Proceedings, The Twelfth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE-16)

A Generic Approach for Player Modeling Using Event-Trait Mapping and Feature Weighting

M. Akif Gunes, Gokhan Solak
Artificial Intelligence and Robotics Laboratory
Computer Engineering Department
Istanbul Technical University
Istanbul, Turkey

Ugur Akin, Omer Erden
Triodor Ar-Ge
Istanbul, Turkey

Sanem Sariel
Artificial Intelligence and Robotics Laboratory
Computer Engineering Department
Istanbul Technical University
Istanbul, Turkey

Abstract

There are a wide variety of studies on player modeling. However, most of these studies target a specific game or genre. In some of these works, the number of in-game actions is used as a feature for modeling a player. However, using this feature leads to a complex model, and the model may miss some high-level relations among actions. In this paper, we propose a generic player modeling method that uses action-trait mapping relations which reveal correlations among actions. Mapping from the action-space to a much smaller trait-space improves interpretability of models. Additionally, to use the differences of impact of actions on player models, we apply feature weighting which uses the inverse of action frequencies. Players are clustered by Expectation Maximization. We demonstrate our method on a casual mobile game, Dusk Racer. We evaluate the feature weighting method using cluster validation with internal criteria. We conclude that using traits and feature weighting improves clustering quality and usability of the player model.

Keywords: Player Modeling, Feature Weighting, Clustering, Game Analytics.

Introduction

Player modeling enables many novel applications on games to increase user satisfaction. These applications vary from tailoring a game during development to adapting it during runtime. Whichever the application is, modeling the player is the first step to proceed. We aim to present a generic approach to player modeling which can be used in such applications. We particularly validate our approach on a casual mobile game, Dusk Racer of which screenshots are illustrated in Figure 1.

Creating a generic approach is challenging because there are various types of games. In every game, different aspects of behaviours and motivations of players may emerge. Finding the relevant set of features describing these characteristics becomes necessary. One approach to satisfy this need is to reduce the features to basic game actions (Etheredge, Lopes, and Bidarra 2013). This method is based on the motivation that players play based on specialized patterns. Although this is correct in theory, when the number of actions increases, extracting distinctive patterns may become hard. Another shortcoming of the pure action-based modeling method is that it can miss some high-level relations among actions. More than one action can be indicators of the same player behaviour. One action can support more than one player behaviour . For this reason, merging actions under high-level events is not enough to catch all these relations.

Figure 1: Screenshots from game Dusk Racer.

We use trait modeling to address the shortcomings of pure action-based models. Traits can be thought as classes for player behaviours. In-game events are used as features to model trait scores of players. An event can be a high-level combination of game actions or an indirect behaviour such as an achievement or a menu interaction and is defined to capture the context. Every event is mapped to one or more traits. These mappings are called trait characteristics and they are assessed by domain experts' intuition to reflect correlations among events. In addition to trait characteristics, our mapping contains a scoring of event strengths for each trait. With this addition, our approach enables a more accurate model compared to the previous trait-based approaches . Since traits are designed to model the essence of player motivations and personalities, the models produced by an action-trait mapped system become less complex due to abstraction. We created and used our own trait set for our experiments; however, the method is not limited to this trait set. We believe that a generic method should be independent from a particular trait theory, until (if ever) a generic trait theory is widely accepted.

In addition to event-trait mapping, we integrate *feature weighting* into our method. Since game events appear in many different forms, from frequent simple events to very rare but complex events, their impacts on models must be taken into account differently. This condition has been given far less importance in existing player modeling studies than it deserves. We apply a feature weighting method, as the inverse of frequency of an action and validate its contribution to the effectiveness in modeling.

After acquiring the game events and pre-processing them using their mappings to traits and feature weighting, we obtain the trait scores for each user. Users are clustered using an Expectation Maximization (EM) algorithm with the desired number of clusters. Clusters represent models of players, ready to be analysed by the game designer.

The contributions of this paper are three fold: (1) We present a generic approach for player modeling (2) Our event-trait relation model is used as a novel method to represent event correlations. We have a two-way mapping between traits and actions unlike previous methods (i.e., from traits to actions and from actions to traits). (3) the importance of feature weighting on player modeling is validated.

