

Predicting Retention in Sandbox Games with Tensor Factorization-based Representation Learning

Rafet Sifa*, Sridev Srikanth[†], Anders Drachen[‡], Cesar Ojeda* and Christian Bauckhage*[†]

*Fraunhofer IAIS, 53745 St. Augustin, Germany

[†]B-IT, University of Bonn, 53113 Bonn Germany

[‡]Aalborg University and The Pagonis Network, Aalborg, Denmark

Abstract—Major commercial (AAA) games increasingly transit to a semi-persistent or persistent format in order to extend the value of the game to the player, and to add new sources of revenue beyond basic retail sales. Given this shift in the design of AAA titles, game analytics needs to address new types of problems, notably the problem of forecasting future player behavior. This is because player retention is a key factor in driving revenue in semi-persistent titles, for example via downloadable content. This paper introduces a model for predicting retention of players in AAA games and provides a tensor-based spatio-temporal model for analyzing player trajectories in 3D games. We show how knowledge as to trajectories can help with predicting player retention. Furthermore, we describe two new algorithms for three way DEDICOM including a fast gradient method and a semi-nonnegative constrained method. These approaches are validated against a detailed behavioral data set from the AAA open-world game *Just Cause 2*.

I. INTRODUCTION

Aspects of player behavior that are of importance from the point of view game development companies vary. With respect to company sizes, business models, types and formats of games, hardware platforms, and game design, there are different analytical methods. An important aspect is the business model, notably whether Free-to-Play (F2P) or retail-based, which in the former case leads to an interest in predicting the behavior of the players. As of this writing, there is a general lack of predictive models in games being released based on a retail business model. This makes sense given the non-persistent nature of most retail games. However, the situation is changing as the game industry has realized that there are revenue streams available from semi-persistent or persistent games that can be tapped into to increase overall revenue. Therefore, more and more games are released to facilitate longer periods of play, and the lifetime of games is extended.

Changes in how retail based games can be designed to extend the relationship with the player are particularly noticeable among for game productions involving large production and marketing costs, typically referred to as “AAA games”. Here we specifically focus on non-persistent AAA games for which there is now an increasing interest in extending the player-game interaction period by enabling online play via introducing downloadable contents (DLCs) or episodic play, allowing player-generated mods, or similar mechanics. From the point of view of game design, this requires new forms of analytical support including the prediction of player behavior, retention, churn, monetization, or social interactions.

Prediction is however a virtually unexplored topic in AAA games with a few noticeable exceptions [1]. The test case we consider here is the action-adventure game *Just Cause 2* (JC2). JC2 is an Open-World Game (OWG) with large degrees of freedom in player navigation, in world interaction, and narration. While AAA games vary too much in their design to generalize across, it should be noted that spatio-temporal navigation and -tactics are important in many of these games. Spatio-temporal freedom is a characteristic of OWGs but these dimensions are important in any digital game, and have also formed a focus in research in game analytics [2]. Space and time are essentially the dimensions through which user experience occurs, and integrating these in behavioral analysis thus enables the study of gameplay as it is experienced by the player.

A. Contribution

To the best of our knowledge, the work presented here is the first attempt to build a combined retention prediction model for AAA games using an ensemble approach and tensor models for player-wise representation learning and yielding actionable results (up to 81% accuracy) even for the high degrees of player freedom in the OWG format. Secondly, we show that the spatio-temporal dimensions of player behavior can inform the prediction of retention in OWG games, possibly because these are also the dimensions of the player experience and thus a potential proxy of this experience. This result contrasts work in F2P mobile games where spatio-temporal information has not been utilized but retention/churn prediction has been highly successful [1], [3]. Thirdly, we introduce tensor DEDICOM (T-DEDICOM) into the domain of behavioral analytics and present new algorithms for finding its factors efficiently. That is, we develop a tensor-based spatio-temporal model based on the DEDICOM technique that has previously been successfully applied to mine player based spatio-temporal and migration patterns in games [4], [5]. With T-DEDICOM it now becomes possible to account for the movement of many players in one sitting by allowing the spatio-temporal representation to be learned for each player while retaining global information for interpretation.

B. Related work

The idea of studying player behavior to inform game design and development dates back to the earliest digital games but

TABLE I: Dataset description

Feature Type	Descriptor(s)
<i>Telemetry</i>	Number of sessions
	Total playtime
	Total absence time
	Number of days
	Number of actions
	Progress
	Parachutes used
	Vehicles used
	Enemies killed
	Weapons used
	Explosives used
	Number of deaths
	Causes of deaths
	Difficulty level
<i>Meta</i>	Heavy-drop item count
	Blackmarket item count
<i>Temporal</i>	Language
	Platform-played
	Time between daily first and last session
	Inter-day time distribution
<i>Composite</i>	Inter-session time distribution
	Correlation coefficients on time
	Intercept and standard error on time
	Mean and deviation on Time

has significantly advanced with the emergence of MMOGs in the 1990s, due to the need to monitor and respond to a persistent customer base. With the rise of social networking platforms (such as Facebook) and mobile devices and the introduction of the freemium business model, behavioral analytics has now become an integral part of game development [2], [6].

Since an exhaustive coverage of work on game analytics is beyond the scope of this paper, we will focus our discussion on two research directions:

1) Behavioral prediction in games has generally targeted either future player (customer) behavior or sought to inform a variety of situations related to Game AI. Regarding the former the focus has been on persistent games, either F2P mobile/online games [1], [7] or MMOGs [8]. Methods range from pattern recognition and historical analysis over forecasting using simple regression methods to standard machine learning techniques such as decision trees [1], [6], support vector machines [7], and Hidden Markov Models (HMMs) [3]. Hadiji et al. [1] and Runge et al. [3] investigated churn prediction in F2P games using a binary classification approach and benchmarked several methods to predict churn. Sifa et al. [6], Xie et al. [7] and others targeted the problem of predicting purchase structures in games and Thawonmas et al. [8] analyzed player revisitation in a MMOG using login frequencies as the principal proxy. Weber et al. [9] used regression modeling to evaluate retention in *Madden NFL 11*.

