Discovering Playing Patterns: Time Series Clustering of Free-To-Play Game Data

Alain Saas, Anna Guitart and África Periáñez
Silicon Studio
1-21-3 Ebisu Shibuya-ku, Tokyo, Japan
Email: alain.saas@siliconstudio.co.jp, anna.guitart@siliconstudio.co.jp
africa.perianez@siliconstudio.co.jp

***Abstract*—The classification of time series data is a challenge common to all data-driven fields. However, there is no agreement about which are the most efficient techniques to group unlabeled time-ordered data. This is because a successful classification of time series patterns depends on the goal and the domain of interest, i.e. it is application-dependent.**

In this article, we study free-to-play game data. In this domain, clustering similar time series information is increasingly important due to the large amount of data collected by current mobile and web applications. We evaluate which methods cluster accurately time series of mobile games, focusing on player behavior data. We identify and validate several aspects of the clustering: the similarity measures and the representation techniques to reduce the high dimensionality of time series. As a robustness test, we compare various temporal datasets of player activity from two free-to-play video-games.

With these techniques we extract temporal patterns of player behavior relevant for the evaluation of game events and game-business diagnosis. Our experiments provide intuitive visualizations to validate the results of the clustering and to determine the optimal number of clusters. Additionally, we assess the common characteristics of the players belonging to the same group. This study allows us to improve the understanding of player dynamics and churn behavior.

I. Introduction

In the past years, free-to-play (F2P) has emerged as the dominant monetization model of games on mobile platforms [1], [2]. F2P games are offered for free, and monetized by charging for in-game content through in-app purchases, with player retention being key to a successful monetization. The always-connected nature of mobile devices allows to constantly collect data about player behavior in the game. These data are used to guide design decisions for updates and release of additional content to maintain players' interest, sometimes in the form of periodic events giving access to new game content for a limited period of time [3].

This study is motivated by the idea that the automatic clustering of time series of player behavior can lead to a better understanding of player engagement. With daily active user bases ranging from thousands to millions of players, a game developer cannot know how every player reacts to a game or content update. At best, she can visualize averages of manually defined segments [4]. In this paper, we show that we can automatically cluster and visualize the main trends in player behavior and that we can determine differentiating characteristics of players belonging to different clusters. We also consider the evolution of players after the end of the time series studied, and we investigate the use of this clustering as a feature addressing temporal dynamics for further supervised learning applications, e.g. a churn prediction model.

Previous efforts on clustering game data appear in [5], [6], [7], [8], [9], with the common goal of extracting player pattern behavior. However, the focus of these studies is non-time-oriented data. On the other hand, in the work presented by [10], a clustering of time series is performed, but the measurements are obtained from PC games, not from F2P game data which allow a robust behavioral analysis.

The aim of the present paper is to identify similar patterns in unlabeled temporal datasets of player activity in F2P games. In order to discover natural groups of players, based on their behavior and interaction with the game, we apply diverse clustering techniques which focus on maximizing the dissimilarity between different clusters and maximizing the homogeneity within the groups.

To the best of our knowledge this is the first article that applies unsupervised learning techniques to cluster time series of player behavior from F2P games. We have successfully extracted relevant user patterns from two F2P games: *Age of Ishtaria* and *Grand Sphere*, which helps us to examine quickly the player activity, allowing a visual game diagnosis and an intuitive evaluation of the weekly-based game events.

The games chosen for this study are representative of the most played F2P mobile social role-playing games in Japan and they have also been successful worldwide, reaching several millions of players.

II. Clustering Time Series of Game Data

Time series consist of sequential observations collected and ordered over time. Nowadays, almost every application, web or mobile based, produces a massive amount of time series data. The goal of unsupervised time series learning, e.g. clustering methods, is to discover hidden patterns in time ordered data.

Clustering time series data has received high attention over the last two decades [11], starting with the seminal work of [12] in 1993. It has faced many challenges [13], among which one of the most important is probably the high dimensionality level that time series contains and therefore the difficulty of defining a similarity measure, i.e. the *distance* between series, in order to classify close patterns in the same group. Working with *raw* time series is computationally expensive and technically complicated. So as to cluster them efficiently, their complexity must be reduced through *representation* techniques [14], trying to maintain the characteristic features of the data. Both the dimensionality reduction and the similarity distance definition are obviously application-dependent.

Furthermore, with the fast increase of digital data, the clustering algorithms must be ready to deal with *Big Data* challenges [15], e.g. a vast volume of data to be processed with high efficiency and speed, sometimes even in real time. The literature on time series clustering is very extensive. For a comprehensive review about time series similarity search methods, check [16], [17], [18], [19].

In this Section we review the methods applied in the present paper to cluster the time series of player behavior. We briefly explain separately the representation methods and similarity measures used to evaluate the clustering results.

A. Similarity Measures

Given a time series, defined as a sequence such as

$$X_n = (x_1, x_2, \cdots, x_N), \tag{1}$$

where x are the observations measured at different times n, we need to determine the level of *similarity/dissimilarity* (i.e. agreement/discrepancy) between a pair of them in order to cluster a sample of K time series.

Traditional distance computations, such as the Euclidean distance, can produce interesting results. However, in the case of *time series*, notions of distance need not to be confined to this simple geometric paradigm.

There are several ways to measure the dissimilarity between pairs of time series. This paper aims to cluster player profiles from *Age of Ishtaria* and *Grand Sphere* games based on their in-game behavior, hence dissimilarity measures were chosen according to this business target.