In the following sections we present the background studies which form the foundations and motivations of this work. After that we present the importance of feature weighting. Then in the next section, we explain our method in detail. We present our experimental results on a mobile game, Dusk Racer (Fig. 1) and provide a discussion. Finally, we conclude the paper with the possible future works.

Background

Player modeling is used for many purposes. It can help to understand if the game is played as it is intended or to discover different player dynamics (Tychsen and Canossa 2008). Players can be clustered to better understand their motivations (Drachen, Canossa, and Yannakakis 2009; Etheredge, Lopes, and Bidarra 2013). Furthermore, games can be personalized using player models to adapt game behaviour and game content for many purposes (Bakkes, Tan, and Pisan 2012; Charles and Black 2004). For example, Missura and Gärtner (2009) adjust game difficulty to increase player retention. Harrison and Roberts (2011) apply collaborative filtering to recommend new game content. Thue, Bulitko, and Spetch (2008) guide the storyline by considering motivations of the player. Bakkes, Spronck, and Van Den Herik (2009) model the player to shape game AI strategy. Borbora et al. (2011) and Runge et al. (2014) predict the players who will abandon the game, to take precautionary actions.

There are various methods used to model players. Some methods use a theoretical basis to model players while some are purely data-driven. Sources of information about player types can be *in-game*(online) player behaviour or *off-game*(offline) questionnaires and identities (Charles and Black 2004).

Using questionnaires is a method of observing player typology features, that is player identification and player personality elements (Bateman, Lowenhaupt, and Nacke 2011). Results of player typology questionnaires can be used as input features for player models (Charles and Black 2004; Monterrat et al. 2015). However, questionnaires may be costly, and their validity is debatable (Tychsen and Canossa 2008).

Using game actions as indicators of player type is a common approach for player modeling. Pure data-driven approaches use actions alone to group players (Etheredge, Lopes, and Bidarra 2013; Drachen, Canossa, and Yannakakis 2009; Runge et al. 2014). However, results of pure data-driven approaches may become hard to interpret for a large set of features. Data-driven approach can be used together with player typology theories to improve modeling results. Borbora et al. (2011) uses an existing player motivation theory to select a subset of game actions. They compare this approach with a data-driven approach and conclude that using theory decreases the complexity. Using higher level abstractions is another way of making game data more interpretable (Tychsen and Canossa 2008). Monterrat et al. (2015) uses existing trait theories to map game-features to trait-spaces. Although their approach is similar to ours as game features are mapped to some traits, our method is different by that we also include weightings among game actions to compare their impacts on respective traits.

Some of the action-based methods (Zook et al. 2012; Kilic, Gunes, and Sariel 2016) use the order of actions as a feature in addition to their counts . This allows to discover relevant temporal relations. However, these types of features do not properly represent some asynchronous events. For example, players can pause the game any time to check menu options which may break temporality.

Theories used for guiding player models are results of player typology research. Bartle's classification of MUD (Multi-user Dungeon) players (Bartle 1996) is one of the earlier works in classification of player typologies. Yee (2006) conducts surveys with massively multi-player online role playing game (MMORPG) players. Analysis of these surveys leads to a set of motivations which partially confirm and extend Bartle's types. Bateman, Lowenhaupt, and Nacke (2011) suggest candidates for a future theory of play, moving from psychometric theories. Nacke, Bateman, and Mandryk (2014) combine the existing theoretical types with neurobiological insights. Ferro, Walz, and Greuter (2013) analyse the relations between existing psychometric theories, play theories and game mechanics. They group more than 30 player types and traits in the existing theories into 5 classes.

