Guns, Swords and Data: Clustering of Player Behavior in Computer Games in the Wild

Anders Drachen, Rafet Sifa, Christian Bauckhage and Christian Thurau

***Abstract*—Behavioral data from computer games can be exceptionally high-dimensional, of massive scale and cover a temporal segment reaching years of real-time and a varying population of users. Clustering of user behavior provides a way to discover behavioral patterns that are actionable for game developers. Interpretability and reliability of clustering results is vital, as decisions based on them affect game design and thus ultimately revenue. Here case studies are presented focusing on clustering analysis applied to high-dimensionality player behavior telemetry, covering a combined total of 260,000 characters from two major commercial game titles: the Massively Multiplayer Online Role-Playing Game *Tera* and the multi-player strategy war game *Battlefield 2: Bad Company 2*. K-means and Simplex Volume Maximization clustering were applied to the two datasets, combined with considerations of the design of the games, resulting in actionable behavioral profiles. Depending on the algorithm different insights into the underlying behavior of the population of the two games are provided.**

I. Introduction

Since the first computer game, the behavior of players has been registered, and responses to these behaviors calculated in real-time by the game software. It was not long into the life of computer (or video) games that measures of player behavior began appearing in a form visual to the player, for example high score lists, which stem back to the earliest arcade games. With the advent of the massively multi-player online game (MMOG), e.g. *Meridian 59* and *EverQuest*, player behavior analysis became important to monitor the population of persistent virtual worlds, e.g. ensuring stable economies and detecting fraudulent behavior [22]. Contemporaneously, user-oriented testing and research methods have been widely adopted by game development [9, 12,36]. Initially, laboratory-based methods have been utilized to analyze the behavior of users of computer games and the resulting experience. Over the past decade, principles and practices from game user research and telemetry analysis have begun to merge, providing hitherto unprecedented analytical power to user research, e.g. via permitting the collection of behavioral data in "the wild", from user-game interaction, purchasing behavior, social behavior, etc., giving rise to a series of different forms of analytics enabling high-resolution and large-scale behavioral analysis [25,33,38].

The growing interest in user-oriented behavior analysis in computer games is in part driven by the emergence of the multi-player and MMOG and Free-to-Play (F2P) game forms, which can support populations in the millions, as well as millions of objects and AI/script-driven entities. These games form a rules-governed complex of potential or realized interactions, and can be highly complex in terms of the user interactions they provide [2,5,12]. Given such complexity, obtaining insights that are meaningful and actionable for game developers can be challenging, and means that behavior analysis is difficult to perform without dimensionality reduction, e.g. clustering.

Behavioral analysis based on player-centric telemetry data can be carried out in numerous ways, and there is a vast body of research and practice to draw upon, e.g. from web analytics [10]. A central challenge is dimensionality: Complexity of games varies from the simple mechanics of *Tic Tac Toe* to the hundreds of potential player actions in major commercial titles like *World of Warcraft*. In the case of games with complex game mechanics, clustering (and classification) analysis provides a means for discovering any underlying patterns in the behavior of players which can be of immense value to game development [1,33,37].

II. Contribution and Main Result

In this paper, the challenge of obtaining actionable insights from behavior clustering in computer games is investigated. Two case studies are presented, both focused on large-scale, high-dimensionality player behavior telemetry analysis, across two major commercial game titles: the Massively Multiplayer Online Role-Playing Game *Tera* and the multi-player strategy war game *Battlefield 2: Bad Company 2* (Figure 1). A combined total of 260,000 characters with associated behavioral features are included in the analysis.

Simplex Volume Maximization (SIVM), an adaptation of Archetype Analysis (AA) that is applicable to large-scale datasets [22,26] and k-means clustering were applied to the datasets, resulting in behavioral classes which are described in terms of design language [1,33,34] and the specific advantages of each algorithm in terms of evaluating player behaviors are discussed . The challenges in performing cluster analysis across high-dimensional game telemetry datasets, containing different data types, are described, and solution strategies are outlined.

III. Player Clustering in the Wild

In the context of customer behavior analysis in computer

A. Drachen is with Aalborg University, Ballerup, Denmark (email: drachen@hum.aau.dk) and Game Analytics, Copenhagen, Denmark (email: anders@gameanalytics.com); R. Sifa is with the University of Bonn, Bonn, Germany (email: thrafet@gmail.com) ; C. Bauckhage is with Fraunhofer IAIS and the University of Bonn, Bonn, Germany (email: christian.bauckhage@iais.fraunhofer.de); C. Thurau is with Game Analytics, Copenhagen, Denmark (email: christian@gameanalytics.com).

978-1-4673-1194-6/12/$31.00 ©2012 IEEE

game development, clustering and classification analysis provides a means for reducing the dimensionality of a dataset in order to find the most important features, and locate patterns which are expressed in terms of user behavior as a function of these features, which can be acted upon to test and refine a game design (or specific parameters of a design) [1,15]. Clustering and classification are also of interest in game research, e.g. areas focusing on player experience modeling, game AI and the development of adaptive games [25,38].

Current practices in the game industry are incredibly difficult to evaluate because telemetry data and analysis methods are considered proprietary. It is only through game industry events such as the Game Developers Conference [e.g. 12,34], industry magazines, and various blogs/news-sites, that publicly available information about how the industry is currently performing telemetry can be found. However, information from these sources is generally not detailed and provides few insights into the specific algorithms being used.