2) Spatio-temporal analysis of player behavior has revealed temporal features to play an important role in every predictive game analytics [2]. Operating with both spatial and temporal dimensions, however, is comparatively rare in game analytics but has a strong tradition in Game AI where, for instance, agent behavior models require both dimensions to function. Bauckhage et al. [4] adopted DEDICOM to cluster players of *Quake: Arena* and develop waypoint graphs for behavior-based partitioning. This work in particular forms the basis for the spatio-temporal considerations in the work presented here.

C. Just Cause 2: Gameplay

Just Cause 2 is an open-world (or sandbox) game characterized by a 3D environment covering over 1000 square kilometers of virtual real estate depicting the fictional island nation of Panau and substantial degrees of freedom in terms of spatio-temporal behavior, actions, and narrative progression. The game was published by Square Enix in 2010 and retains an active player base, selling over 6 million units. Similar to other open-world titles, the environment of JC2 is highly varied. Players assume the role of an agent working for an outfit called the Agency. Their goal is to cause disruption leading to the downfall of the resident dictator's reign. This disruption is caused by weakening the power base of the dictator and by taking control of the 9 administrative units of Panau. Players thus have to travel and fight their way across Panau and the game encourages them to traverse the map through missions and exploration, which can be done using a wide variety of vehicles (cars, trucks, boats, planes, helicopters, ...). The game also features a grappling hook which allows players to pull themselves quickly towards stationary or moving objects and can be used in combination with a parachute to rapidly move around.

II. DATA SET AND PRE-PROCESSING

The data set we used in this study has been extracted from Square Enix metrics suite containing records of randomly sampled 5331 players with 7 months coverage and 10,794,666 recorded timestamped actions with their locations. We used a sample of 3572 of the players that have played at least three days to account for the dependencies between the daily behavior. Our objective is a retention prediction case [1] in which we observe the behavioral activities of each player within a 14 days period and determine if they play the game after the following 7 days. In the following we define important behavioral features based on [1], [6], [10] to provide a baseline feature set for retention prediction and explain the main data structure that allows us to model player movements given spatio-temporal actions of players.

A. Extracting Behavioral Telemetry Features

To capture aspects of player behavior, we extract game specific and game independent features that also apply to games of similar genre or setting and therefore group them into four categories: *Telemetry*, *Meta*, *Temporal*, and *Composite*.

In the category called *Telemetry* [6], we have a set of aggregated features quantifying players' basic game play aspects (such as number of sessions or total playtime) and game-play related features (such as progress and number of weapons used), whereas the category *Meta* groups information about the language of the player and the platform on which they play the game. Previous studies [1], [6] showed the importance of temporal features for purchase and churn and we group particular time related features quantifying player's appearance under the *Temporal* category. These include information about the inter-session and -day time distribution as well as the distribution of the daily time interval between the earliest

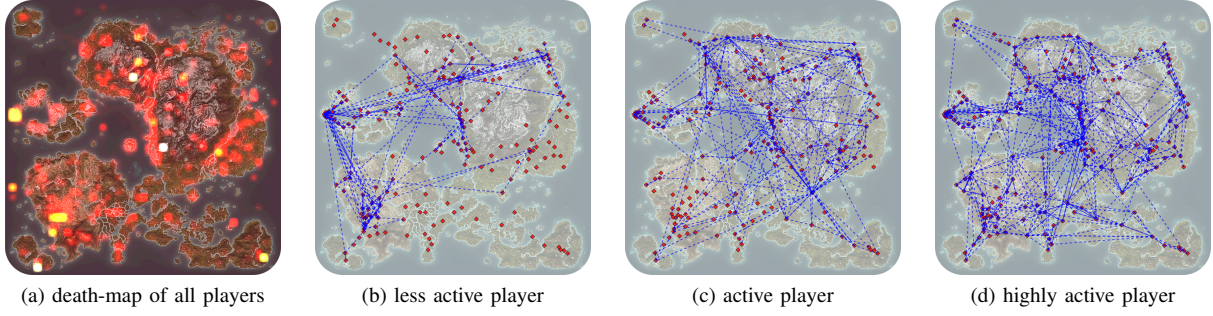


Fig. 1: Spatial activities of players across the islands of Panau. (a) illustrates death-map of all players and (b, c, d) show the movements of three players between the automatically detected waypoints. Best seen in color.

and the latest session. Finally, we consider the daily and session-wise evolution of features in categories *Telemetry* and *Temporal* by defining the category *Composite* that contains features about the mean, deviation, standard error, correlation, correlation coefficient, and the intercept of the values on time. An overview of the extracted behavioral features and their categories are given in Tbl. I.

B. Waypoint Learning with Neural Networks

Player trajectories in JC2 can be described as locally dense but spatially more scattered than the ones we analyzed in [4] due to the increased size and complexity of the game world and game mechanics, see Fig. 1. It is important to note that the use of land-, sea-, and air-vehicles as well as the game specific use of grappling hook as a means of transportation provide nearly unlimited freedom for navigation.

In order to dynamically learn sectors visited by players, we build a *waypoint transition graph* from all observed players trajectories. Waypoint transition graphs provide an informative way to organize spatio-temporal information. They allow for partitioning game maps into coherent parts and can also account for temporal aspects such as transitions between parts [4], [11]. First, we find *prototypical* waypoints capturing the topology of the data points at hand and assign them to be the vertices of a waypoint transition graph. Formally, given a player's position $\mathbf{x}_t \in \mathbb{R}^3$ at time t and player trajectories $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_q\}$ that contain $q \in \mathbb{N}$ observations, we determine $n \in \mathbb{N}$ waypoints $W = \{\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_n\}$ where $\mathbf{w}_i \in \mathbb{R}^3$ and $n \ll q$. Due to the complexity of the world in JC2, we used the Neural Gas Algorithm [11], [12] for learning the map topology. In our experiments, we found the Neural Gas to converge faster and to provide more robust data for predicting player behavior than k -means clustering. Second of all, once topology-capturing waypoints are obtained, a waypoint transition graph $G = (V, E)$ with vertices $V = W$ and edges $E \subseteq V \times V$ is constructed for *each player* by considering movements between waypoints. Namely, for each time-asymmetric pair of subsequent locations $(\mathbf{x}_t, \mathbf{x}_{t+1})$, we find their closest waypoints \mathbf{w}_i and \mathbf{w}_j respectively assign a directed edge between the i -th and j -th node of the graph.