We are interested in the so-called *model-free* measures [20]. A naive *model-free* approach is to treat each series as an n-dimensional vector, and to calculate a shape-based geometric distance measure (without taking into account the absolute value of the time series selected). Such measures are the focus of our work, as we are interested in the *shape* pattern behavior (geometric comparison) rather than the magnitude of the time series.

Among the dissimilarity methods tested, those which provide the most robust results to classify time series are: Euclidean distance, Correlation (COR), Raw Values and Temporal Correlation (CORT) and Dynamic Time Warping (DTW). In addition, a *complexity-based* approach [20] called Complexity Invariant Distance (CID) [21] is applied. With this method, instead of focusing on the shape of the series, we expect to group profiles from a different perspective, taking into account the degree of variability over time.

Some other measures were also evaluated, e.g autocorrelation-based dissimilarity, Frechet distance measure, periodogram-based dissimilarity, among others (a review about these techniques can be found in [20]). However, these methods did not output as satisfactory results as the ones mentioned in the previous paragraph. The selected measures of interest are defined below, considering two time series X,Y of size T (which is the temporal dimension).

1) Dynamic Time Warping (DTW): DTW is a *non-linear* similarity measure obtained by minimizing the distance between two time series [22]. This method permits to group together series that have similar shape but out of phase [23]. Figure 3 shows in the right upper panel how DTW aligns series with delayed but similar patterns.

DTW distance can be expressed as

$$DTW(X,Y) = \min_{r \in M} \left(\sum_{m=1}^{M} |x_{im} - y_{jm}| \right), \tag{2}$$

where the path element $r = (i, j)$ represents the relationship between the two series. The goal is to minimize the time warping path r so that summing its M components gives the lowest measure of minimum cumulative distance between the time series. DTW searches for the best alignment between X and Y, computing the minimum distance between the points x_i and y_j.

2) Correlation-based measure (COR): It performs dissimilarities based on the estimated Pearson's correlation of two given time series. The COR computation can be expressed as

$$COR(X,Y) = \frac{\sum_{n=1}^{N}(x_n - \bar{X})(y_n - \bar{Y})}{\sqrt{\sum_{n=1}^{N}(x_n - \bar{X})^2}\sqrt{\sum_{n=1}^{N}(y_n - \bar{Y})^2}}. \tag{3}$$

3) Temporal Correlation and Raw Values Behaviors measure (CORT): It computes an adaptive index between two time series that covers both dissimilarity on raw values and dissimilarity on temporal correlation behaviors. It can be written as

$$CORT(X,Y) = \frac{\sum_{n=1}^{N-1}(x_{n+1} - x_n)(y_{n+1} - y_n)}{\sqrt{\sum_{n=1}^{N-1}(x_{n+1} - x_n)^2}\sqrt{\sum_{n=1}^{N-1}(y_{n+1} - y_n)^2}}. \tag{4}$$

4) Complexity-Invariant Distance measure (CID): CID computes the similarity measure based on the Euclidean distance but corrected by the complexity estimation of the series [21]. CID is written as

$$CID(X,Y) = dist(X,Y) \cdot CF(X,Y), \tag{5}$$

with CF being the complexity correction factor defined by

$$CF(X,Y) = \frac{max(CE(X), CE(Y))}{min(CE(X), CE(Y))}. \tag{6}$$

And $CE(\cdot)$ corresponds to the complexity estimations of a time series of length N, given by

$$CE(X) = \sqrt{\sum_{n=1}^{N-1}(x_n - x_{n+1})^2}. \tag{7}$$

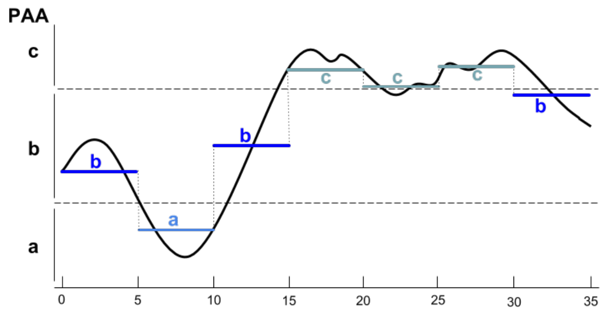

Fig. 1: Illustration of SAX representation method (dimensionality reduction of time series), performed with the following parameter values: $w = 7$ and $\alpha = 3$.

B. Representation Methods

Time series data collected from F2P games are high dimensional objects. In order to reduce their complexity and make the comparison feasible, it is convenient to perform a dimensional reduction transformation beforehand. There are several ways to reduce the data from n-dimensions to N-dimensions, and depending on the application domain, there are techniques more suitable than others. We focus on function approximation procedures to simplify the time series objects we aim to cluster. In the following subsections, we briefly summarize the most successful procedures for the video-game data tested in the present study.

1) Discrete Wavelet Transform (DWT): This method uses a wavelet decomposition to approximate the actual series. A wavelet is a function used to approximate the target time series by means of superposition of several (wavelet) functions. A *wavelet* object provides information about variations of the time series locally, as it can be shown in Figure 3 in the right lower panel. DWT assigns a coefficient to each *wavelet* component and the distance is computed between the wavelet-approximated time series.

2) Symbolic Aggregate Approximation related functions measure (SAX): SAX is a symbolic representation to simplify continuous time series [24]. The series is discretized and divided into sequential frames of equal size. Firstly, the series is divided in w set intervals and it is represented by its corresponding mean Piece wise Aggregate Approximation (PAA) dimensional reduction. Afterwards SAX is represented by a subset of alphabet letters of size α where $\alpha = (l_1, ..., l_\alpha)$ and the transformed series $\hat{X} = (\hat{X}_1, ..., \hat{X}_\alpha)$ is computed by determining equal-sized zones under a Gaussian distribution. The distance is then computed between the approximated time series. Figure 1 depicts a schematic view of the SAX dimensionality reduction method.