A relatively new source of information on industry standards is the about two dozen analytics companies that have emerged in the past few years to supply middleware to the game industry. Some of these offer segmentation or clustering/classification tools; however, players are typically segmented into pre-defined classes, e.g. "whales" who are characterized by spending a lot of money on virtual items or upgrades in-game [37]. This approach can be useful but has the inherent problem of fitting data to classes that may not exist in the dataset. This is notably dangerouos for persistent games where the population of players can change over time. Using pre-defined features also prevents exploratory behavior analysis and runs the risk of missing important patterns in the data being worked with. Furthermore, people inexperienced with behavioral analysis can be hard pressed to design classes that are actionable. In summary, the use of pre-defined classes for segmentation should at the least be checked against unsupervised cluster analysis.

There exist numerous methods for unsupervised and supervised learning which can be applied to user behavior in games. Not only are there a substantial number of algorithms to choose from, but the application of these to game telemetry analysis is not straight forward. There are several major challenges, notably: 1) The potential extreme high-dimensionality of the behavioral data [1,33,34,38]; 2) It can be necessary to mix datatypes, e.g. binominal, categorical and numerical values, in the same analysis, in order to include all the necessary behavioral features; 3) Telemetry datasets are often noisy [e.g. 5,9]; 4) The techniques generally require informed decisions as to the number of clusters to extract [28]; 5) The results of player behavior clustering have to be actionable. This means being able to relate results to the design of the game in question, which entails converting results to a language understandable by developers who may not be expert analysts.

None of these challenges are unique to game telemetry, and solutions developed for handling large-scale data exist. However, there is as yet no firm body of knowledge guiding the application of clustering to behavioral data from games. This includes interpretability which is important in a practical development context, where the results of a clustering analysis should be as easy as possible to interpret. Given the potential massive size and high dimensionality this is an important knowledge gap [23]. In an ideal situation, being able to assign a simple expressive label to each basis vector or cluster centroid describing the component behaviors is optimal; however, there is no objective criterion available for defining what a descriptive representation is, although it is generally assumed that results are interpretable when they embed data whose basis vectors correspond to data points [1,22,26].

IV. Related work

Within game-oriented AI research, agent modeling, adaptive game research, player experience modeling, adaptive game research and not the least game-user research, the use of telemetry data extracted from gameplay behavior has been in use for about a decade [12,25,38]; however it is only in recent years that the game industry (with a few exceptions) outside of the massively multi-player online (MMOG) segment has begun adopting player-derived telemetry data to evaluate player behavior in games, e.g. [12,22,31,34].

Early work in the area was championed by what is now known as *Microsoft Studios Research*, who gained international recognition for their user research in the *Halo*-series of games [12,36]. In the past five years, other major game development publishers such as *Bioware, Blizzard, Bioware, Square Enix* and *EA Games* have been collecting and analyzing massive-scale behavioral telemetry from their games [38], although the details of the methods used are kept confidential [16,34] outside the rare academic-industry partnerships [e.g. 1,33].

In recent years, the rise of the free-to-play (F2P) genre, e.g. on platforms such as *Facebook* and *Google Play,* has added to the industry´s focus on behavior analysis. In F2P games, which can be of a persistent nature similar to MMOGs, playing the game itself is free, and revenue dependent on the ability of the developer to convince a portion of the customer base to purchase virtual items for real money via micro-transactions [16,37]. In order to be successful as a business model, these games require continued analysis of player behavior in order to be financially profitable [37].

Academic and industrial research has built considerable knowledge, but there is minimal knowledge exchange between the two. It is for example only recently that academic experts have gained access to commercial game datasets [38]. The recency of game telemetry as a research topic also means that most available work is case-based, e.g. application of a specific algorithm to behavioral data from a specific game.

Focusing on academic research, there are an increasing number of publications data have been obtained from **online services** [e.g. 22] or **obtained datasets from developers** [e.g. 1,5,33]. These studies highlight the need for robust algorithms to classify data which can scale to large datasets.

Clustering and classification of player behavior has been explored by e.g. [17], who used k-means and support vector machines to predict dynamic difficulty adjustments for a shooter-type game. [21] used frequency analysis to find behavior patterns in the MMOG Cabal Online, focusing specifically on bot-detection via identifying aberrant behavior. Bot-detection is a key topic in online game development, where in-game resources represent considerable value in terms of gameplay and sometimes also in terms of real-world value (e.g. *World of Warcraft* gold, Linden dollars in *Second Life*). Weber & Mateas [24] employed a series of classification algorithms for recognizing player strategy in *StarCraft*, employing regression in order to predict when specific unit or building types would be produced. Thurau et al. [22] applied nonnegative-matrix factorization to classify player guilds in *World of Warcraft*. Also in the MMOG domain, Ducheneaut & Moore [6] examined group player behaviors in the MMOG *Star Wars Galaxies* via action frequency analysis. Focused on behavior prediction, [20] studied revisitations in MMOGs (to the game or a specific area in the game world). Mahlman et al. [15] used classification on a sample of 10,000 players from *Tomb Raider: Underworld*, demonstrating how behavior can be predicted based on analysis of early play. In related work, Southey et al. [31] developed a general purpose analysis tool (SAGA-ML) to analyze metrics data from the game FIFA´99, a soccer game. The aim was to identify "sweet spots", i.e. faults in the game design that permits maneuvers that can be sued to repeatedly score goals. Similarly focused on gameplay monitoring, Marsh et al. [32] focused on telemetry as a means for capturing user-player behavior to inform design. Drachen et al. [1], which based on a selection of behavioral variables from key game mechanics, classified the behavior of 1365 players of *Tomb Raider: Underworld*. The authors used Emergent Self-Organizing Networks, locating four clusters of user behavior that encompassed over 90 percent of the players. These were converted into behavioral profiles for the developers of the game.

Spatiotemporal behavior clustering and classification integrates the spatial component of games, which assists with evaluating play experience [5]. E.g. a number of approaches for trajectory analysis and –classification have been adapted to spatial game analytics [30], used e.g. to detect illegal bot programs, study player tactics or to train bots [20,21,30].

