Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 112 (2017) 2259–2268

Procedia Computer Science

www.elsevier.com/locate/procedia

International Conference on Knowledge Based and Intelligent Information and Engineering Systems, KES2017, 6-8 September 2017, Marseille, France

# Game Data Mining: Clustering and Visualization of Online Game Data in Cyber-Physical Worlds

Peter Braun[a], Alfredo Cuzzocrea[b,]*, Timothy D. Keding[a], Carson K. Leung[a,]*, Adam G.M. Padzor[a], Dell Sayson[a]

[a]*University of Manitoba, Winnipeg, MB, R3T 2N2, Canada*
[b]*University of Trieste and ICAR-CNR, 34127 Trieste (TS), Italy*

**Abstract**

Since its debut in May 2016, Overwatch has quickly become a popular team-based online video game. Despite the popularity of Overwatch, many new players—who join the game unsure how to compete with the game's veterans—feel overwhelmed with the vast knowledge required to properly play at higher skill levels. In this paper, a data mining algorithm is designed and developed for clustering and visualization of online game data at the cyber-physical world boundary. Scientifically, the algorithm uses affinity propagation for clustering and two-dimensional graphs for visualizing online game data. The algorithm analyzes the Overwatch game data for the discovery of new knowledge about current players and the clustering of data for each hero character. This knowledge enables the analysis of individual clusters and provides statistics that have a high correlation with winning player strategies. These statistics are expected to have a large influence on how a character is played, and thus can aid new players in learning their priorities as each hero character. In other words, the algorithm helps analyze the online game playing data, get insight about the grouping or clusters of players, and offer suggestions to new players of the game.

© 2017 The Authors. Published by Elsevier B.V.
Peer-review under responsibility of KES International

*Keywords:* Data mining, clustering, visual analytics, cluster visualization, cyber-physical world, online game, applications, innovative artificial intelligence technologies

* Corresponding author.
E-mail address: kleung@cs.umanitoba.ca (C.K. Leung)

1877-0509 © 2017 The Authors. Published by Elsevier B.V.
Peer-review under responsibility of KES International
10.1016/j.procs.2017.08.141

## 1. Introduction

*Overwatch*[†] is a team-based online multiplayer first-person shooter video game, in which players are assigned into two teams of six players. Each player selects one of 24 predefined characters called heroes. The player can take on the role of offense, defense, tank, or support.

Since its debut in May 2016, Overwatch has quickly become a popular team-based online video game. With its excellent balance of role-playing game (RPG) elements, and first-person shooter (FPS) combat, Overwatch has filled a unique spot in the gaming environment, in which its main competition appears to be *Team Fortress 2*[‡] (a game over 10 years old). The surging popularity of Overwatch generated a competitive league, with several highly-ranked competitive players forming sponsored teams. This popularity resulted in many new players joining, who may feel overwhelmed with the vast knowledge required to properly play at higher skill levels. One feature to note is that, while a levelling system exists, it does not affect the hero character one plays as, resulting in gameplay based primarily on skill and game knowledge.

Natural questions to ask are: “Can one do better than just relying primarily on skill and game knowledge?” and “Can additional information be extracted from current players through knowledge discovery?” Hence, our *key contribution of this paper* is that the design and development of an online game data mining algorithm that extracts additional information from current players by clustering data for each hero. This allows analysts to analyze individual clusters and visualize the statistics having a high correlation with winning rate. These statistics may have a large influence on how a hero character is played, and can aid new players in learning their priorities as each hero. For example, Zenyatta is a support hero who is able to deal out a large amount of damage in comparison to his support counterparts. Due to his mediocre healing capabilities, high-level players may focus more on dealing damage and eliminating enemies because they are aware that solely healing their team will not provide the support the team needs.

The remainder of this paper is organized as follows. Section 2 discusses related work. Section 3 gives background. Section 4 presents our data mining algorithm for clustering and visualizing online game data. Evaluation and conclusions are given in Sections 5 and 6, respectively.

## 2. Related work

The world of competitive gaming has a distinct advantage over typical competitive sports, in that it is much easier to record each individual statistic [1]. Analyzing the statistics of each hero has been done in the past. However, the target game has constantly shifted due the quick rise and fall of many popular games.

Let us take an online game called *League of Legends*[§] [2-6] as an example. In it, players have a similar selection of predefined champions. However, there are over 130 champions in League of Legends (cf. only 24 heroes currently in Overwatch). This means that there is far too much information to handle each hero manually. Existing website *Champion.gg*[**] is constantly analyzing how each champion is affected by each update in the game. It calculates the win rate in the current patch, and records statistics about changes (e.g., raise or fall) in the win rate. It also covers other aspects of the game, such as order that skills are used, trinkets chosen to improve performance, and the most common skill “trees” (i.e., a tiered system of skill sets earned by players). All these aspects of the game play an important role in how to use each hero efficiently.

In addition, there are a few recent studies on analyzing data from online game such as League of Legends. For instance, Ferrari [2] studied the generative and conventional play on the general multiplayer online battle arena and the specific one on League of Legends. Kou and Gu [3] tried to understand temporary teams in League of Legends when playing with strangers. Lee and Ramier [4] investigated the impact of game features on champion usage in

---

[†] https://playoverwatch.com/en-us/
[‡] http://www.teamfortress.com/
[§] http://na.leagueoflegends.com/
[**] http://champion.gg/

League of Legends. Kica et al. [5] analyzed the impact of patching on League of Legends. Kim et al. [6] used collective intelligence to make a strong team of players by predicting team performance in League of Legends.