3) Trend Extraction: A time series is composed of different elements such as seasonal components, medium or long-term trend, cyclical movement (repeated pattern but non-periodic) or irregular fluctuations (also known as residuals). Depending on the clustering application, some of these components can be representative of the characteristics of the *raw* time series.

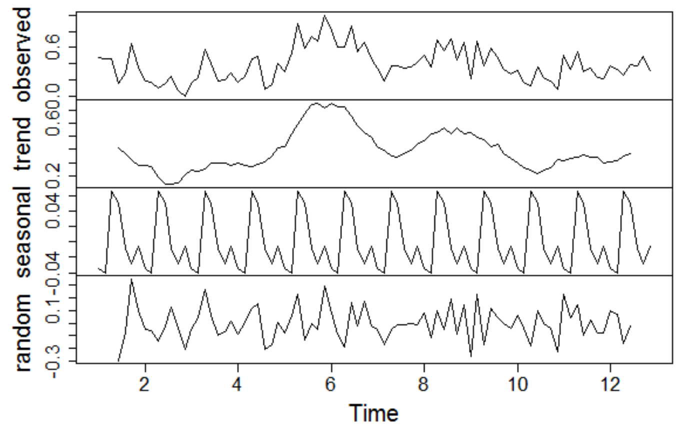

Fig. 2: Time Series Decomposition used for dimension reduction. Original series in the upper panel, trend extraction in the second plot, the seasonal and the random (residual fluctuations) components shown in the third and fourth panel, respectively.

As we are interested in characterizing players by their behavior, we focus on the trend component as it provides essential information for this purpose. Seasonality behavior or irregular data are not the center of our attention, the trend rather reflects significant information about player's interests. As it is mentioned in [25] "The trend of a time series is considered as a smooth additive component that contains information about global change". The method used in this work to extract the trend is the moving average filtering.

C. Hierarchical clustering

In this subsection, we describe the methods to create the nested partitions in order to classify the total number of time series.

Hierarchical clustering creates homogeneous partitions of data according to their level of dissimilarity, maximizing the difference between clusters [26], [27]. The clustering growing method can be increasing (agglomerative clustering or bottom-up) or decreasing (divisive clustering or top-down) at each step. It is normally represented by dendrograms that show the clustering levels in a tree-based graph. Figure 4 illustrates with a dendrogram the hierarchical clustering performed to classify player behavior of *Age of Ishtaria*.

The method selected to cluster the datasets studied in the current paper is *agglomerative clustering*. There are different methods of agglomerative clustering [26], but the one used for our analysis is the so-called *Ward method* which is a minimum variance technique. In Ward, the distance between two clusters is defined as the deviance between them. The clusters that are merged in the same group are the ones that lead to a minimum increase in the total within-cluster variance (calculated from the dissimilarity measure selected between the time series) [27]. This method is used to obtain the results presented in the Section V, as our goal is to obtain a low variance within the clusters.

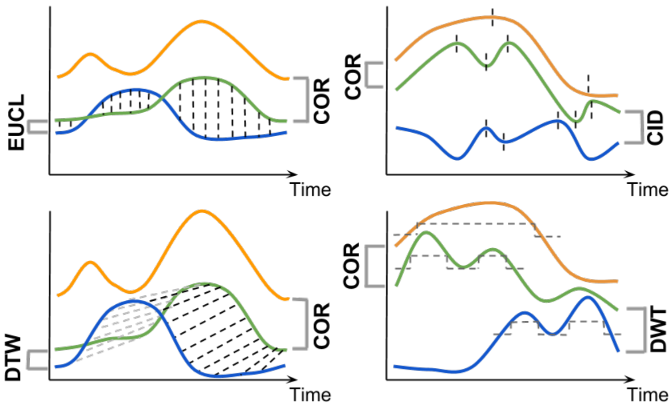

Fig. 3: Illustrations of the difference between time series clustering results obtained using Euclidean (EUCL) and Correlation dissimilarity measure (COR) (left upper panel), Dynamic Time Warping (DTW) in the upper right panel, Complexity Invariant Distance (CID) in the left lower panel, and Discrete Wavelet Transform (DWT) in the right lower panel.

III. Comparison of Clustering Methods

The selection of an adequate technique to cluster time series depends on the application and business interest. All the methods reviewed in Section II were tested to cluster the time series of game data. We want to classify players by pattern-shape, without giving too much attention to small fluctuations or to the total magnitude of the time series. Based on this aim we can conclude:

- DTW works particularly well to group similar player profiles with a shift on the time axis. DTW also groups together similar patterns but at different scale. However, as we are interested in evaluating the impact of game events on player activity, we rather focus on clustering synchronized profiles. Therefore, this is not the most suitable tool to measure the distance between time series for the purpose of the current study.
- In the DWT method of dimensionality reduction, the wavelets define the frequency of the series, which sometimes does not fit with the weekly seasonality we want to study.
- SAX representation method could be a useful tool for our problem as we are interested in identifying the pattern behavior, and not detailed aspects of the time series. However, the manual tuning of the two parameters w and a can be a drawback, although it is easily done once we introduce apriori information about the seasonality of the time series (the weekly events in our case). We hoped that the dimensionality reduction offered by SAX would allow us to cluster longer time series (2 months or more) but we did not obtain conclusive results.
- COR is a promising method for our goal. It groups similar geometric and synchronous profiles. As a drawback, COR seems to be sensitive to noise data and outliers (which are present in our datasets).
- CORT is similar to COR, but we ultimately obtained the most convincing results with the second.
- COR+trend is the combination of COR and trend extraction, which addresses COR's sensitivity to noise. This method allows us to obtain the best results for non-sparse time series (such as the time series of time played). Indeed, the trend extraction does not work well with time series containing many zero values (such as the time series of in-app purchases).
- CID groups series that have similar complexity patterns. This method performs poorly in classifying similar geometric profiles, which is what we do successfully with COR+trend applied to the time series of time played. However, it is the method that provides the best results when it comes to classifying time series containing large amount of zero values (such as the time series of purchases).