Figures 1b, 1c and 1d illustrate waypoints learned from the movements of all observed players and movements between waypoints for particular players. In the next section we study a tensor factorization model to partition asymmetric similarity matrices while preserving the directional information.

III. DEDICOM MODELS

Decomposition into directional components (DEDICOM) is a family of matrix and tensor decomposition methods originally introduced to analyze asymmetric social ties between people [13]. DEDICOM models deal with asymmetric similarity matrices and decompose them into low rank counterparts involving a loading matrix and a family of affinity matrices [4], [5], [13], [14]. DEDICOM has successfully been used in variety of context including social network analysis [15], natural language processing [16], entity resolution [14], population based player churn analysis [5], and spatio-temporal analysis of player trajectories [4]. Following the analogy introduced in [4], we particularly aim to build a player-based spatio-temporal feature learning framework to compress high level in game interactions to low dimensional representations that also preserves directional information of movements. To do so, we study two- and three-way DEDICOM models and propose an easy-to-implement, novel, and hybrid algorithm that applies projected gradient descent to find DEDICOM factors faster than previous methods. For consistency we start with the two-way DEDICOM model that deals with a single similarity matrix and continue with three-way DEDICOM that allows to decompose multiple similarity matrices.

Formally given a matrix $S \in \mathbb{R}^{n \times n}$ containing asymmetric relations between n entities and $k \in \mathbb{N}$ where $k \ll n$, DEDICOM aims to find the following factorization

$$S \approx \mathbf{A} \mathbf{R} \mathbf{A}^T \quad (1)$$

where $\mathbf{A} \in \mathbb{R}^{n \times k}$ and $\mathbf{R} \in \mathbb{R}^{k \times k}$. The resulting loading matrix \mathbf{A} denotes the hidden structures in S and the affinity matrix \mathbf{R} encodes the asymmetric relations between those structures. Finding a DEDICOM partitioning can be cast as a matrix norm minimization problem with the following loss

function that depends on \mathbf{A} and \mathbf{R}

$$E(\mathbf{A}, \mathbf{R}) = \left\| \mathbf{S} - \mathbf{A} \mathbf{R} \mathbf{A}^T \right\|^2. \quad (2)$$

Typically, alternating least squares algorithms are used that minimize (2) by fixing one of the factor matrices and solving for the other. Various constrained and unconstrained versions have been studied to increase interpretability and speeding up computations [4], [5], [17]. Examples as to these include semi non-negativity constrained DEDICOM [5] which forces \mathbf{R} to be non-negative to interpret the relations as proportions and Decomposition Into Simple Models (DESICOM) [17] which constrains matrix \mathbf{A} to be sparse. For more details about the above models and implementation details, we refer the readers to [4], [5], [15].

The models discussed so far are able handle two dimensional asymmetric similarity data. Given a whole set of asymmetric similarity matrices, we can generalize the above methods to find patterns here as well [13], [14]. Formally, we group a set of $m \in \mathbb{N}$ asymmetric similarity matrices (a.k.a *slices*) $\{\mathbf{S}_1, \mathbf{S}_2, \dots, \mathbf{S}_m\}$ in a three dimensional array (or a *third order tensor*) $\mathcal{S} \in \mathbb{R}^{n \times n \times m}$ where s_{ijr} defines the directional relation between i -th and j -th entity in the r -th slice. Three dimensional generalizations of DEDICOM have been studied from two points of view to reveal the underlying directional patterns. The first in [13], [15] represents the global relationships between all of the slices by encoding them in an asymmetric affinity matrix and each slice's affinity is doubly scaled by a particular diagonal matrix. The second in [14] represents the affinity matrix of each slice individually allowing for a more relaxed representation with fewer factors to optimize over. In the context of individual player trajectory analysis, the later approach has the advantage of learning a player specific representations. Namely given a tensor \mathcal{S} , a relaxed Tensor-DEDICOM partitioning is defined as

$$\mathbf{S}_r \approx \mathbf{A} \mathbf{R}_r \mathbf{A}^T \quad \forall r \in [1, 2, \dots, m] \quad (3)$$

where $\mathbf{A} \in \mathbb{R}^{n \times k}$ is the loading matrix and $\mathbf{R}_r \in \mathbb{R}^{k \times k}$ is an affinity matrix which is the r -th slice of $\mathcal{R} \in \mathbb{R}^{k \times k \times m}$.

Similar to the procedure for its two-way counterpart, finding a three way DEDICOM partitioning can be cast as a norm minimization problem for the loading matrix \mathbf{A} and tensor of affinities \mathcal{R} as

$$E'(\mathbf{A}, \mathbf{R}_1, \mathbf{R}_2, \dots, \mathbf{R}_m) = \sum_{r=1}^m \left\| \mathbf{S}_r - \mathbf{A} \mathbf{R}_r \mathbf{A}^T \right\|^2. \quad (4)$$

Following an alternating least square scheme we can minimize (4) for \mathbf{A} and each slice of \mathcal{R} independently by keeping the rest of the factors fixed. It is important to note that the loss function defined in (4) is convex in any arbitrary slice of \mathcal{R} but not in \mathbf{A} . This leads us to consider approximate solutions for \mathbf{A} which can be approached in numerous ways. For this purpose, we generalize the algorithms derived for two way DEDICOM [4], [5], [15] and propose two novel algorithms to find optimal DEDICOM factors and to generalize the notion of semi non-negativity for interpretability of the factors. Both

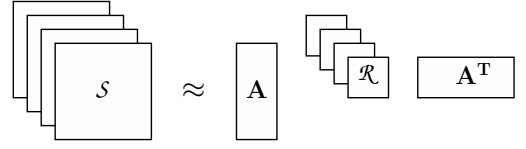


Fig. 2: Illustration of relaxed Three Way DEDICOM. Asymmetrical relationships in $\mathcal{S} \in \mathbb{R}^{n \times n \times m}$ are decomposed into a combination of a latent factor matrix $\mathbf{A} \in \mathbb{R}^{n \times k}$ and a collection of asymmetric mode transition matrices $\mathcal{R} \in \mathbb{R}^{k \times k \times m}$.

algorithms can be easily implemented by extending the python scripts in [4].