V. Dimensionality reduction by clustering

Unsupervised clustering techniques vary, from clustering algorithms such as k-means and c-means, to low rank methods such as Principal Components Analysis [8,11] and Non-negative Matrix Factorization [14,18], and Archetype Analysis (AA) [4,22,26]. Methods for unsupervised learning include Self-Organizing Maps (SOM). Artificial Neural Networks (ANNs) on the other hand are used for classification and prediction [1,7]. Using approaches such as v-fold cross validation that adds a quasi-supervised component to cluster analysis. Additionally, interpretation of clusters from the perspective of actual game design can be difficult.

A key assumption in clustering [35] is that the behavior telemetry data can be stored in a $d \, x \, n$ matrix $\boldsymbol{V} = [v_{1,\ldots,}v_{\mathrm{n}}] \in \mathrm{R}^{d \times n}$, s.t. each column corresponds to a particular player (or agent). Essentially, when dealing with a situation where n samples of d-dimensional vectorial data are gathered in a data matrix $\boldsymbol{V}^{d \times n}$, the problem of determining useful clusters corresponds to finding a set of $k \ll n$ centroid vectors $\boldsymbol{W}^{d \times k}$. If membership of the data points $\boldsymbol{V}$ to the centroids in $\boldsymbol{W}$ is expressed via a coeffient matrix $\boldsymbol{H}^{k \times n}$, clustering can be cast as a matrix factorization problem; where the aim is to minimize the expected Frobenius norm $||\boldsymbol{V} - \boldsymbol{WH}||$. While methods such as PCA, NMF and k-means all try to minimize the same criterion, they impose different constraints and thus yield different matrix factors [7,11]. For example, NMF assumes $\boldsymbol{V}$*,* $\boldsymbol{W}$, and $\boldsymbol{H}$ to be non-negative matrices and often leads to sparse representations of the data. PCA constrains $\boldsymbol{W}$ to be composed of orthonormal vectors and produces a dense $\boldsymbol{H}$*,* where k-means clustering constrains $\boldsymbol{H}$ to unary vectors. K-means is perhaps the most widely adopted unsupervised clustering algorithm, and is theoretically suited for game telemetry features, however, it is focused on retrieving compact cluster regions, and can therefore in practice be hard to interpret [35].

Archetype Analysis as introduced in [4] and recently extended to large-scale datasets by [22,26] via Simplex Volume Maximization (SIVM), applies an alternating least squares procedure where each iteration requires the solution of several constrained quadratic optimization problems. It solves the case where **G** is restricted to convexity instead of to unarity. SIVM appears to be attractive to game telemetry analysis because it allows for the detection of "special" player behaviors, as it is focused on finding extremes in the dataset. In essence, what SIVM does is automatic detection of a combination of features that leads, when being locked in pairs, to a similar but more complex segmentation as k-means, without any user intervention (e.g. in determining the value of *k*). Where k-means produce cluster centroids, SIVM is different in that it does not look for commonalities between players, but rather archetypical (extreme) profiles that do not reside in dense cluster regions, but at the edges of the multi-dimensional space. This attractive feature of SIVM is however also its central weakness in the current situation, as it is highly sensitive to outliers. If the goal of analysis is to detect outliers, e.g. for detection of bots, cheating or otherwise subversive player behavior, or conversely players that are valuable to keep in the game [21], this is desirable. This problem does not occur for pure Archetype Analysis, but SIVM is an approximation of AA for large-scale data, which unlike AA fundamentally ignores the distribution of the data, thus adding a weakness to outliers. For a better coverage of distorted distributions, Kersting et al. [26] suggested a hierarchical extension to SIVM that automatically selected regions of the target space to explore and thereby reduces outlier influence. As an alternative, it was here decided to exclude outliers by peeling the convex hull of the data [27]. By definition, outliers have to reside on the convex hull of the data. To reliably detect and exclude them, we apply the *Fastmap* algorithm which iteratively yields pairs of convex hull data points [27]. As *Fastmap* scales linearly with the number of data points, and as we are only interested in the first two samples, selection of outliers can be done efficiently for vast numbers of data. Essentially, data points are iteratively excluded which exhibit the largest pairwise distance among all data points as these have to reside on the convex hull and therefore are possible outliers.

VI. DATASETS

For the study presented here, two datasets containing player behavior telemetry were analyzed. These include one massively-multiplayer online role-playing game (MMORPG), and one multi-player online war strategy/action game.

A. Tera

Tera (abbreviation for *The Exiled Realm of Arborea)* is a MMORPG set in a fantasy-themed virtual world. The game was released on South Korea in January 2011 and in North America/Europe on May 3, 2012. At the time of writing this paper, the game is in closed beta-testing for these releases, and the data used for this paper originate from this test phase. The game has typical MMORPG features such as a questing system, crafting and player vs. player action, as well as an integrated economy. Players generate one or more characters, which fall within one of seven races (e.g. Aman, Baraka, Castanic). In addition, players choose a class (e.g. Warrior, Lancer, Beserker), each tuned to specific roles in the game (e.g. having a high damage output or being able to absorb high amounts of damage).

The dataset from *Tera,* includes 250,000 player characters, containing two groups of features: 1) **Character ability features:** are related to the abilities ("stats") of the characters, e.g. class, race, strength, attack, defense etc. (8 features in total). These are not analyzed here due to space restrictions, 2) **Gameplay features:** are related to how the characters are played e.g. monsters killed, in-game friends, quests completed). The high standard deviations in these features are largely the result of the level-based design of *Tera*, i.e. the presence of characters of 32 different levels in the dataset. A total of 18 features were included in the analysis presented here, selected from a range of 33 pre-selected features in the dataset provided by Bluehole Studio, the developer of the game. The features provided are all aggregate, which defines the level of detail in the analyses that can be performed, e.g. we do not know which quests the player completed, just the total number of quests. This limits the detail of the results that can be derived from the analysis of the dataset, which is taken into account in the below.