Figure 3 shows an intuitive comparison between similarity measures and representation methods to help to understand the difference between different techniques. The similarity methods and representation techniques described in this paper were tested to obtain the results presented in Section V.

A. Evaluation Metrics of Clustering results

The validation of the clustering methods is a challenging task, as we do not have any *truth* we can rely on to compare the accuracy of the classification, contrary to the supervised learning models. Several techniques to evaluate the adequacy of the similarity measures and representation methods to cluster time series were tested, among them: Dunn and average of Silhouette width [28], Normalized Hubert's statistic [29] and Entropy [30]. However, due to the difficulty of the task and the high complexity of time series objects, the results were not satisfactory. We used several kinds of visualization techniques to validate the clustering results and to determine the optimal number of clusters.

IV. Datasets

A. Data Source

Our data come from the games *Age of Ishtaria* and *Grand Sphere* by Silicon Studio.

We worked with time series of the following variables that are game independent and can be measured in all free-to-play games. These variables are measured per user and per day.

- *Time*: The amount of time spent in the game
- *Sessions*: The total number of playing sessions
- *Actions*: The total number of actions performed
- *Purchase*: The total amount of in-app purchases

Time, sessions and actions time series are highly correlated and produce very similar results. Thus, for the purpose of our study, we focus on the Time variable, which has a lower measurement error.

Purchases time series are different from the others because they are sparse (they contain many zero values), as the majority of the paying users do not complete an in-app purchase every day.

TABLE I: Summary description of the clustering results with *Age of Ishtaria* and *Grand Sphere* data.

Clustering result name	*Game*	*Data*	*Technique*	*Clusters*	*Start date*	*End date*
Age of Ishtaria time clustering	Age of Ishtaria	Time played per day	COR+trend	8	11-Jun-2015	1-Jul-2015
Age of Ishtaria spending clustering	Age of Ishtaria	Spending per day	CID	5	30-Jan-2015	19-Feb-2015
Grand Sphere time clustering	Grand Sphere	Time played per day	COR+trend	8	11-Sept-2015	1-Oct-2015

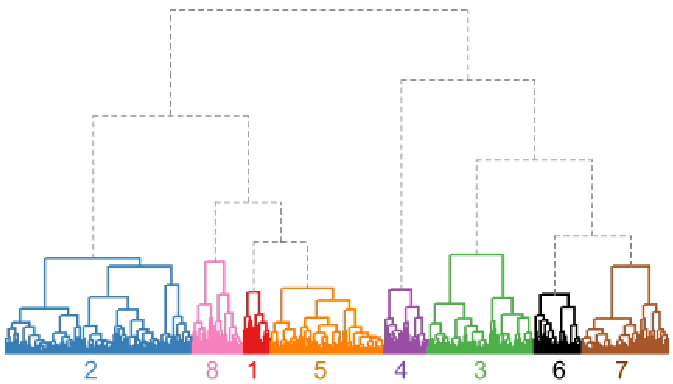

Fig. 4: Hierarchical Clustering represented by a dendrogram of *Age of Ishtaria* time-played data. The Age of Ishtaria time clustering is performed with COR similarity measure and trend extraction as representation method.

B. Time Series Studied

The frequency of our time series is daily. We study them on a weekly basis since there are weekly game events influencing player behavior. The studied period P, therefore contains $N_{week} \times 7$ values, with $N_{week} = 3$ being the number of weeks selected for this study. We synchronize the starting date P_{start} and the ending date P_{end} of our time series with the starting date of the game events.

In order to avoid a bias due to the partial absence of data from players who join or leave the game during P, we only consider data from players who installed the game before P_{start} and who are still active after P_{end}.

For the final results of clustering the time series of Time played, we consider only data from the users who played at least 6 days per week. There are two reasons for this choice. Firstly, from a free-to-play game developer perspective, we are interested in the most active players. Secondly, the clustering technique that allowed us to obtain the best results for this clustering performs poorly with sparse time series.

For the final results of clustering the times series of Purchases, which are mostly sparse, except for the very top spenders, we considered the players who did at least one purchase during P. Due to the sparse nature of these time series, we then obtain the best results using a different clustering technique.

Finally, we take random samples of 1000 time series respecting the conditions above for our experiments.

V. Results

In this section, we present the results of the clustering experiments summarized in Table I. For each experiment, we call event A, B and C the game events released respectively on week 1, 2 and 3. This is a naming convention independent of the content of the events.

A. Clustering and Visualization

1) Clustering time series of time played: Figure 5 visualizes the classification obtained by clustering time series of time played per day by Age of Ishtaria players, using the correlation dissimilarity measure on the trend extracted from the raw time series. We call this method *COR+trend*.

For each cluster, we plot the mean of the time series and a heatmap containing all the time series (one time series for each player included in the sample).