Methodology

We aim to obtain in-game personality traits of players using in-game events. Therefore, each game event is associated with one or more traits. The number of players, events and traits are denoted by p, e, t, respectively. We create 9 traits ($t = 9$) inspired from Belbin's team roles (Belbin 2004) and studying the existing player trait theories (Ferro, Walz, and Greuter 2013). These traits are *Explorer*, *Meticulist*, *Competitor*, *Compulsivist*, *Strategist*, *Hoarder*, *Loner*, *Social* and *Exploiter*:

- An *Explorer* likes to explore new areas and interactions.
- A *Meticulist* understands the game mechanics and plays the game as the designer expects.
- A *Competitor* plays the game to win against a game-bot or another player.
- A *Compulsivist* acts on impulse by completing game mechanics as they appear.
- A *Strategist* utilizes the game mechanics in a smart way to achieve better outcomes.
- A *Hoarder* enjoys positive impulses of picking up all in-game prizes.
- A *Loner* does not enjoy social interaction. They may share games on social media but do not use *in-game* social interaction mechanics.
- A *Social* uses all social interaction mediums presented in or outside of a game session.
- An *Exploiter* exploits a failure in the game design that affects the way players interact with the game mechanics.

Some important matrices are defined to represent relations in our methodology.

UT (size $p \times t$): This matrix represents user-trait relations. Each row contains trait scores belonging to a player. The rows correspond to players' scores for each related trait given in columns. This matrix is the result of a series of operations described in this section. UT is then given to the clustering algorithm as input.

UE (size $p \times e$): User-event matrix contains information on the number of game events done by players. An event can be a high-level combination of game actions or an indirect behaviour such as an achievement or a menu interaction.

EW (size $e \times e$): Event-weight matrix is a diagonal matrix containing the importance weights of game events. This matrix is used to apply feature weighting on events. It can be filled by a domain expert or can be extracted from the game data. In our case study, it is obtained by the latter approach. We made an assumption that the impact of each event is inversely proportional to its count.

ET and TE shows a matrix size of $e \times t$ that represents event-trait relations. Finally, the generic formula given in Equation (1) is used to obtain a user-trait score matrix.

$$UT = normalize(UE \times EW) \times (ET \circ TE) \tag{1}$$

Feature weighting is applied using EW matrix. In many real world applications, there are correlations among attributes in raw data sets. If raw data sets are used directly, clustering results are not reasonable (Song et al. 2007). To overcome this problem, feature weighting techniques can be used as a preprocessing method in data mining. When applied to features, importance of features are discriminated. One of the criteria used in feature weighting is inverse of frequency (Yu et al. 2003). The goal of this criterion is to decrease the impact of features which occur in high frequencies. The diagonal value EW_{ii} is the inverse frequency of the event E_i . This value is calculated as in Equation (2).

$$EW_{ii} = \left(\frac{C_i}{\sum\limits_{E_j} C_j} \right)^{-1} \tag{2}$$

where C_j is the total count of E_j events done by all players.

$normalize$: It shows normalization processes. Normalization is an important issue in behaviour analysis. Mixing data types as behavioural features are handled in various ways (Bauckhage, Drachen, and Sifa 2015). The purpose of the multiplication $UE \times EW$ is to find how much weight is computed for each event type and user pair. To achieve a better clustering, we normalize it for each user (row-based %). After this, column-based feature normalization is done as in Equation (3).

$$x_{new} = \frac{x - X_{min}}{X_{max} - X_{min}} \tag{3}$$

Here, x is the scaled feature value. X_{min} and X_{max} are the minimum and the maximum values of that feature in the whole dataset. x_{new} is the scaled value of x. Unlike others, in our method, normalization is done by considering the weights of events.

ET : Event-Trait matrix (events to traits). This matrix represents how much an event is related to each trait. It can be formed by two approaches. It can be directly assigned by the game designer or it can be indirectly inferred from the result of a questionnaire which contains questions to compare a given event with the predefined event types. In this work, the relation of events with each trait is directly determined with expert knowledge. When scoring an event, it is considered isolated from the other events. Second approach is left as a future works.

Scoring is done considering the question *"Which traits are indicated by this event?"*. A score between $0-5$ is given where 0 means *no-relation* and 5 means *high-relation*.

TE : Event-Trait matrix (traits to events). This matrix shows how much a trait is related to an event. It is different than ET by the direction of the relation.