Tera, gameplay features:

- **Quests completed:** Number of quests completed (μ = 40.9, σ = 59.3**).**
- **Friends:** Number of friends in the game (μ=0.44, σ=1.4)
- **Achievements**: The number of achievements earned (μ =10.7, σ = 13.4).
- **Mining** and **Plants**: Level in the Mining and Plants skill respectively (μ = 5.3 and 2.6 respectively, σ = 16 and 8.9 respectively).
- **Kills_monsters**: The number of AI-controlled enemies killed by the character (combining small, medium and large monsters in one feature) (μ = 625.2, σ = 1242.9).
- **Loot_total_items**: The total number of items the character has picked up during the game (μ=100.9, σ =200.5).
- **Deaths_monsters**: The number of times the character has been killed by AI-controlled enemies (μ = 2.7, σ = 12.1).
- **Auction**: The combined number of times the character has either created an auction or purchased something from an auction (μ = 818.3, σ = 1667.9).
- **Character level:** Ranges from level 1 to 32 (μ = 7.96, σ = 7.6). Note that a player can have multiple characters in *Tera*, and the dataset therefore probably represent a number of players lower than the actual number of characters. From the perspective of behavior clustering, the discrepancy between number of players and characters is not important as it is the in-game behavior of each character that is of interest.

B. Battlefield 2: Bad Company 2

Battlefield 2: Bad Company 2 (*BF2BC2*) is a first-person shooter (FPS) game with strategic and tactical elements reflecting small-scale warfare, developed by EA DICE. It was released on March 2nd, 2010 for PC, Xbox360 and PlayStation 3. The game puts the player in a fictional war scenario between the United States of America and the Russian Federation. The game has a single-player, campaign mode, and a multi-player mode supporting up to 24 concurrent players (32 on PC for conquest and rush mode), the latter being by far the most popular version of the game.

In the multi-player version of the game, each player takes control of one character (a soldier), playing as part of a team against another team, in various types of scenarios (e.g. conquest mode). Unlike in *Tera*, the player can select between one of four different classes (or "kits") every time a new multi-player game is started: Assault, Engineer, Medic and Recon. The classes are descriptive of the kind of role that the player will have on the battlefield, and determines the starting equipment packs. E.g. Engineer classes are equipped with anti-vehicle weapons (RPGs) and anti-tank mines. Players can earn ranks, awards and special equipment over the course of their multi-player career.

In *BF2BC2*, each player has one account, with a dedicated name. This name is used for all instances of play and permits tracking of telemetry across different classes. The dataset from *BF2BC2* includes 10,000, randomly sampled from a larger dataset of 69,313 players, all PC players, which were again selected randomly from the p-stats network (http://bfbcs.com/), a service which collects telemetry data from individual game clients, aggregates the data and makes them accessible to the players. P-stats provide an API for fetching *BF2BC2* stats from the player, which can be used by software engineers to integrate telemetry-based statistics (*BF2BC2* runs via a server which hosts the game, to which clients connect). A total of 11 features were extracted from the dataset, with some of these being compound features. All features are gameplay features, and notably include playtime information. Given that more than a hundred features are available from the p-stats network, selecting the 11 features required consideration. In this, we followed the method suggested by Drachen et al. [1], i.e. selecting features that allow for evaluating of the most important gameplay mechanics in the game under evaluation. In the case of *BF2BC2*, this means features relating to **character performance** (score, skill level, accuracy etc.) and **game feature** use (kit stats, vehicle use), and **playtime**:

- **Score**: Total number of points scored (μ = 2,283,057, σ = 3,092,352).
- **Skill level:** An aggregate measure of player skill (μ = 378.78, σ = 209.74).
- **Total playtime**: The sum total of time the player´s account has been active (μ = 214.60 hours, σ = 7.6 hours)
- **Kill/Death ratio:** K/D ratio, the number of kills the player has scores divided with the number of deaths suffered (μ = 0.96, σ = 0.52).
- **Accuracy:** The percentage of hits scores with weapons (μ = 76.75, σ = 88.73).
- **Score per minute**: The average number of points scored per minute of play while on active combat missions (μ = 160.39, σ = 68.15).
- **Deaths per minute/Kills per minute**: Average deaths (μ = 0.78, σ = 0.21) or kills per minute (μ = 0.72, σ = 0.33).
- **Rounds played:** The number of game rounds the player has played (μ = 916.5, σ = 1,126.12).
- **Kit stats**: The number of points scored with each kit (class) and the number of kills and deaths for each class [Tbl. 1].
- **Vehicle use**: Total time spent in air, water, land-based or stationary vehicles (μ = 32.1 hours, σ = 51.69 hours).

TABLE I
DESCRIPTIVE STATISTICS FOR THE FOUR KITS (CLASSES) IN *BF2BC2*.

Kit (feature)	μ	σ
Assault (Kills)	3.380,45	5.756,96
Assault (Deaths)	3.067,45	4.336,74
Assault (Score)	312.599,76	530.139,59
Engineer (Kills)	2.737,85	5.071,96
Engineer (Deaths)	2.436,23	3.646,37
Engineer (Score)	251.834,96	449.919,46
Medic (Kills)	1.861,15	3.461,02
Medic (Deaths)	1.830,00	2.779,41
Medic (Score)	227.608,75	406.931,31
Recon (Kills)	2.336,99	4.229,14
Recon (Deaths)	1.858,77	2.697,33
Recon (Score)	218.546,18	399.957,06

VII. DATA PREPARATION AND ANALYSIS

Threshold definition: Unsupervised learning contain the inherent challenge of lacking an objective way to define threshold values. E.g. in case of k-means, the number of clusters, or the number of neurons in a SOM [1]. This makes defining the number of clusters to use a subjective decision, adding to the difficulty in adopting such methods by non-experts in development contexts. Different approaches for alleviating this problem exist, notably mean squared error estimates, cross validation and the popular Scree plots [35].