Visualizing the mean allows to see the top trends in player behavior. For example, the activity of class 3 plunged for event B but spiked for event C, while class 4 followed an opposite pattern. Since the game events are usually designed based on predefined game templates (i.e. they are reused throughout the lifetime of the game), this analysis helps the game designer to better understand the interest of several groups of players in different kinds of game events. This supports future game event planning and improves the knowledge about the impact of the game events on the player activity.

Visualizing the time series of each cluster on a heatmap allows to quickly validate the quality of the clustering. Figure 5 shows that the time series follow the same patterns within each cluster.

Heatmap along with a dendrogram visualization, represented in Figures 4 and 5, proves to be a better tool than the statistical measures tested, mentioned in Section III-A, for choosing the optimal number of clusters. Using these tools we determine that the most optimal clustering is obtained with 8 clusters.

We apply the same clustering technique on time series of time played by the players of Grand Sphere, and obtain similar results, as it can be checked in Figure 6. This is a promising fact towards obtaining an adequate technique ready to cluster data from other F2P games.

2) Clustering time series of purchases: Figure 7 depicts the clusters obtained by clustering time series of purchases per day by Age of Ishtaria players, using the Complexity Invariant Distance (CID) on the raw time series.

For each cluster, we represent the distribution of the data separated per week in a box-plot, and a heatmap containing all the time series.

Visualizing the time series of each cluster on a heatmap allows to distinguish different purchase patterns. For example, players from class 1 and class 3 purchase sparsely while players from class 2 purchase nearly every day.

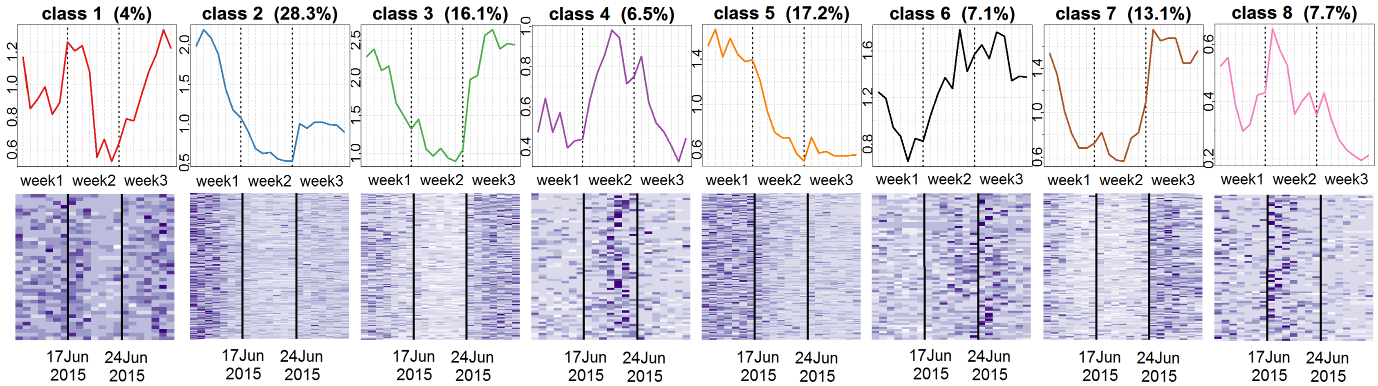

Fig. 5: Mean of the time series and heatmap for each cluster from Age of Ishtaria time clustering (time played per day). Vertical lines delimiting the game events. Clustering performed with COR similarity measure and trend extraction.

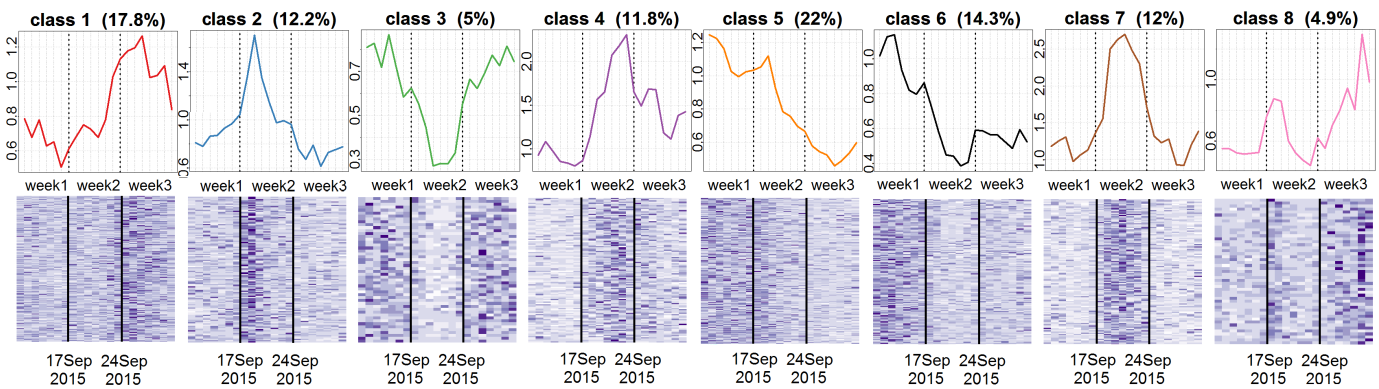

Fig. 6: Mean of the time series and heatmap for each cluster from Grand Sphere time clustering (time played per day). Vertical lines delimiting the game events. Clustering performed with COR similarity measure and trend extraction.

