE Press, 2009.

[8] C. Thurau and C. Bauckhage, "Analyzing the evolution of social groups in world of warcraft," in *Computational Intelligence and Games (CIG), 2010 IEEE Symposium on*, pp. 170–177, 2010.

[9] T. Mahlmann, A. Drachen, J. Togelius, A. Canossa, and G. N. Yannakakis, "Predicting player behavior in tomb raider: Underworld," in *CIG*, pp. 178–185, IEEE, 2010.

[10] B. G. Weber and M. Mateas, "A data mining approach to strategy prediction," in *Proceedings of the 5th international conference on Computational Intelligence and Games*, CIG'09, (Piscataway, NJ, USA), pp. 140–147, IEEE Press, 2009.

[11] R. Suguna and D. Sharmila, "An efficient web recommendation system using collaborative filtering and pattern discovery algorithms," *International Journal of Computer Applications*, vol. 70, pp. 37–44, May 2013. Published by Foundation of Computer Science, New York, USA.

[12] Y. Koren and R. M. Bell, "Advances in collaborative filtering.," in *Recommender Systems Handbook*, pp. 145–186, Springer, 2011.

[13] B. Zhou, S. C. Hui, and K. Chang, "An intelligent recommender system using sequential web access patterns," in *Cybernetics and Intelligent Systems, 2004 IEEE Conference on*, vol. 1, pp. 393–398 vol.1, 2004.

[14] U. R. M. Suneetha K., "Web page recommendation approach using weighted sequential patterns and markov model," *Global Journal of Computer Science and Technology*, 2012.

[15] O. Pluskal and J. Šedivỳ, "Predicting players behavior in games with microtransactions," in *STAIRS 2014*, vol. 264, pp. 230–239, IOS Press, 2014.

[16] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

About Authors...

Ondřej Pluskal Was born in 23.1.1989. After finishing grammar school he proceeded to acquire his bachelor degree in Intelligent Systems and then his masters degree in Artificial Intelligence on CTU in Prague. Now he is pursuing his doctoral degree in Artificial Intelligence and Biocybernetics. His research topics include machine learning application in malware detection, CRM applications in games and more general rare events detection and classification in time series and sequential data. Recently he and a group of machine learning enthusiasts founded a series of talks about machine learning called Machine Learning Meetups Prage, where people from academia network with people from the industry and discuss current trends in machine learning.