A. Approximate Alternating Least Squares Algorithm

Our first algorithm is based on approximating the solution of a matrix equation formed by stacking the data matrices $\{\mathbf{S}_1, \mathbf{S}_2, \dots, \mathbf{S}_m\}$ and their transposes $\{\mathbf{S}_1^T, \mathbf{S}_2^T, \dots, \mathbf{S}_m^T\}$ together as $\mathbf{T} = [\mathbf{S}_1 \mathbf{S}_1^T \dots \mathbf{S}_m \mathbf{S}_m^T]$. Considering the DEDICOM approximation of the each block of \mathbf{T} we have

$$\mathbf{T} = \mathbf{A} [\mathbf{R}_1 \mathbf{R}_1^T \dots \mathbf{R}_m \mathbf{R}_m^T] [\mathbf{I}_{2m} \otimes \mathbf{A}^T], \quad (5)$$

where \mathbf{I}_{2m} is the $2m \times 2m$ identity matrix and \otimes denotes the Kronecker product. The approximation comes into play if we solve for \mathbf{A} by holding \mathbf{A}^T fixed [4], [14], [15]. Namely, if we define the right hand side of \mathbf{A} in (5) as

$$\mathbf{B} = [\mathbf{R}_1 \mathbf{R}_1^T \dots \mathbf{R}_m \mathbf{R}_m^T] [\mathbf{I}_{2m} \otimes \mathbf{A}^T] \quad (6)$$

and assume \mathbf{A}^T being fixed, the problem of finding optimal \mathbf{A} becomes a matrix regression problem with a close form solution given by

$$\mathbf{A} = \mathbf{T} \mathbf{B}^\dagger = \mathbf{T} \mathbf{B}^T (\mathbf{B} \mathbf{B}^T)^{-1}. \quad (7)$$

Therefore, substituting (6) in (7), we obtain the following ALS update for \mathbf{A} :

$$\mathbf{A} \leftarrow \left(\sum_{r=1}^m \mathbf{S}_r \mathbf{A} \mathbf{R}_r^T + \mathbf{S}_r^T \mathbf{A} \mathbf{R}_r \right) \left(\sum_{r=1}^m \mathbf{C}_r + \mathbf{D}_r \right)^{-1}, \quad (8)$$

where $\mathbf{C}_r = \mathbf{R}_r \mathbf{A}^T \mathbf{A} \mathbf{R}_r^T$ and $\mathbf{D}_r = \mathbf{R}_r^T \mathbf{A}^T \mathbf{A} \mathbf{R}_r$. Next, since minimization (4) for a fixed \mathbf{A} is a matrix regression problem [5], [15] the update for each slice \mathbf{R}_r of \mathcal{R} is defined as

$$\mathbf{R}_r \leftarrow \mathbf{A}^\dagger \mathbf{R}_r \mathbf{A}^{T\dagger} \quad \forall r \in [1, 2, \dots, m]. \quad (9)$$

It is important to note that, when working with large matrices, to avoid numerical instabilities during the implementation and work with smaller matrices, optimization for \mathbf{R}_r can be carried out by projecting \mathcal{X} onto a basis of \mathbf{A} using QR decomposition of \mathbf{A} [14], [15].

B. Projected Gradient Descent for Constrained DEDICOM

We next derive an algorithm that is based on the generalization of the projected gradient descent algorithm (called Hybrid Orthogonal [HO] DEDICOM) introduced in [5] to find optimal DEDICOM factors. Unlike the approximate ALS algorithm, by introducing an orthogonality constraint, our method uses

Algorithm 1 Three-way HO-DEDICOM

Randomly initialize \mathbf{A} and \mathcal{R}

while Stopping condition is not satisfied **do**

//Compute the gradient

$$\frac{\partial E'}{\partial \mathbf{A}} \leftarrow -2 \sum_{r=1}^m \left(\mathbf{S}_r^T \mathbf{A} \mathbf{R}_r + \mathbf{S}_r \mathbf{A} \mathbf{R}_r^T \right)$$

//Update \mathbf{A} in the gradient's opposite direction

$$\mathbf{A} \leftarrow \mathbf{A} - \eta_A \frac{\partial E'}{\partial \mathbf{A}}$$

//Project \mathbf{A}_{t+1} by means of QR -Decomposition

$$\mathbf{A}, \mathbf{U} \leftarrow QR(\mathbf{A})$$

//Update slices of \mathcal{R} in ALS manner

for $r \in [1, 2, \dots, m]$ **do**

$\mathbf{R}_r \leftarrow \mathbf{A}^T \mathbf{S}_r \mathbf{A}$

end for

end while

each occurrence of \mathbf{A} when minimizing (4) and this results in a computationally more efficient method that avoids matrix inversions in the updates of \mathbf{A} and \mathbf{R} . Starting with the update for \mathbf{A} , we can write (4) in terms of traces as

$$E'(\mathbf{A}, \mathbf{R}_1, \mathbf{R}_2, \dots, \mathbf{R}_m) = \sum_{r=1}^m \text{tr}[\mathbf{S}_r^T \mathbf{S}_r] - 2 \text{tr}[\mathbf{S}_r^T \mathbf{A} \mathbf{R}_r \mathbf{A}^T] + \text{tr}[\mathbf{A} \mathbf{R}_r^T \mathbf{A}^T \mathbf{A} \mathbf{R}_r \mathbf{A}^T]. \quad (10)$$

Due to the orthogonality constraint imposed on \mathbf{A} , i.e. $\mathbf{A}^T \mathbf{A} = \mathbf{I}_k$, and the invariance under cyclic permutation of the traces, the gradient matrix of E' boils down to

$$\frac{\partial E'}{\partial \mathbf{A}} = -2 \sum_{r=1}^m \frac{\partial}{\partial \mathbf{A}} \text{tr}[\mathbf{S}_r^T \mathbf{A} \mathbf{R}_r \mathbf{A}^T] \quad (11)$$

which results in

$$\frac{\partial E'}{\partial \mathbf{A}} = -2 \sum_{r=1}^m \left(\mathbf{S}_r^T \mathbf{A} \mathbf{R}_r + \mathbf{S}_r \mathbf{A} \mathbf{R}_r^T \right). \quad (12)$$

Referring to (9), owing to the convexity with respect to \mathcal{R} and the orthogonality of \mathbf{A} each slice of \mathcal{R} can be updated globally [5] as

$$\mathbf{R}_r \leftarrow \mathbf{A}^T \mathbf{S}_r \mathbf{A} \quad \forall r \in [1, 2, \dots, m]. \quad (13)$$