TABLE II: Characteristics of the players at the starting date of the studied period (Age of Ishtaria time clustering)

variables	*class 1*	*class 2*	*class 3*	*class 4*	*class 5*	*class 6*	*class 7*	*class 8*
number of players	40	283	161	65	172	71	131	77
ratio PU	30.0%	33.2%	44.7%	20.0%	33.1%	33.8%	35.1%	14.3%
average level	47	53	75	35	51	49	58	31

TABLE III: Cumulative churn ratio in the months following the clustering, after period P (Age of Ishtaria time clustering)

churners ratio	*class 1*	*class 2*	*class 3*	*class 4*	*class 5*	*class 6*	*class 7*	*class 8*
July	15.0%	11.7%	4.3%	15.4%	19.2%	18.3%	5.3%	22.1%
August	27.5%	19.8%	13.0%	26.2%	30.8%	31.0%	14.5%	28.6%
November	45.0%	48.4%	29.2%	47.7%	51.7%	20.7%	32.8%	58.4%

Since the scale of each heatmap is normalized separately to be able to visualize properly the full range of purchases on each heatmap, we can not compare the amount of the purchases between the clusters using only this visualization. And, contrary to the clustering of the time series of time played described above, the time series of purchases are mostly sparse, which makes it irrelevant to plot the mean of these time series. That is why we use the box-plot representation of the spending for each week, in order to visualize the difference of scale between the different groups. This additional plot allows us, for example, to see that class 5 contains very high spenders even if they have a relatively sparse purchase behavior like class 1 and 3.

As for the time series of time played, we used the heatmap visualizations and the dendrogram to choose to use $k = 5$ clusters.

B. Extraction of Players Characteristics

We are not only interested in clustering game users and discovering hidden patterns, but also want to analyze the characteristics they have in common.

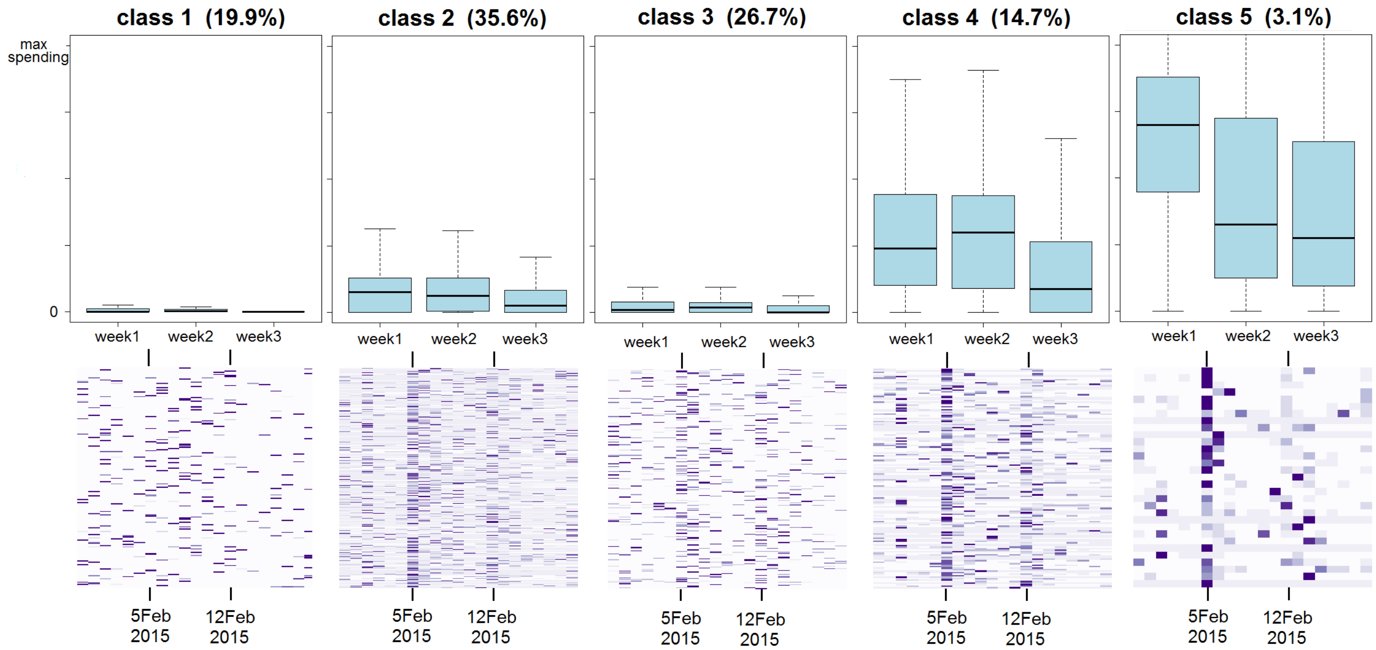

Fig. 7: Clustering results from the visualization of purchase's time series from Age of Ishtaria data using CID similarity measure (Age of Ishtaria spending clustering). Box plots of the spending per player and per week in the upper panel. Corresponding heatmap for each cluster below. The dates on the x-axis delimiting the weekly game events.

After performing the clustering, we measure how players behave during the period P by analyzing their characteristics on the start date P_{start} of the time series.

Table III reflects that class 3 contains players with the highest playing levels and also the highest ratio of paying users, while classes 4 and 8 contain the players with the lowest levels and the lowest ratios of paying users.

It is interesting to note that these clusters coincide with the ones already discussed earlier as reflecting an opposite interest in certain game events. With these two observations, we can conclude that event B was unpopular for advanced players and more popular for less advanced players, and that event C was more popular for advanced players and unpopular for less advanced players. A game planner visualizing this could conclude that she had better avoid triggering an event of event C's type soon after a user acquisition campaign, as it would likely be unpopular for the new coming less advanced players just acquired.