$ET \circ TE$: This element-wise product (or Hadamard product) produces an event-trait matrix balanced both horizontally and vertically. Here, the values vary between $0-25$. An illustration of how the formation of this matrix is given below.

$$\begin{array}{ccc} & T_1 & T_2 \\ E_1 & 1 & 5 \\ E_2 & 5 & 3 \\ E_3 & 0 & 1 \end{array} \circ \begin{array}{ccc} & T_1 & T_2 \\ E_1 & 5 & 1 \\ E_2 & 1 & 5 \\ E_3 & 0 & 3 \end{array} = \begin{array}{ccc} & T_1 & T_2 \\ E_1 & 5 & 5 \\ E_2 & 5 & 15 \\ E_3 & 0 & 3 \end{array} \tag{4}$$

After the related matrices are computed and the preprocessing is done as described above, we apply clustering to group players. The input of the clustering algorithm is the trait scores of each player. We use Expectation Maximization for clustering players.

$$
\begin{array}{ccccc}
 & HC & SL & FS & SO \\
U_1 & 292 & 2 & 2 & 2 \\
U_2 & 864 & 1 & 1 & 33 \\
U_3 & 289 & 5 & 3 & 4 \\
U_4 & 1010 & 1 & 2 & 82
\end{array}
\xrightarrow{row\ based\,\%}
\begin{array}{cccc}
HC & SL & FS & SO \\
97.98 & 0.67 & 0.67 & 0.67 \\
96.10 & 0.11 & 0.11 & 3.67 \\
96.01 & 1.66 & 0.99 & 1.32 \\
92.23 & 0.09 & 0.18 & 7.48
\end{array}
\xrightarrow{normalize}
\begin{array}{cccc}
HC & SL & FS & SO \\
1 & 0.36 & 0.63 & 0 \\
0.67 & 0.01 & 0 & 0.43 \\
0.65 & 1 & 1 & 0.09 \\
0 & 0 & 0.08 & 1
\end{array}
$$

Figure 2: Preprocessing without feature weighting. U_1-U_4 represent players. Columns represent events; HC: Hitting to the other cars, SL: Looking at the leaderboard screen, FS: Facebook share, SO: Closely overtaking other cars.

$$
\xrightarrow{feature\ weighting\%}
\begin{array}{ccccc}
 & HC & SL & FS & SO \\
U_1 & 272.96 & 510 & 573.75 & 37.93 \\
U_2 & 807.69 & 255 & 286.87 & 625.90 \\
U_3 & 270.16 & 1275 & 860.62 & 75.86 \\
U_4 & 944.17 & 255 & 573.75 & 1555.28
\end{array}
\xrightarrow{row\ based\,\%}
\begin{array}{cccc}
HC & SL & FS & SO \\
19.57 & 36.56 & 41.13 & 2.71 \\
40.88 & 12.90 & 14.52 & 31.68 \\
10.88 & 51.37 & 34.67 & 3.05 \\
28.36 & 7.66 & 17.23 & 46.73
\end{array}
\xrightarrow{normalize}
\begin{array}{cccc}
HC & SL & FS & SO \\
0.28 & 0.66 & 1 & 0 \\
1 & 0.12 & 0 & 0.65 \\
0 & 1 & 0.75 & 0.00 \\
0.58 & 0 & 0.10 & 1
\end{array}
$$

Figure 3: Preprocessing with feature weighting. U_1-U_4 represent players. Columns represent events as in Figure 2.

A Case Study on the Dusk Racer Game

In the scope of our study, a casual mobile game, the Dusk Racer, is selected as the benchmark platform to implement our approach. It is a single player racing game. Two screenshots of the game are shown in Figure 1.

Players use the steering wheel at the bottom of the screen to drive a car while trying to avoid crashes and gain virtual money and a higher score by executing actions. There are 25 kinds of events in this game that we use in our study. While some of the events directly count game mechanics, some of them consist of count of skipping some game mechanics. Players can use Facebook or Twitter share buttons to share their scores or they can explore game menus.

After defining in-game events and traits for the given game, ET and TE can be created as mentioned in Methodology Section. Before the clustering phase, game data are preprocessed. Our approaches of preprocessing with and without feature weighting are shown in Figure 2 and Figure 3, using the following game events:

- HIT_CAR (HC): Hitting to the other cars
- SCREEN_LEADERBOARD (SL): Looking at the leaderboard screen
- FACEBOOK_ATTEMPT_SHARE (FS): Facebook share
- SCORE_OVERTAKE (SO): Obtaining scores by closely overtaking the other cars resulting in a 3x combo or higher

The HC event is a frequent event in this game. However, amateur players do this action more frequently than the others. SL action is generally done by players with *explorer* and *competitor* traits. FS action is preferred by *social* players. SO action is done by experienced players from *strategist* and *competitor* traits.