Algorithm 1 summarizes the steps for DEDICOM with projected gradient descent. In experiments with the same random initial conditions and stopping criteria, we found that HO-DEDICOM converged faster than the Approximate ALS algorithm (3.26% average speedup) but retains almost the same reconstruction error rate (HO-DEDICOM performs 0.25% better on average in terms of the fitting error). Figure 3 shows runtime and reconstruction error comparisons between HO-DEDICOM and Approximate ALS for asymmetric similarity

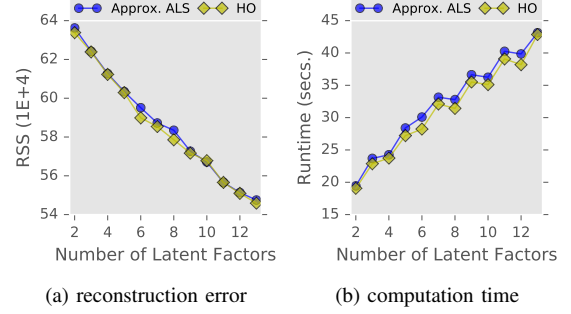


Fig. 3: Evaluation of our proposed algorithm with respect to computation time and reconstruction error from 3126 asymmetric similarity matrices extracted from the waypoint graphs with 200 nodes. While our algorithm yields the same accuracy, it is faster.

matrices extracted from waypoint graphs with 200 nodes covering movement information of 3126 players.

Due to the structure of the DEDICOM approximation, constraining the affinity matrices to be non-negative helps us to interpret both the loadings in \mathbf{A} and \mathcal{R} better [5]. This becomes especially useful when we are dealing with data sets with only non-negative values for which the resulting affinity matrices will be equivalent to their compressed versions. When non-negative affinities are required, we transform the update of slices of \mathcal{R} to a non-negative least square problem as done in [5]. That is, at each ALS step, instead of finding the optimal affinities as done in (13), we find optimal non-negative slices of \mathcal{R} by solving for

$$E''(\mathbf{R}_r) = \left\| \text{vec}(\mathbf{S}_r) - \left(\mathbf{A} \otimes \mathbf{A} \right) \text{vec}(\mathbf{R}_r) \right\|^2 \quad (14)$$

such that $r_{ijr} \geq 0 \quad \forall r \in [1, 2, \dots, m] \wedge i, j \in [1, 2, \dots, k]$. With this latest derivation we conclude the theoretical background about DEDICOM and turn our attention to another tensor factorization model that only deals with symmetric similarity matrices.

IV. INDSCAL

To compare our DEDICOM based feature learning framework and emphasize the importance of finding asymmetric affinity matrices for behavior prediction, we briefly study a tensor based multidimensional scaling model called Individual Differences Scaling (INDSCAL) [18]. The model was proposed in [18] as a special (symmetric) case of Canonical Decomposition in which tensors are decomposed into combinations of rank-1 tensors. Formally, given a tensor \mathcal{L} containing m $n \times n$ similarity matrices in the same domain, INDSCAL partitions each slice as

$$\mathbf{L}_r \approx \mathbf{G} \mathbf{U}_r \mathbf{G}^T \quad \forall r \in [1, 2, \dots, m] \quad (15)$$

where $\mathbf{G} \in \mathbb{R}^{n \times k}$ is the basis matrix representing importance of the ties of each entity and $\mathbf{U}_r \in \mathbb{R}^{k \times k}$ is the diagonal salience matrix, which is the r th slice of $\mathcal{U} \in \mathbb{R}^{k \times k \times m}$,

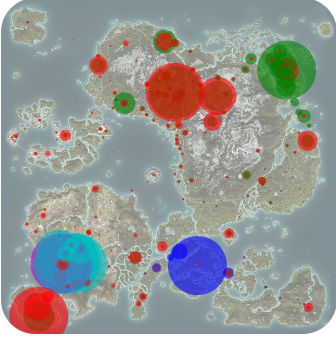


Fig. 4: Color coded bubble diagram of the loadings denoting the sectors that are automatically captured by DEDICOM analysis of the movements of 3126 players. The most important sectors align with the central points in the game play, i.e. the strongholds.

to weight the sum of the rank-1 matrices resulting from the outer product of the columns of the basis matrix. It is important to note that, INDSCAL is a constrained version of the relaxed DEDICOM introduced in (3) to decompose only symmetric similarity matrices. Similar to relaxed DEDICOM, fitting INDSCAL can be cast as a norm minimization problem

$$E'''(\mathbf{G}, \mathbf{U}_1, \mathbf{U}_2, \dots, \mathbf{U}_m) = \sum_{r=1}^m \left\| \mathbf{L}_r - \mathbf{G} \mathbf{U}_r \mathbf{G}^T \right\|^2. \quad (16)$$

A common way to find INDSCAL decompositions is to treat the matrix \mathbf{G} on the left- and right-hand side in (16) as separate matrices (as we considered for approximating DEDICOM's solution above) and to optimize them independently and at the end these two are expected to be equal. Instead of considering an indirect solution, a direct fitting algorithm for INDSCAL is proposed in [19].

Ten Berge et al. [19] find optimal INDSCAL factors by writing (16) in terms of l th column of \mathbf{G} and l th diagonal elements of slices of \mathbf{U} . This can be written as the l th column of \mathbf{G} as

$$E'''(\mathbf{g}_l, u_{1l}, \dots, u_{ml}) = \sum_{r=1}^m \left\| \mathbf{L}_{rl} - \mathbf{g}_l u_{rl} \mathbf{g}_l^T \right\|^2. \quad (17)$$

where $\mathbf{L}_{rl} = \mathbf{L}_r - \sum_{i \neq l}^k \mathbf{g}_i u_{ri} \mathbf{g}_i^T$. Considering an ALS update for unit length constrained \mathbf{g}_l , if we write (17) in terms of traces and separate the terms that only relate to \mathbf{g}_l , minimizing (17) keeping the saliences fixed amounts to maximize

$$E'''(\mathbf{g}_l) = \sum_{r=1}^m \mathbf{g}_l (u_{rl} \mathbf{L}_{rl}) \mathbf{g}_l^T = \mathbf{g}_l \left(\sum_{r=1}^m u_{rl} \mathbf{L}_{rl} \right) \mathbf{g}_l^T. \quad (18)$$