This example shows that it is possible to extract differentiating player characteristics from the clustering we obtained.

C. Churn behavior

Player retention is of crucial importance in F2P games. Several models have been proposed to help to understand and predict the churn of players [31], [32], [33].

Based on the results obtained in Table II and Figure 5, we study the evolution of the players after the period of time P covered by the time series, in order to see if there is a relation between their behavior during P and after P.

Table II shows the churning rate 1, 2 and 5 months after the period P for each cluster. We observe that class 3 and 7 have a significantly lower churning rate than class 4 and 8, being 3 to 4 times lower after 1 month and 1.5 to 2 times lower after 5 months.

According to this result, players have a different churn behavior following their profile classification performed during the period P.

Therefore, the use of the unsupervised classification of player profiles suggested in this article could be an interesting feature to address the temporal dynamics of players data for a churn supervised learning model. In [32] an alternative approach was proposed using a Hidden Markov Model.

However, in order to use this predictor in a supervised model some changes need to be performed in the definition of the problem as we discussed in Section IV. This comprehensive analysis is beyond the scope of this paper. For example, this would involve to cluster players based on their last weeks behavior, e.g. the time series starting date would be 3 weeks before the last day the players connected to the game instead of taking fixed dates as in the present work.

This time series classification would allow us to improve the understanding about the churn of players but, on the other hand, it would not provide information about game events reaction, which is a principal target of the current analysis.

VI. Summary and Conclusion

In the present article, we have conducted a research about unsupervised clustering of time series data from two free-to-play games. We evaluate several similarity measures and representation methods to extract meaningful behavioral patterns of players. This allows us to assess the impact of weekly game events and discover hidden playing dynamics regarding purchases and time played per day. An appropriate characterization of time series allows us to find significant attributes in common among players belonging to the same group. Ongoing and future work involve the application of the time series clustering results to churn prediction models and further analysis of the player profiles.

VII. Software

The analyses presented in Section V were performed with the R version 3.2.3 for Windows, using the following packages from CRAN: *TSclust* 1.2.3 [20], *timeSeries* 3022.101.2 [34], *fpc* 2.1-10 [35], *Rmisc* 1.5 [36], *reshape* 0.8.5 [37], *ggplot2* 2.0.0 [38].

Acknowledgments

We thank our colleagues Sovannrith Lay, Hiroshi Okuno, Takeshi Kimura, Tomomi Hamamura, Kotaro Narizawa and Yumi Kida for their help to collect the data and their support during this study. We also thank Thanh Tra Phan for the careful review of the article.

References

[1] T. Fields, "Mobile & social game design: Monetization methods and mechanics," *CRC Press*, 2014.

[2] A. Annie, "IDC. 2014. Mobile App Advertising and Monetization Trends 2012-2017: The Economics of Free."

[3] W. Xing, "Unleash the Power of Events to Drive Revenue in Games," Mar. 2014. [Online]. Available: http://www.gamasutra.com/blogs/XingWang/20140319/213429/Unleash_the_Power_of_Events_to_Drive_Revenue_in_Games.php

[4] E. Fradley-Pereira, "A Beginners Guide to Player Segmentation," May 2015. [Online]. Available: https://www.fusepowered.com/blog/2015/05/a-beginners-guide-to-player-segmentation/

[5] C. Bauckhage, A. Drachen, and R. Sifa, "Clustering game behavior data," *Computational Intelligence and AI in Games, IEEE Transactions on*, vol. 7, no. 3, pp. 266–278, 2015.

[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*. IEEE, 2012, pp. 163–170.

[7] A. Drachen, C. Thurau, R. Sifa, and C. Bauckhage, "A comparison of methods for player clustering via behavioral telemetry," *CoRR*, vol. abs/1407.3950, 2014. [Online]. Available: http://arxiv.org/abs/1407.3950

[8] R. Sifa, C. Bauckhage, and A. Drachen, "The Playtime Principle: Large-scale cross-games interest modeling," in *2014 IEEE Conference on Computational Intelligence and Games, CIG 2014, Dortmund, Germany, August 26-29, 2014*, 2014, pp. 1–8. [Online]. Available: http://dx.doi.org/10.1109/CIG.2014.6932906

[9] A. Drachen, R. Sifa, C. Bauckhage, and C. Thurau, "Guns, swords and data: Clustering of player behavior in computer games in the wild." in *CIG*. IEEE, 2012, pp. 163–170.

[10] H. D. Menéndez, R. Vindel, and D. Camacho, "Combining time series and clustering to extract gamer profile evolution," in *Computational Collective Intelligence. Technologies and Applications*. Springer, 2014, pp. 262–271.

[11] T. Fu, "A Review on Time Series Data Mining," *Eng. Appl. Artif. Intell.*, vol. 24, no. 1, pp. 164–181, Feb. 2011. [Online]. Available: http://dx.doi.org/10.1016/j.engappai.2010.09.007

[12] R. Agrawal, C. Faloutsos, and A. N. Swami, "Efficient similarity search in sequence databases," in *Proceedings of the 4th International Conference on Foundations of Data Organization and Algorithms*, ser. FODO '93. London, UK, UK: Springer-Verlag, 1993, pp. 69–84. [Online]. Available: http://dl.acm.org/citation.cfm?id=645415.652239

[13] P. Esling and C. Agon, "Time-series Data Mining," *ACM Comput. Surv.*, vol. 45, no. 1, pp. 12:1–12:34, Dec. 2012. [Online]. Available: http://doi.acm.org/10.1145/2379776.2379788