In the first matrix of Figure 2, the number of actions in four action types are given for four players. In the second step, horizontal normalization is done by calculating the percentage of actions over the total actions of corresponding players. The final matrices are obtained in the third step by vertical feature scaling. We use a popular feature scaling method shown in Equation (3).

In the second approach (Figure 3), feature weights are found as explained in Methodology Section and multiplied with action counts. It is followed by horizontal normalization and vertical feature scaling similar to the first approach.

Experiments

The data used in this paper are collected from 1751 players playing the Dusk Racer game. After some preprocessing steps, only 259 players data are selected for the experiments. The selected players each exhibit more than 200 events in total. The data are obtained by working with the game company that logged the in-game data into a common structure.

The Results

In our experiments, we aimed to compare event-based, trait-based and trait-based with feature weighting approaches. We apply the clustering algorithm with the same parameters for each approach. The number of clusters (NC) is fixed to 5.

In the first experiment, we use 25 events as the feature set. The results of the event-based approach are presented in Figure 4. Cluster distributions are given as columns. Each color represents a feature, i.e., game event. As it can be seen in the figure, the clusters are not easy to interpret. There are too many features to consider at the same time. Gray, yellow and blue colors represent events "race end", "race start" and "hitting to a car", respectively. They are dominant because their quantities are higher than that of all others. They are not very indicative to discriminate our traits.

In the second experiment, we map events to traits. We assume that an event can be associated with a maximum of 3 traits. Each trait is given scores between $0 - 5$ for each event and vice versa. Feature weighting is not done in this experiment. This is equivalent to using identity matrix as the event-weight matrix (EW). The resulting clusters are presented in Figure 5. Each color represents a trait in this figure. This model of clusters appears to be better for seperation of users . When we examine the clusters, it can be seen that social players are grouped into cluster C4. Cluster C2 consists of more meticulist and strategist players. These two clusters have a similar distribution. Cluster C0 consists of exploiter players, which are not well acquainted with gameplay mechanics. Clusters C1 and C3 are hard to discriminate from each other. They can be said to be balanced.

The third experiment has feature weighting in addition to

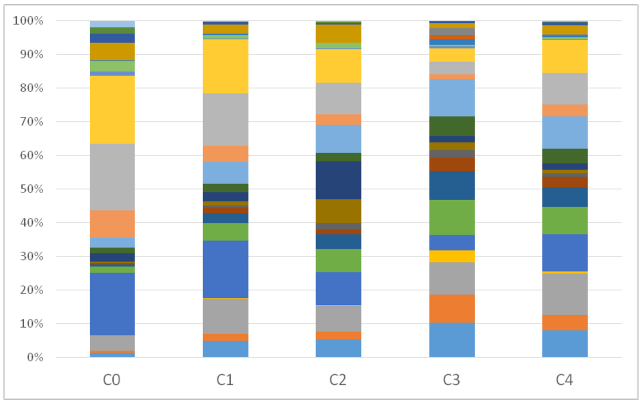

Figure 4: Clusters belonging to different player types as 5 groups by using event-based analysis. Each color represents a game event.

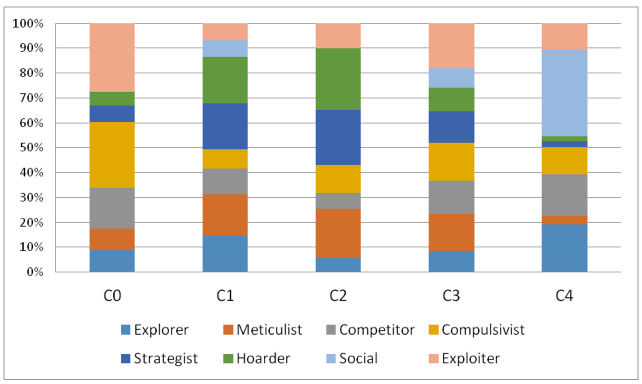

Figure 5: Clusters belonging to different player types as 5 groups by using trait-based analysis. Each color represents a trait.

event-trait mapping. Here event counts are multiplied by the event-weight matrix (EW). The results are given in Figure 6 with colors according to traits. The main difference in the resulting clustering scheme is that there is a cluster (C2) consisting of explorer behaviours. In this experiment, all clusters are easily discriminated. C0 is balanced, C1 is social, C3 is hoarder-strategist and C4 is exploiter-compulsivist.