Data type mixing and normalization: A typical problem of behavioral analysis in games is the mixing of data types. This requires the adoption of normalization strategies such as min-max and variance (or zero mean normalization, ZMN) [35].

ZMN normalizes the field values according to the mean and the σ (standard deviation) values. ZMN substracts the values from the mean and divides the result by σ. If $\boldsymbol{F}$ is the values of the fields to be normalized, the normalized values $\boldsymbol{F}'$ are: $\boldsymbol{F}' = \frac{(\boldsymbol{F}-\boldsymbol{\mu_F}\,)}{\sigma_F}$. Min-max normalization transforms the data into a defined range normalized min value $(\alpha)$and normalized max value (β). If $\boldsymbol{F}$ are the values of the fields to be normalized, the normalized values $\boldsymbol{F}'$ are defined by: $\boldsymbol{F}' = \left[\frac{\boldsymbol{F}-\min(\boldsymbol{F})}{\max(\boldsymbol{F})-\min(\boldsymbol{F})}\right] * (\beta - \alpha) + \alpha$. When normalizing the values of $\boldsymbol{F}$ to the range [0,1], the following is performed: $\boldsymbol{F}' = \frac{(\boldsymbol{F}-\min(\boldsymbol{F}))}{(\max(\boldsymbol{F})-\min(\boldsymbol{F}))}$. Both strategies were applied to all clustering runs of the two dataset used here, and results found to be similar. However, it is important to note that the similar results are likely the result of the lack of outliers in the two datasets. As Min-max normalization is sensitive to outliers, the recommendation is to use variance normalization in these situations. Here the results from variance normalization are presented (Min-max normalization results are available on request).

Level-based games: *Tera* is a level-based and class-based MMOG, and this provides important information guiding analysis. Firstly, the level mechanic in the game means that player characters will start out with relatively low scores in all the features. As characters increase in level, scores will increase, but how much and at what rate depending on the choices the player makes. For example, some features are voluntary – the skills "Mining" and "Plants" are optional resource gathering skills the player can choose to develop, which requires time.

Different level bins were evaluated in order to determine the trending effect of character level, and a compromise between having as few bins as possible and managing the trend effect of level reached at four bins, each comprising 10 levels (1-10 [166,003 characters], 11-20 [48,270 characters], 21-30 [21,703 characters]) except one, which contains only data from the highest recorded level, 32 [4,241 characters]. The reason for binning the characters with the maximal level separately was based in part on their comparatively higher scores in the different features, as well as the end-game mechanics of most MMORPGs: These typically have a maximally attainable level. Upon reaching this, increasing in the relative power of the character occurs mainly via improving the equipment/items the character possesses. Binning data according to character level provides the added benefit of being able to evaluate the distribution of behavior clusters at different steps in the games´ progression arc.

Following level-based binning and dataset splitting, k-means and SIVM were applied to both datasets, with features normalized using Min-max and variance normalization. For k-means and SIVM, cluster runs from *k*=2 to *k*=24 were performed, and decisions about which number of clusters to accept based on mean squared error and Scree plots.

Additionally, cluster distributions were assessed manually to investigate the effect of adopting different values of *k*.

Ratio data: The metrics data for *BF2BC2* comprise a different perspective on player behavior as compared to the data from *Tera*. The presence of ratio data in the feature list (e.g. Kill/Death ratio) requires similar considerations. The problem observed here is that if a player has zero deaths or kills, the ratio value is zero. To avoid this issue, LaPlace Smoothing can be employed, increasing counts that determine the ratio by one [35]. This yields different results as Death and Kills would be different, however, especially for high values of the features in question, the effect of smoothing may not be problematic. In the current case, only 0.27% of the players had zero values for Kills or Deaths. It was therefore chosen to not employ smoothing, and it was not possible to detect an effect on the resulting clusters even if employed. For other ratios that could be useful when analyzing player behavior in *BF2BC2*, e.g. a Win/Loss (W/L) ratio, a substantial portion of the dataset would be affected, and in this situation smoothing might be necessary. Ratios like D/K or W/L are generally useful for evaluating player skill in games, and can be calculated as soon as a player has participated in one combat round in the case of *BF2BC2*. Skill-based estimates are crucial e.g. when it comes to matchmaking in online games [33].

VIII. Results and Discussion

A substantial amount of analysis was performed (multiple datasets, different bins, multiple k-values, different algorithms). Due to restrictions of space, not all results can be presented here. The focus will here be on describing the overall results backed by examples, and the applicability of the utilized algorithms for player behavior clustering (all results are available upon request).

As described in the above sections, the two algorithms employed here, SIVM and k-means, operate differently, broadly speaking focusing on cluster centroids vs. extreme values. This means that the two algorithms are useful for different purposes when it comes to behavioral analysis in computer games: K-means is useful for gaining insights into the general distribution of behaviors in a game´s population, whereas SIVM is useful for identifying players with extreme behaviors. The former is notably useful for checking asset use and game balance. For example, if k-means identifies clusters of players not utilizing particular features of the game, e.g. non-combat skills, this indicates that these features are under-used and that development resource is wasted. Similarly, identifying player clusters with low performance, typically those players at risk of leaving the game, and investigating what is causing the low performance assists with retention, an important consideration in any persistent-world game [37]. K-means is overall less useful for finding players exhibiting extreme behaviors, which means detecting subversive behaviors such as gold farmers, cheaters and bots can be difficult. This is where SIVM excels. Extreme behaviors are not only interesting to detect subversive play, but also for identifying players with attractive features such as big social networks. Such players are important in order to build and maintain a player community in a persistent-world game like *Tera*, and it is therefore in the interest of the developers to retain them in the game, e.g. by offering them special incentives [37]. The two algorithms thus support different goals of behavioral clustering in game development, supplementing each other.