[14] H. Ding, G. Trajcevski, P. Scheuermann, X. Wang, and E. Keogh, "Querying and Mining of Time series Data: Experimental Comparison of Representations and Distance Measures," *Proc. VLDB Endow.*, vol. 1, no. 2, pp. 1542–1552, Aug. 2008. [Online]. Available: http://dx.doi.org/10.14778/1454159.1454226

[15] X. Xi, E. Keogh, C. Shelton, L. Wei, and C. A. Ratanamahatana, "Fast Time Series Classification using Numerosity Reduction," in *Proceedings of the 23rd International Conference on Machine Learning*, ser. ICML '06. New York, NY, USA: ACM, 2006, pp. 1033–1040. [Online]. Available: http://doi.acm.org/10.1145/1143844.1143974

[16] T. W. Liao, "Clustering of time series data - a survey," *Pattern recognition*, vol. 38, no. 11, pp. 1857–1874, 2005.

[17] X. Wang, K. Smith, and R. Hyndman, "Characteristic-Based Clustering for Time Series Data," *Data Mining and Knowledge Discovery*, vol. 13, no. 3, pp. 335–364, 2006. [Online]. Available: http://dx.doi.org/10.1007/s10618-005-0039-x

[18] E. Keogh and S. Kasetty, "On the need for Time Series Data Mining Benchmarks: A Survey and Empirical Demonstration," *Data Min. Knowl. Discov.*, vol. 7, no. 4, pp. 349–371, Oct. 2003. [Online]. Available: http://dx.doi.org/10.1023/A:1024988512476

[19] E. Keogh, "A Decade of Progress in Indexing and Mining Large Time Series Databases," in *Proceedings of the 32Nd International Conference on Very Large Data Bases*, ser. VLDB '06. VLDB Endowment, 2006, pp. 1268–1268. [Online]. Available: http://dl.acm.org/citation.cfm?id=1182635.1164262

[20] P. Montero and J. A. Vilar, "TSclust: An R package for time series clustering," *Journal of Statistical Software*, vol. 62, no. 1, pp. 1–43, 2014. [Online]. Available: http://www.jstatsoft.org/v62/i01/

[21] G. E. Batista, E. J. Keogh, O. M. Tataw, and V. M. de Souza, "CID: an efficient complexity-invariant distance for time series," *Data Mining and Knowledge Discovery*, vol. 28, no. 3, pp. 634–669, 2014.

[22] D. J. Bemdt and J. Clifford, "Using Dynamic Time Warping to Find Patterns in Time Series," 1994.

[23] C. A. Ralanamahatana, J. Lin, D. Gunopulos, E. Keogh, M. Vlachos, and G. Das, "Mining time series data," in *Data Mining and Knowledge Discovery Handbook*. Springer, 2005, pp. 1069–1103.

[24] J. Lin, E. Keogh, L. Wei, and S. Lonardi, "Experiencing SAX: a novel symbolic representation of time series," *Data Mining and knowledge discovery*, vol. 15, no. 2, pp. 107–144, 2007.

[25] T. Alexandrov, S. Bianconcini, E. Dagum, P. Maass, and T. McElroy, "A review of some modern approaches to the problem of trend extraction," *Econometric Reviews*, vol. 31, no. 6, pp. 593–624, 2012.

[26] F. Murtagh and P. Contreras, "Algorithms for hierarchical clustering: an overview," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 2, no. 1, pp. 86–97, 2012.

[27] F. Murtagh and P. Legendre, "Ward's Hierarchical Agglomerative Clustering Method: Which algorithms implement Ward's Criterion?" *Journal of Classification*, vol. 31, no. 3, pp. 274–295, 2014. [Online]. Available: http://dx.doi.org/10.1007/s00357-014-9161-z

[28] B. Desgraupes, "Clustering Indices," *University of Paris Ouest-Lab ModalX*, vol. 1, p. 34, 2013.

[29] M. Halkidi, Y. Batistakis, and M. Vazirgiannis, "On clustering validation techniques," *Journal of intelligent information systems*, vol. 17, no. 2, pp. 107–145, 2001.

[30] M. Meilă, "Comparing clusterings - an information based distance," *Journal of multivariate analysis*, vol. 98, no. 5, pp. 873–895, 2007.

[31] F. Hadiji, R. Sifa, A. Drachen, C. Thurau, K. Kersting, and C. Bauckhage, "Predicting player churn in the wild," in *Computational Intelligence and Games (CIG), 2014 IEEE Conference on*. IEEE, 2014, pp. 1–8.

[32] J. Runge, P. Gao, F. Garcin, and B. Faltings, "Churn prediction for high-value players in casual social games," in *Computational Intelligence and Games (CIG), 2014 IEEE Conference on*. IEEE, 2014, pp. 1–8.

[33] A. Periáñez, A. Saas, A. Guitart, and C. Magne, "Churn prediction in mobile social games: towards a complete assessment using survival ensembles," *Submitted to DSAA*, 2016.

[34] D. Wuertz and Y. Chalabi, "timeSeries: Rmetrics-financial time series objects," *R package version 2100*, 2009.

[35] "fpc: Flexible Procedures for Clustering."

[36] R. Hope, "Rmisc: Ryan miscellaneous. R package version 1.5," 2013.

[37] H. Wickham, "reshape: Flexibly reshape data, 2007," *R package version 0.8.0*.

[38] H. Wickham and W. Chang, "ggplot: An implementation of the Grammar of Graphics," *R package version 2.0.0*, 2013.