We observed that HC event dominated all other events in the second experiment, as an undesired situation. However, its effects is weaker in the third experiment. Therefore, the exploiter trait which is connected to action HC appears lesser in the results. That is a positive result of feature weighting.

Subjective comparisons made about the advantages of trait-based analysis with feature weighting to the simple trait-based analysis is also validated using internal cluster criteria, in the following section.

Evaluation

Evaluation of clustering quality is done using three different approaches; internal criteria, external criteria and relative

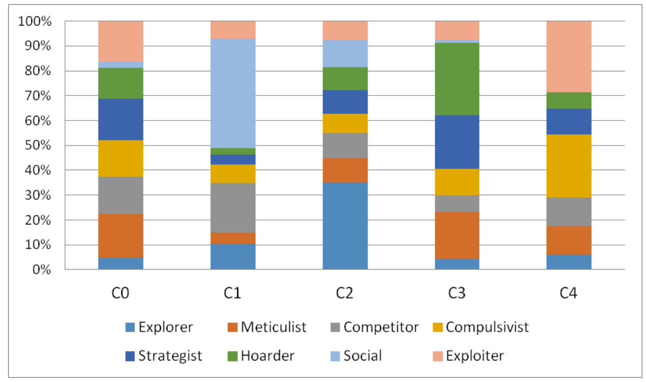

Figure 6: Clusters belonging to different player types as 5 groups by using trait-based analysis with feature weighting. Each color represents a trait.

criteria (Halkidi, Batistakis, and Vazirgiannis 2001). External criteria is used when there is prior knowledge about the real clustering scheme. It compares the resulted clusters with the ground truth ones. Internal criteria of cluster evaluation need only the inherent data of clusters. They can be calculated without any knowledge other than the training data. Relative criteria uses the cluster schemes resulting from different clustering parameters, such as inter-cluster distance (Halkidi, Batistakis, and Vazirgiannis 2001).

Since there is no prior knowledge about the real grouping of players, evaluation of the clustering schemes was an important issue. Labelling of players according to their play style by a domain expert is impractical, if not impossible. Therefore, usage of an external criteria in evaluation is not employed. To evaluate our method, we use *Hubert's Γ Statistic* (Hubert and Arabie 1985) which is a well-known internal criteria of cluster validation. It uses direct distances between instances as an indicator of similarity, as many of other internal criteria. Hubert's Γ Statistic measures the inter-cluster dissimilarity. This value is higher as the clusters are highly separated.

Hubert's Γ Statistic is calculated as shown in Equation (5).

$$\Gamma = \frac{2}{N.(N-1)} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} P(i,j).Q(i,j) \tag{5}$$

Proximity matrix P holds distances of instances, where $P(i,j)$ is the distance between two data points, x_i and x_j. Membership matrix Q consists of $\{0,1\}$ values, where $Q(i,j)$ is 0 if x_i and x_j are the members of the same cluster, 1 otherwise.

We calculate Hubert's Statistic to validate the effect our feature weighting operation. Table 1 contains the Γ values of simple trait-based analysis and trait-based analysis with feature weighting. Euclidian distance is used in the creation of the proximity matrix. Both approaches are tested using Expectation Maximization with the same parameters. We apply feature scaling among trait scores to prevent different scales to cause a bias. Therefore Γ values appear in [0,1]. We repeat

NC = *optimum*		
	trait-based	trait-based with feature weighting
Γ	0.3948	0.5005
NC = 5		
	trait-based	trait-based with feature weighting
Γ	0.4147	0.5896
NC = 10		
	trait-based	trait-based with feature weighting
Γ	0.4325	0.6421

Table 1: Comparison of cluster qualities of simple trait-based analysis and trait-based analysis with feature weighting, using Hubert's Γ Statistic, with different cluster counts (NC). Without weighting the *optimum* cluster number is 4 , while with weighting it is 3.

the experiment with multiple cluster count NC parameters. NC is given three different values 5, 10 and the *optimum* value.