Note that the latest derivation in (18) shows that finding a particular optimal basis vectors for INDSCAL boils down to finding solution for quadratic form which is the eigen-vector corresponding to the largest eigen-value of $\left(\sum_{r=1}^m u_{rl} \mathbf{L}_{rl} \right)$ as

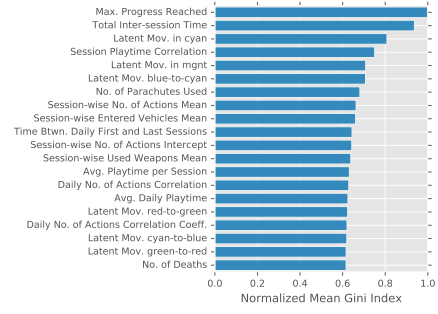


Fig. 5: Normalized gini index values for feature importance extracted from an ensemble of Random Forests that are trained to predict retention. The results indicate that maximum progress reached so far, time dependencies and spati-temporal activities in particular areas of the game map are crucial indicators of future game-play.

the solution. Updating the corresponding saliences of the unit length \mathbf{g}_l is defined as $u_{rl} \leftarrow \mathbf{g}_l \mathbf{L}_{rl} \mathbf{g}_l^T \forall r \in [1, 2, \dots, m]$. Repeating these steps for all of the basis vectors and their corresponding saliences for $l = [1, 2, \dots, k]$ will yield the optimal INDSCAL factors. For more details about INDSCAL and algorithms for finding its appropriate factors we refer the reader to [18], [19].

V. RESULTS

Here we describe the empirical results obtained using the above representation learning framework. First, details of how the framework is used to extract features are presented and then we discuss a case study showing the behavior of DEDICOM for comparative player analysis and feature evaluation. Finally, we report cross validation performance comparisons of the different algorithms.

A. Settings

We consider four cases: a) prediction with purely behavioral features as in [1], [3], [6], b) with purely behavioral features and saliences learned by fitting INDSCAL to the symmetrized similarities between the waypoints and c) + d) constrained and unconstrained affinities obtained by fitting DEDICOM to the asymmetric similarities between the waypoints. As the main aim of the proposed framework is to be used in an agile development cycle, we rely on previous player experiences and build every model by *only* using a training set and evaluate the predictive power of our models using an independent validation set. It is important to note that, when obtaining the saliences \mathbf{U}_{test} and the affinities \mathcal{R}_{test} of the test players we use the basis matrices \mathbf{G}_{train} and \mathbf{A}_{train} that we obtain by respectively fitting INDSCAL and DEDICOM on the similarity matrices of the players in the training set. We measure the performance of the prediction by observing the accuracy of predicting the returners and the non returners (denoted by **Recall** and **Acc** resp.), the **Precision** of predicting the returner and the geometric and the harmonic

TABLE II: Retention prediction results with cross validation

Algo.	Representation	Acc.	Recall	Precision	G-Mean	F-Score
KNN	Pure	0.68	0.61	0.65	0.65	0.63
	Pure & HO	0.72	0.62	0.70	0.67	0.66
	Pure & SNN-HO	0.76	0.67	0.75	0.71	0.72
	Pure & INDSCAL	0.69	0.63	0.67	0.66	0.65
RF	Pure	0.69	0.64	0.72	0.66	0.68
	Pure & HO	0.70	0.67	0.72	0.68	0.70
	Pure & SNN-HO	0.79	0.69	0.75	0.74	0.73
	Pure & INDSCAL	0.75	0.61	0.77	0.67	0.67
GBC	Pure	0.70	0.62	0.68	0.67	0.66
	Pure & HO	0.73	0.68	0.72	0.71	0.71
	Pure & SNN-HO	0.72	0.68	0.72	0.70	0.70
	Pure & INDSCAL	0.72	0.66	0.70	0.69	0.68
ADA	Pure	0.70	0.65	0.72	0.68	0.69
	Pure & HO	0.72	0.67	0.70	0.69	0.70
	Pure & SNN-HO	0.78	0.69	0.75	0.72	0.73
	Pure & INDSCAL	0.70	0.66	0.71	0.68	0.69
Voting	Pure	0.70	0.67	0.70	0.68	0.68
	Pure & HO	0.72	0.69	0.70	0.71	0.71
	Pure & SNN-HO	0.75	0.71	0.72	0.73	0.72
	Pure & INDSCAL	0.70	0.67	0.70	0.69	0.69
Ensemble	Pure	0.69	0.71	0.69	0.70	0.70
	Pure & HO	0.71	0.75	0.70	0.74	0.73
	Pure & SNN-HO	0.75	0.81	0.75	0.78	0.77
	Pure & INDSCAL	0.72	0.69	0.73	0.71	0.71

mean of Recall and Precision (denoted by **G-Mean** and **F-Score** respectively), see [6] for more details about the use of the composite classification measures for classification tasks for games.

B. Dissecting Our Framework

As an example of how our framework learns and represents spatio-temporal information, we set aside one eighth of our JC2 player data for testing and use the remainder for training. We extract waypoints and create a waypoint graph containing information about the asymmetric movements between waypoints which we denote as S_{train} . For ease of interpretation and visualization of the results, we fit SNN DEDICOM on S_{train} with $k = 5$ to obtain the global loading matrix A_{train} and the tensor of player affinities in \mathcal{R}_{train} . The loadings in A_{train} represent how much particular waypoints contribute to latent movements of all of the analyzed players. We present the loadings as a color coded bubble diagram in Fig. 4 where each color represents a particular latent role which at the end forms a sector in the map and the size of each point represents how much it contributes to the sector. Analyzing the loadings on the map we notice that highest loadings correspond to some of the faction strongholds in JC2. Stronghold take-over missions are the endpoints in relation to taking control of a district of Panau, and form some of the most complex and time-consuming missions in the game. It is therefore reasonable to expect JC2 players to spend substantial time around these sites.