It is important to note that hard cluster assignment was used for SIVM. This means that any player is assigned to one specific extreme behavior point, and can result in the formation of very large clusters because even players almost in the middle between two extremes will be assigned to the one closest. The size of the clusters should therefore be used as indicative of the relative distribution of players between the behavioral extremes. An alternative strategy is to use soft clustering; describing players in terms of the relationship to each extreme (archetype). For example, a *Tera* player might be 90% Elite and 9% Planter (remaining archetypes covering 1%). This approach is a topic for future investigation.

A. Behavioral classes in Tera

The difference between k-means and SIVM is exemplified in the behavioral dataset for *Tera*. This was divided into four bins, determined by character level (1-10, 11-20, 21-31, 32). For all of these, a relatively consistent 6-7 clusters offer the best fit, irrespective of whether k-means or SIVM is applied. For each algorithm, the clusters remain relatively constant throughout the level bins, with generally one cluster of players having the highest values across all or most features (the Elite), and another the lowest values (the Low Performers). Remaining clusters generally split into four groups: Two with middling scores, one better than the other, but low Plants and Mining skills; and two with comparable scores, but high Plants and Mining skills.

To take an example, and adopting the strategy of Drachen et al. [1] of converting the results of cluster analysis to descriptive behavioral profiles, the level 32 bin results for k-means (Table 2) indicate that the high level players in *Tera* exhibit fairly diversified behavior (results for the other three bins omitted due to space constraints, but will be reported in future work – results available upon request). A small group make up the **Elite** of the players, with the highest scores across all features, except for middling deaths from monsters (expected from elite players), and very low skill levels in Plants and Mining. This indicates players focused on performance, without interest in skills that do not impact on their performance (Plants and Mining provides access to resources and equipment, however, resources can also be obtained via solving quests or auctioning off found items). Contrasting is the **Stragglers**, the players with the lowest score for all features (including deaths from monsters), comprising 39.4% of the players. Next to the Low Performers are two clusters with successively better scores, **Average Joes** and **The Dependables** respectively, the latter with the highest scores except for the Elite. Both of these groups of players exhibit low Plants and Mining skills; however, they are matched by the last two groups, the **Worker I** and **Worker II**. These have scores similar to the Average Joes and The Dependables respectively, but with high Mining and Plants skill, and comparably higher loot values, i.e. they have looted more items.

The results for SIVM (Table 3) similarly identify the Elite and Stragglers behavioral profiles, as could be expected given that these are the most extreme performers identified by k-means, but the other four clusters exhibit more extreme behaviors than for k-means: **Planters** and **Miners** respectively have average scores across the performance features, but very high Planting or Mining skill respectively, reflecting players specialized in either of these skills. The **Auction Devils** represent players focused on using the auction house feature of the game and gaining achievements, apparently like to gain loot, and with strong social networks and high Mining skills (presumably in order to obtain resources for auctions). Finally, the **Friendly Pros** exhibit behaviors similar to the Auction Devils, but exhibit low Auction and Loot scores, and otherwise strong scores in the performance features. These two behavioral profiles are interesting because they basically reflect players focused on financial gain through any means available (looting, skill, auctions) and those without this drive (low loot score, few auctions started or bought, highest friend score).

TABLE 2: INTERPRETED BEHAVIORAL CLUSTERS FOR *TERA*, LEVEL 32 BIN ONLY, K-MEANS. %P = %PLAYERS IN BIN.

Title	%P	Characteristics
Elite	5.78	Highest scores for all features except Mining and Plants which are the lowest in the game.
Stragglers	39.4	Lowest scores for all features, including deaths from monsters.
Average Joes	12.7	Better scores across all categories than Low Performers, 4th ranked overall
Dependables	18.6	Average scores across all categories, high number of friends, 3rd ranked overall, 2nd rank in monster kills
Worker I	15.9	Similar to the Average Joes, but high Mining and Plants, and loot value 3rd ranked.
Worker II	7.6	Similar to The Dependables, but highest Mining and Plants value in the game. 2nd ranked overall. Loot 2nd ranked.

TABLE 3: INTERPRETED BEHAVIORAL CLUSTERS FOR *TERA*, LEVEL 32 BIN ONLY, SIVM. %P = %PLAYERS IN BIN

Title	%P	Characteristics
Elite	3.9	High scores overall, except for Mining/Plants and deaths from monsters. No auctions created.
Stragglers	7.6	Low scores overall, dies a lot from monsters
Planters	21.6	Middling scores, but high Plants skill
Miners	15.0	Middling scores, but high Mining skill
Auction Devils	1.1	Highest auction and achievement score. 2nd ranked loot and kills scores. 2nd ranked friends score, high mining score
Friendly Pros	50.8	Highest friends score, scores similar to Auction Devils apart from low auction score and 2nd lowest loot score

That only two of six clusters of players appear to spend time on learning non-combat skills could indicate a design problem for *Tera*. Resource-gathering skills like these are fundamental to the economy of a MMORPG, and with only roughly 25-35% (depending on the level bin) of the player base having high values in these skills, the flow of new raw materials may not be sufficient. Additionally, from a cost-benefit perspective, core gameplay features such as the non-combat skills should be utilized by most of the player base. The **Low Performer** class bears closer investigation as these are those with the highest risk of leaving the game, and finally the Elite behavioral profile and those with strong social networks (high number of friends) are of interest because retaining them in the game assists with ensuring a sustainable community [37]. On a final note, for reasons of available space we have not reported on whether the same players cluster similarly across the two algorithms and any variations across level bins; but this is a topic for work currently in preparation.