As it can be seen in Table 1, feature weighting increases the cluster quality. This difference is obtained not by changing the parameters of the clustering algorithm, but by preprocessing the input data. Therefore, we can claim that this preprocessing method exploits the player grouping pattern, by increasing the inter-cluster distances.

Discussion

The result of the experiments show that relating events with traits have made the model generate clear cut clusters which increases the usability of the model. Since our approach does not strictly condition the choice of the trait set, it can be modified according to the objective of the game designer and makes it possible to focus on different aspects of player behaviours.

Applying our method on the game data has shown that there are some high-level relations among the game events, which are lost when the model is built solely on event counts. For example, two events may have a similar meaning such as *facebook-share* and *twitter-share*. In our system, these events can simply be mapped to the *social* trait. It can be argued that in an event-based approach, basic events can be grouped under umbrella events to obtain the same effect. However, event relations do not need to be direct and crisp. One event can contribute to the scores of more than one trait. For example, when the player checks the scoreboard screen, this can contribute to both *social* and *competitor* traits.

We developed our method together with the actual designer of the Dusk Racer game. We observed the behaviours of some chosen players from the clusters, and investigated how similar they are and whether the tested models can discriminate them or not. We iteratively improved our method to achieve a better separation of players during the development phase. It revealed the different behaviour groups of players and their distributions, leading to a better understanding of user-game interaction. However, a wider study on game designer satisfaction may be informative as a future work. This can be done as a workshop or through surveys.

Choice of events, traits and their relations with each other is a subjective matter. These can be seen as a language for one to understand the player models. This language can be altered to focus on different aspects of a game. The model of a player is not solely dependent on the personality of the player but also on the choices of the game designer. Moving from this fact, we can think of the models here as not only models of players but player-game interactions.

Evaluation with internal criteria has shown that feature weighting improves clustering quality, which is inter-cluster dissimilarity. This is a good indication about the positive aspect of our method.

It is clear that our method puts some workload on the game designer. Event-trait mapping is done by the game designer. A trait-set can be chosen and modified optionally. But it is still more practical than tailoring an ad-hoc method for a single game. Our method guides the process of evaluating user behaviours. It simplifies the job of the game designer while being able to catch the most important aspects of game-player interaction in regards to predefined game mechanics. It lowers the burden of interpreting a complicated model by decreasing the model complexity.

Conclusion

We presented our player modeling method that uses event-trait mapping relations and feature weighting method which uses the inverse of action frequencies. Our method was shown to be effective in clustering users to traits by Expectation Maximization in a case study.

The event-trait mappings revealed the different behaviour groups of players and their distributions, leading to a better understanding of user-game interaction. However, a wider study on game designer satisfaction may be informative as a future work.

Our method is a good fit with the Dusk Racer game, since it has many countable actions which give insights about the player. Its application on games which have other types of features such as the choice of the character type, the choice of story branches, time/score values worth further investigation.

Indirect effects of our models about user satisfaction and commercial output are not validated in this study. To be able to assess positive business effects, some decisions should be made according to the knowledge acquired using this method. After some decisions are made, A/B tests or user retention measures can be used for conclusions. This remains as a topic for future work.

Acknowledgements

This research is funded by Triodor Software and a grant from the Scientific and Technological Research Council of Turkey (TUBITAK), Grant No. TEYDEB 3140713.

References

Bakkes, S. C.; Spronck, P. H.; and Van Den Herik, H. J. 2009. Opponent modelling for case-based adaptive game ai. *Entertainment Computing* 1(1):27–37.

Bakkes, S.; Tan, C. T.; and Pisan, Y. 2012. Personalised gaming: a motivation and overview of literature. In *Proceedings of the 8th Australasian Conference on Interactive Entertainment: Playing the System*, 4. ACM.

Bartle, R. 1996. Hearts, clubs, diamonds, spades: Players who suit muds. *Journal of MUD research* 1(1):19.