To see how the affinities in \mathcal{R}_{train} are structured, we chose the most extreme affinities that are furthest away from each other using k -maxoid clustering [20]. Fig. 6 displays the resulting *extreme* players in terms of their affinities and shows how diverse the movement behaviors can be. The player in Fig. 6a was active in most of the parts of the map whereas the player in Fig. 6b was active mostly in the green sector. Moreover, activities of the player in Fig. 6c are centered around the blue sector whereas the players in Figures 6d, 6e and 6f focused on the lower left part of the map. As a

final step, we trained an ensemble of *only* Random Forests (RF) [21] (a more extensive analysis with more algorithms is provided below) with the pure behavioral data and the spatio-temporal representations learned with SNN-DEDICOM with $k = 5$ where we obtained G-mean and F-score values of 0.77 and 0.76 respectively for the players in the test set. Similar to the analysis in [6], we use the embedded feature ranking capability of Random Forests to investigate how much spatio-temporal features contribute to retention prediction. We present the 20 features with the highest importance values in Fig. 5 and observe that the most important indicator of player retention is the maximum progress reached within the observation time window and, similar to the F2P mobile and social games [1], inter-session time plays an important role for determining if the player will remain in the game or not. Other important indicators of retention are the latent movements in the cyan and magenta sector as well as the ones from blue to cyan (note again the relationship with stronghold missions). It is important to note that the magenta and the cyan sectors form an important milestone in the game as they are in a desert area where the player has to move much and they also contain missions that appear much later than missions in the red and the green sectors. Going down in the list, we see that temporal telemetry features appear to be important as well.

C. Comparative Retention Prediction

Having seen that spatio-temporal features play an important role when it comes to predicting retention, we next analyze what happens if we do not incorporate them into the prediction process and how affinities and the saliences obtained from HO-DEDICOM and INSCAL perform against SNN-HO DEDICOM. We again partition our data into train and test sets to perform cross validation and fit HO- and SNN-DEDICOM on S_{train} and INDSCAL on the corresponding symmetrized version $S'_{train} = (S_{train} + S_{train}^t)/2$ where S_{train}^t denotes the tensor that contains the transpose of the slices of S_{train} . In this case S'_{train} denotes the pairwise similarities of each node visited by the player regardless of its direction. Having fitted the models with latent factors $k \in [2, \dots, 13]$, we compare how well they perform when predicting retention using various classifiers: k -Nearest Neighbors (KNN), Random Forest, Gradient Boosting Classifier (GBC), and Ada Boost (ADA). In addition to using single classification models, we used a community based voting classifier that classifies given players based on the results of the 4 algorithms and (similar to the above analysis) we also used an ensemble of these four classifiers together with Linear and Quadratic Discriminant Analysis in which the clustering of the players done by Neural Gas algorithm. Tbl. II summarizes our retention prediction results categorized by the data representation and classifier used. We observe that adding spatio-temporal features always increases the overall prediction performance with respect to the composite metrics indicating the importance of incorporating such features for activity prediction in games. Additionally, we note that the non-negative features extracted from SNN-DEDICOM yield

	mgnt	blue	cyan	green	red		mgnt	blue	cyan	green	red		mgnt	blue	cyan	green	red
mgnt	86.421	0.003	7.309	0.234	7.274	mgnt	0.001	0.000	0.001	0.373	0.021	mgnt	9.278	0.031	2.715	0.030	1.219
blue	0.003	48.025	0.051	0.015	0.172	blue	0.000	0.000	0.000	0.015	0.001	blue	0.031	5129.513	0.000	0.000	0.000
cyan	7.203	0.051	229.170	0.229	2.257	cyan	0.001	0.000	0.002	0.438	0.014	cyan	2.719	0.000	63.756	0.139	0.266
green	0.234	0.025	0.230	408.387	0.000	green	0.373	0.015	0.438	291.414	5.770	green	0.000	0.000	0.000	49.029	0.000
red	7.281	0.174	2.271	0.000	1372.764	red	0.021	0.001	0.014	5.772	1.393	red	2.699	0.000	0.170	0.000	224.154

(a) A “balanced” player

	mgnt	blue	cyan	green	red		mgnt	blue	cyan	green	red		mgnt	blue	cyan	green	red
mgnt	2447.254	0.000	15.212	0.000	0.000	mgnt	760.357	0.024	87.055	0.000	0.000	mgnt	1201.763	0.000	0.000	0.000	0.000
blue	0.000	0.000	0.705	0.000	0.000	blue	0.008	0.000	0.349	0.000	0.000	blue	0.000	0.000	0.060	0.000	0.000
cyan	0.000	0.706	2184.270	0.000	0.000	cyan	35.252	0.350	1083.662	0.000	0.000	cyan	0.000	0.060	185.911	0.000	0.000
green	0.000	0.000	0.000	0.007	0.021	green	0.000	0.000	0.000	0.003	0.007	green	0.000	0.000	0.000	0.001	0.008
red	0.000	0.000	0.000	0.021	0.126	red	0.000	0.000	0.000	0.008	0.040	red	0.000	0.000	0.000	0.008	0.042

(d) Active in lower part of the map

	mgnt	blue	cyan	green	red		mgnt	blue	cyan	green	red		mgnt	blue	cyan	green	red
mgnt	0.001	0.000	0.001	0.373	0.021	mgnt	0.001	0.000	0.001	0.373	0.021	mgnt	9.278	0.031	2.715	0.030	1.219
blue	0.000	0.000	0.000	0.015	0.001	blue	0.000	0.000	0.000	0.015	0.001	blue	0.031	5129.513	0.000	0.000	0.000
cyan	0.001	0.000	0.002	0.438	0.014	cyan	0.001	0.000	0.002	0.438	0.014	cyan	2.719	0.000	63.756	0.139	0.266
green	0.373	0.015	0.438	291.414	5.770	green	0.373	0.015	0.438	291.414	5.770	green	0.000	0.000	0.000	49.029	0.000
red	0.021	0.001	0.014	5.772	1.393	red	0.021	0.001	0.014	5.772	1.393	red	2.699	0.000	0.170	0.000	224.154

(b) Active in green sector

	mgnt	blue	cyan	green	red		mgnt	blue	cyan	green	red		mgnt	blue	cyan	green	red
mgnt	0.001	0.000	0.001	0.373	0.021	mgnt	0.001	0.000	0.001	0.373	0.021	mgnt	9.278	0.031	2.715	0.030	1.219
blue	0.000	0.000	0.000	0.015	0.001	blue	0.000	0.000	0.000	0.015	0.001	blue	0.031	5129.513	0.000	0.000	0.000
cyan	0.001	0.000	0.002	0.438	0.014	cyan	0.001	0.000	0.002	0.438	0.014	cyan	2.719	0.000	63.756	0.139	0.266
green	0.373	0.015	0.438	291.414	5.770	green	0.373	0.015	0.438	291.414	5.770	green	0.000	0.000	0.000	49.029	0.000
red	0.021	0.001	0.014	5.772	1.393	red	0.021	0.001	0.014	5.772	1.393	red	2.699	0.000	0.170	0.000	224.154