B. Behavioral classes in BF2BC2

The clustering analysis of the 22-feature dataset for BF2BC2 resulted in 7 clusters, with Scree plots and mean squared error indicating the same amount of cluster across SIVM and k-means. The most extreme behavioral patterns were consistent across the two algorithms (as they were in the *Tera* analysis): **Assassins**, who are characterized by extremely high Kill/Death ratios across all kits, and overall K/D ratio, the highest Kills per Minute (KpM) ratio, but low-middle playtime. Assassins represent the most lethal players in the game. Their overall skill rating is similar or slightly lower (for k-means) to the **Veterans**, however. Where the Assassins are specialized, Veterans display highest or second highest values across all features, including very high playtime and overall score values, indicating committed and highly stable players. They represent a small fraction of the players, however (Table 4,5). **Target Dummies** form the opposite of these two behavioral profiles, with lowest or very low values for all features, an comprise about 25% of the players (Table 4,5). They have not played the game for long, have low K/D ratios and middling Accuracy, and are highly susceptible to being killed, with the lowest Score per Minute (SpM) of any clusters (almost half the next-lowest cluster).

For the remaining clusters, SIVM results in profiles that exhibit a higher degree of difference, and a relationship with the four “kits” available to the players: K-means generally results in similar values across the four kits internally for each of the four remaining clusters, except one which has high scores for the Assault and Engineer kits. These clusters require close study to meaningfully separate (typically varying in one feature only), which is an example of a problem that can occur when employing algorithms searching for cluster centroids, i.e. that clusters become somewhat similar. In comparison, SIVM provides profiles focused on combinations of the four kits, identifying Assault-Recon-, Medic-Engineer-, Engineer-focus with very high Vehicle time (4 times higher than closest second), and Assault-focus as the four remaining clusters. These represent well two of the fundamental ways of playing *BF2BC2*: combat-oriented or support-oriented.

TABLE 4: INTERPRETED BEHAVIORAL CLUSTERS FOR *BF2BC2*, SIVM, %P = %PLAYERS IN SAMPLE.

Title	%P	Characteristics
Assault-Recon	1.4	High KpM and DpM, low Accuracy, average SpM, 2nd highest K/D overall.
Medic-Engineer	0.8	High Vehicle time, Skill level and Accuracy, 2nd highest SpM.
Assault "specialist"	5.0	Focus on Assault, but low score, high DpM and Playtime, low Skill, K/D and Accuracy.
Driver Engineers	1.1	Extremely high Vehicle time, high Playtime, Score and Accuracy, 2nd highest K/D, lowest DpM, low KpM.

Assassins	61.6	Highest K/D, high KpM, lowest Playtime, very low DpM.
Veterans	2.01	Highest Score, Playtime and Rounds played, overall high values
Target Dummies	28.1	Extremely low K/D, lowest Skill, SpM and KpM min. scores for all features but Playtime and Rounds played.

TABLE 5: INTERPRETED BEHAVIORAL CLUSTERS FOR *BF2BC2*, K-MEANS %P = %PLAYERS IN SAMPLE.

Title	%P	Characteristics
Snipers	7.4	Median SpM, overall low-middling values, high DpM, extremely high Accuracy. Highest kit score is Medic.
Soldiers	27.9	Median SpM, overall low-middling values, high DpM, highest kit score is Assault.
Assault-Engineer	13.1	Similar to Soldiers but better skill value, high Engineer and Assault scores and K/D ratios.
Target Dummies	26.0	Lowest scores for all values (including Playtime) except high DpM.
Trainee veterans	10.7	Comparable to Veterans, but 2nd rank in most features, and lower Playtime.
Assassins	10.9	Highest rank in all K/D-ratios, highest KpM, low Playtime, low DpM.
Veterans	4.1	High Playtime, 2nd rank in most features, highest overall Skill level.

IX. CONCLUSIONS

Successful clustering of player behaviors in contemporary major commercial ("AAA"-level) computer games is challenging due to the large scale and high dimensionality of telemetry data [1,12,15,33,34,38] and because of the requirement for behavioral clusters to be interpretable and actionable by game designers [5,13,34,37]. In the above, a strategy for behavioral profiling "in the wild" is suggested that relies on integrating knowledge of the design of the game being investigated in the feature selection and analysis process, and the use of two algorithms with different properties for obtaining results on general and extreme behaviors respectively, via k-means clustering and SIVM [22,26]. The specific strengths and weaknesses of each algorithm in terms of evaluating player behaviors described. The proposed strategy is evaluated using two case studies, representing to fundamentally different game designs, with a combined total of 260,000 player characters. Behavioral profiles were extracted and described to aid interpretability, and examples of how to apply them described.

REFERENCES

[1] A. Drachen, G. N. Yannakakis, A. Canossa and J. Togelius. Player Modeling using Self-Organization in Tomb Raider: Underworld. In Proc. of IEEE Computational Intelligence in Games, 2009.

[2] J. Bohannon. Game-Miners Grapple With Massive Data. Science, 330(6000):30-31, 2010.

[3] C.Thurau and C. Bauckhage. Analyzing the evolution of social groups in world of warcraft. In Proc. of IEEE Comp. Intelligence in Games, 2010.