Bateman, C.; Lowenhaupt, R.; and Nacke, L. E. 2011. Player typology in theory and practice. *Proceedings of DiGRA: Think Design Play*.

Bauckhage, C.; Drachen, A.; and Sifa, R. 2015. Clustering game behavior data. *Computational Intelligence and AI in Games, IEEE Transactions on* 7(3):266–278.

Belbin, M. 2004. Belbin team roles. *Book Belbin Team Roles*.

Borbora, Z.; Srivastava, J.; Hsu, K.-W.; and Williams, D. 2011. Churn prediction in mmorpgs using player motivation theories and an ensemble approach. In *Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third Inernational Conference on Social Computing (SocialCom), 2011 IEEE Third International Conference on*, 157–164. IEEE.

Charles, D., and Black, M. 2004. Dynamic player modeling: A framework for player-centered digital games. In *Proc. of the International Conference on Computer Games: Artificial Intelligence, Design and Education*, 29–35.

Drachen, A.; Canossa, A.; and Yannakakis, G. N. 2009. Player modeling using self-organization in tomb raider: Underworld. In *Computational Intelligence and Games, 2009. CIG 2009. IEEE Symposium on*, 1–8. IEEE.

Etheredge, M.; Lopes, R.; and Bidarra, R. 2013. A generic method for classification of player behavior. In *Second Workshop on Artificial Intelligence in the Game Design Process*, 1–7.

Ferro, L. S.; Walz, S. P.; and Greuter, S. 2013. Towards personalised, gamified systems: an investigation into game design, personality and player typologies. In *Proceedings of The 9th Australasian Conference on Interactive Entertainment: Matters of Life and Death*, 7. ACM.

Halkidi, M.; Batistakis, Y.; and Vazirgiannis, M. 2001. On clustering validation techniques. *Journal of intelligent information systems* 17(2):107–145.

Harrison, B., and Roberts, D. L. 2011. Using sequential observations to model and predict player behavior. In *Proceedings of the 6th International Conference on Foundations of Digital Games*, 91–98. ACM.

Hubert, L., and Arabie, P. 1985. Comparing partitions. *Journal of classification* 2(1):193–218.

Kilic, A.; Gunes, M. A.; and Sariel, S. 2016. Player profiling using hidden markov models supported with the sliding window method. In *Proceedings of the 8th International Conference on Agents and Artificial Intelligence*, 362–369.

Missura, O., and Gärtner, T. 2009. Player modeling for intelligent difficulty adjustment. In *Discovery Science*, 197–211. Springer.

Monterrat, B.; Desmarais, M.; Lavoué, E.; and George, S. 2015. A player model for adaptive gamification in learning environments. In *Artificial Intelligence in Education*, 297–306. Springer.

Nacke, L. E.; Bateman, C.; and Mandryk, R. L. 2014. Brainhex: A neurobiological gamer typology survey. *Entertainment computing* 5(1):55–62.

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

Song, Y.-C.; Meng, H.-D.; Ogrady, M.; and Ohare, G. 2007. Applications of attributes weighting in data mining. In *IEEE Proc. of SMC UK &RI 6th Conference on Cybernetic Systems*, 41–45. Citeseer.

Thue, D.; Bulitko, V.; and Spetch, M. 2008. Player modeling for interactive storytelling: a practical approach. *AI Game Programming Wisdom* 4:633–646.

Tychsen, A., and Canossa, A. 2008. Defining personas in games using metrics. In *Proceedings of the 2008 Conference on Future Play: Research, Play, Share*, 73–80. ACM.

Yee, N. 2006. Motivations for play in online games. *CyberPsychology & behavior* 9(6):772–775.

Yu, K.; Xu, X.; Ester, M.; and Kriegel, H.-P. 2003. Feature weighting and instance selection for collaborative filtering: An information-theoretic approach*. *Knowledge and Information Systems* 5(2):201–224.

Zook, A.; Lee-Urban, S.; Drinkwater, M. R.; and Riedl, M. O. 2012. Skill-based mission generation: A data-driven temporal player modeling approach. In *Proceedings of the The Third Workshop on Procedural Content Generation in Games*, PCG'12, 6:1–6:8. New York, NY, USA: ACM.