(c) Active in the blue sector

Fig. 6: Prototypical affinity matrices as an outcome of k -maxoid analysis of the vectorized slices of the resulting \mathcal{R}_{train} . The results show the diversity in types of movements players performed between the sectors found by DEDICOM. See text for more details. Best seen in color.

the best performance which is associated to regularization (when compared to the version with unconstrained affinities) and the fact that the transition count data we are factorizing is indeed non-negative. We observe feature importance ratios similar to the analysis above. For an ensemble of random forests, the hit ratio distribution for the occurrences of features in the top 25 list is 100%, 83% and 16% respectively for SNN-HO DEDICOM, HO DEDICOM and INDSCAL. In summary, using asymmetric non-negative affinities as spatio-temporal features through DEDICOM partitioning not only provides interpretable results but also increases the performance of retention prediction.

VI. CONCLUSION AND FUTURE WORK

Understanding player interactions and forecasting future behavior are two major tasks for future resource management in AAA game analytics. In this paper, the problem of predicting player retention in these games has been approached through a tensor factorization-based learning framework that integrates spatio-temporal behavior additional to providing descriptive insights. The notion of three way relaxed DEDICOM has been introduced, proposing this as a fast algorithm for retention prediction. Our results indicate that spatio-temporal features are important proxies for user engagement in Open-World Games in a similar capacity to the game-specific and temporal features known from F2P prediction models. Our future work involves using the proposed framework to model player interactions in the context of other AAA and F2P games that allow the player to move freely in the game world.

ACKNOWLEDGMENT

The authors would like to thank Square Enix for access to the telemetry data from Just Cause 2.

REFERENCES

- [1] F. Hadji, R. Sifa, A. Drachen, C. Thureau, K. Kersting, and C. Bauckhage, “Predicting Player Churn in the Wild,” in *Proc. of IEEE CIG*, 2014.
- [2] El-Nasr, M. S. and Drachen, A. and Canossa, A., *Game Analytics: Maximizing the Value of Player Data*. Springer, 2013.
- [3] J. Runge, P. Gao, F. Garcin, and B. Faltings, “Churn Prediction for High-value Players in Casual Social Games,” in *Proc. of IEEE CIG*, 2014.
- [4] C. Bauckhage, R. Sifa, A. Drachen, C. Thureau, and F. Hadji, “Beyond Heatmaps: Spatio-Temporal Clustering using Behavior-Based Partitioning of Game Levels,” in *Proc. of IEEE CIG*, 2014.
- [5] R. Sifa, C. Ojeda, and C. Bauckhage, “User Churn Migration Analysis with DEDICOM,” in *Proc. of ACM RecSys*, 2015.
- [6] R. Sifa, F. Hadji, J. Runge, A. Drachen, K. Kersting, and C. Bauckhage, “Predicting Purchase Decisions in Mobile Free-to-Play Games,” in *Proc. of AAAI AIIDE*, 2015.
- [7] H. Xie, S. Devlin, D. Kudenko, and P. Cowling, “Predicting Player Disengagement and First Purchase with Event-frequency Based Data Representation,” in *Proc. of CIG*, 2015.
- [8] R. Thawonmas, K. Yoshida, J.-K. Lou, and K.-T. Chen, “Analysis of revisitations in online games,” *Entertainment Computing*, vol. 2, no. 4, pp. 215–221, 2011.
- [9] B. G. Weber, M. John, M. Mateas, and A. Jhala, “Modeling Player Retention in Madden NFL 11,” in *Proc. of IAAI*, 2011.
- [10] A. Drachen, R. Sifa, C. Bauckhage, and C. Thureau, “Guns, Swords and Data: Clustering of Player Behavior in Computer Games in the Wild,” in *Proc. of IEEE CIG*, 2012.
- [11] R. Sifa and C. Bauckhage, “Archetypal Motion: Supervised Game Behavior Learning with Archetypal Analysis,” in *Proc. of CIG*, 2013.
- [12] T. Martinetz and K. Schulten, “A “neural-gas” network learns topologies,” *Artificial Neural Networks*, pp. 397–402, 1991.
- [13] R. A. Harshman, “Models for Analysis of Asymmetrical Relationships among N Objects or Stimuli,” in *Proc. Joint Meeting of the Psychometric Society and the Society for Mathematical Psychology*, 1978.
- [14] M. Nickel, V. Tresp, and H. Krieger, “A Three-way Model for Collective Learning on Multi-relational Data,” in *Proc. of ICML*, 2011.
- [15] B. Bader, R. Harshman, and T. Kolda, “Temporal Analysis of Semantic Graphs using ASALSAN,” in *Proc. of IEEE ICDM*, 2007.
- [16] P. A. Chew, B. W. Bader, and A. Rozovskaya, “Using DEDICOM for Completely Unsupervised Part-Of-Speech Tagging,” in *Proc. Workshop on Unsupervised and Minimally Supervised Learning of Lexical Semantics*, 2009.
- [17] H. A. Kiers, “DESIOM: Decomposition of Asymmetric Relationships Data into Simple Components,” *Behaviormetrika*, vol. 24, no. 2, pp. 203–217, 1997.
- [18] J. D. Carroll and J.-J. Chang, “Analysis of Individual Differences in Multidimensional Scaling via An N-way Generalization of Eckart-Young Decomposition,” *Psychometrika*, vol. 35, no. 3, pp. 283–319, 1970.
- [19] J. M. ten Berge, H. A. Kiers, and W. P. Krijnen, “Computational Solutions for the Problem of Negative Saliences and Nonsymmetry in INDSCAL,” *Journal of classification*, vol. 10, no. 1, pp. 115–124, 1993.
- [20] C. Bauckhage and R. Sifa, “k-Maxoids Clustering,” in *Proc. of KDML-LWA*, 2015.
- [21] L. Breiman, “Random forests,” *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.