[4] A. Cutler and L. Breiman. Archetypal Analysis. Technometrics, 36(4):338-347, 1994.

[5] A. Drachen and A. Canossa. Evaluating motion. Spatial user behavior in virtual environments. Int. Journal of Arts and Technology, v. 4 N3, 2011.

[6] N. Ducheneaut and R. J. Moore. The Social Side of Gaming: A study of interaction patterns in a Massively Multiplayer Online Game. In Proc. of the 2004 ACM Conf. on Computer supported cooperative work, 2004.

[7] L. Finesso and P. Spreij. Approximate Nonnegative Matrix Factorization via Alternating Minimization. In Proc. 16th Int. Symposium on Mathematical Theory of Networks and Systems, 2004.

[8] G. Golub and J. van Loan. Matrix Computations. Johns Hopkins University Press, 3rd edition, 1996.

[9] K. Isbister and N. Schafer. Game Usability. Morgan Kaufman, 2008.

[10] B. J. Jansen. Understanding User-Web Interactions via Web Analytics. Morgan & Claypool Publishers, 2009.

[11] I. Jollie. Principal Component Analysis. Springer, 1986.

[12] J. H. Kim, D. V. Gunn, E. Schuh, B. C. Phillips, R. J. Pagulayan, and D. Wixon. Tracking real-time user experience (true): A comprehensive instrumentation solution for complex systems. In Proc. of CHI, 2008.

[13] D. King and S. Chen. Metrics for Social Games. Presentation at the Social Games Summit, 2009.

[14] D. D. Lee and H. S. Seung. Learning the Parts of Objects by Non-negative Matrix Factorization. Nature, 401(6755):788-799, 1999.

[15] T. Mahlman, A. Drachen, A. Canossa, J. Togelius, and G. N. Yannakakis. Predicting Player Behavior in Tomb Raider: Underworld. In Proceedings of IEEE Computational Intelligence in Games, 2010.

[16] L. Mellon. Applying metrics driven development to MMO costs and risks. Versant Corporation, 2009.

[17] O. Missura and T. Gärtner. Player modeling for intelligent difficulty adjustment. In Proc. of the ECML-09 Workshop From Local Patterns to Global Models, 2009.

[18] P. Paatero and U. Tapper. Positive Matrix Factorization: A Non-negative Factor Model with Optimal Utilization of Error Estimates of Data Values. Environmetrics, 5(2):111-126, 1994.

[19] R. Pagulayan, K. Keeker, D. Wixon, R. L. Romero, and T. Fuller. User-centered design in games. In The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies, and Emerging Applications, pages 883-903. L. Erlbaum Associates, 2003.

[20] J.-K. L. R. Thawonmas, K. Yoshida and K.-T. Chen. Analysis of revisitations in online games. Journal of Entertainment Comp., 2011.

[21] R. Thawonmas and K. Iizuka. Visualization of online game players based on their action behaviors. Int. Journal of Computer Games Technology, 2008.

[22] C. Thurau, K. Kersting, and C. Bauckhage. Convex Non-Negative Matrix Factorization in the Wild. In Proc. IEEE Int. Conf. on Data Mining, 2009.

[23] C. Thurau, K. Kersting, M. Wahabzada, and C. Bauckhage. Descriptive matrix factorization for sustainability: Adopting the principle of opposites. Journal of Data Mining and Knowledge Discovery, 2011.

[24] B. Weber and M. Mateas. A Data Mining Approach to Strategy Prediction. In IEEE Symposium on Computational Intelligence in Games, 2009.

[25] G. N. Yannakakis and J. Hallam. Real-time Game Adaptation for Optimizing Player Satisfaction. IEEE Transactions on Computational Intelligence and AI in Games, 1(2):121-133, 2009.

[26] K. Kersting, M. Wahabzada, C. Thurau, and C. Bauckhage. Hierarchical Convex NMF for Clustering Massive Data. Proc. of ACML, 2010.

[27] G. Ostrouchov. On FastMap and the convex hull of multivariate data: toward fast and robust dimension reduction. IEEE Transactions on Pattern Analysis and Machine Intelligence, 27(8): 1340-1343, 2010.

[28] C. Fraley and A.E. Raftery. How Many Clusters? Which Clustering Method? Answers Via Model-Based Cluster Analysis. The Computer Journal, 41(8), 1998.

[29] Y. Zheng and X. Zhou: Computing with Spatial Trajectories. Springer, 2011.

[30] J. Miller, J. Crowcroft. Avatar Movement in World of Warcraft Battlegrounds. In Proceedings of IEEE Netgames, 2009.

[31] F. Southey, G. Xiao, R. C. Holte, M. Trommelen and J. Buchanan. Semi-Automated Gameplay Analysis by Machine Learning. In proceedings of AIIDE, 2005.

[32] T. Marsh, S. P. Smith, K. Yang and C. Shahabi. Continous and Unobtrusive Capture of User-Player Behavior and Experience to Assess and Inform Game Design and Development. In Proceedings of Fun and Games, 76-86, 2006.

[33] B. G. Weber, M. John, M. Mateas and A. Jhala. Modeling Player Retention in Madden NFL 11. In Proceedings of IAAI, 2011.

[34] G. Zoeller. Game Development Telemetry. In Proceedings of the Game Developers Conference, 2011.

[35] J. Han J. and M. Kamber. Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers, 2006

[36] C. Thompson. Halo 3: How microsoft labs invented a new science of play. Wired Magazine, 15(9).

[37] T. Fields and B. Cotton. Social Game Design: Monetization Methods and Mechanics. Morgan Kauffman Publishers, 2011.

[38] G. N. Yannakakis. Game AI Revisited. In Proceedings of ACM Computing Frontiers Conference, 2012.