Fortunately for Overwatch, despite only being a few months old, the website *Master Overwatch* [††] has accomplished similar results as the website Champion.gg. Specifically, the website Master Overwatch displays statistics which have risen/fallen over the past days. Despite seeing the statistics change (e.g., win rate, amount of kills), this does not necessarily mean that they are correlated. Perhaps the reason a hero is able to achieve a higher win rate is the result of another hero who synergizes well, and allows the hero more opportunities to achieve kills. Another possibility is that the time required to obtain a heroes ultimate ability has decreased, allowing them to achieve more kills as a result. Either way, while the website Master Overwatch displays the win rate and many other statistics, there is no proper correlation between them to determine win rate.

## 3. Background

Overwatch is a team-based online multiplayer first-person shooter video game. Overwatch assigns players into two teams of six, with each player selecting one of 24 pre-defined characters called heroes. Each hero has unique movement, attributes, and abilities, whose roles are divided into four categories:

- *Offense heroes* who have high mobility and are known for their ability to deal large amounts of damage. However, they have a low number of hit points.
- *Defense heroes* who excel at protecting specific locations and creating choke points. Some of them can also provide several means of field support (e.g., sentry turrets and traps).
- *Tank heroes* who have the most hit points out of all the characters in the game. Hence, they are able to draw enemy fire away from their teammates to themselves so as to disrupt the enemy team. Tank heroes also have various ways to protect themselves and their team with shield-like abilities.
- *Support heroes* who serve as utility characters that have abilities that enhance their own team or weaken the enemy team. They do not deal with a lot of damage, nor do they have many hit points, but they provide the buffs and debuffs to support their teammates to fight against their opponents.

Among the 24 heroes, there are seven offense heroes (e.g., Genji Shimada, Jesse McCree, Reaper), six defense heroes (e.g., Widowmaker), six tank heroes (e.g., D. Va), and five support heroes (e.g., Zenyatta). Players on a team work together to secure and defend control points on a map or escort a payload across the map in a limited amount of time. Players gain cosmetic rewards that do not affect gameplay, such as character skins and victory poses, as they play the game. The game was initially launched with casual play, with a competitive ranked mode, various 'arcade' game modes, and a player-customizable server browser subsequently included following its release. Additionally, Blizzard Entertainment—the company that created Overwatch—has developed and added new characters, maps, and game modes post-release, while stating that all Overwatch updates will remain free, with the only additional cost to players being micro-transactions to earn additional cosmetic rewards.

Blizzard Entertainment also hosts a server that provides information on how players play each hero. This server, *Play Overwatch*[‡‡], can be queried with usernames and their player data is summarized in the returned a hypertext markup language (HTML) page. Player statistics on every hero are returned in numerical format. However, this server only returns an individual player's statistics and how that individual player performs with each hero in the game. These statistics include total values accumulated throughout the player's career (total game time played), and average values with respect to that individual player's play history. The data returned by this server does not provide a means for the player to determine how they are performing relative to other players. In contrast, *a goal of the current research* is to provide a means for the player to determine how they are performing relative to other players by using data mining techniques of clustering and visualization.

[††] https://masteroverwatch.com/
[‡‡] https://playoverwatch.com/en-us/

Similarly, other websites such as Master Overwatch and *Overbuff*[§§] do provide users with running sum statistics for the player and each hero the player has used competitively, average statistics relative to the individual player, as well as inter-player statistics. By using their large user base, these websites can likely provide to its users statistics about how they are playing the game relative to global player statistics. When using these websites, a player can determine how they are playing the game relative to themselves; and whether or not they are underperforming or excelling in a particular dimension in the game. For example, a player can use these websites to understand how they are performing when selecting the hero Zenyatta. In addition, these websites also notify the player if the user is doing more healing than the global average. Furthermore, they rank their users so that a certain skilled player is performing among the top 10% of a given statistic. While these websites provide a summary of a player's career and performance relative to the global player base, they do not provide an insight of how these statistics affect winning. In contrast, *another goal of the current research* is to provide an understanding of how these statistics affect winning by using data mining techniques of clustering and visualization.

Our algorithm aims to cluster player data. Specifically, it segments players into similar groups based on player style for a particular hero. As existing research does not offer this service, our algorithm will provide a better understanding of how hero statistics and player performance correlate to win rate.

From a list of playstyle clusters, a small subset of these clusters are chosen based on high win percentage for a hero. There is a guard against excelling individuals—exceptional players that are the sole member of their playstyle clusters—because these players play so differently from other players that perhaps these players are uniquely skilled in how they play. Instead, small groups of skilled individuals that play alike are chosen. From these groups, their statistics and averages in those statistics can be plotted against other clusters. By comparing how high performing clusters perform on each statistic, on each hero, relative to lower performing clusters, insight can be obtained about what these high performing individuals are doing that other players are not that lead them to more games won.

As a preview, an example that will be outlined in Section 5 is that, given some statistics (e.g., dimensions in terms of the clustering algorithms), underperforming in certain aspects of the game may lead to higher win rate such as the healing performance of high performance players when using the hero Zenyatta. This comparison between high win-rate playstyles and low win- rate playstyles can be used by players to understand that Zenyatta is not meant to be the primary healer of team. High performance players do not use Zenyatta to recover from significant amounts of damage. With this knowledge, players can play Zenyatta with less emphasis on healing and improve their performance.

## 4. Clustering and visualization algorithm

In this section, the clustering and visualization algorithm is described. The key idea of the algorithm is to segment players into similar groups based on player style for a particular hero using affinity propagation. More specifically, key steps of the clustering algorithm are described below.

### *4.1. Obtaining the users and data*

As the makers of Overwatch do not seem to have released an official application program interface (API) allowing code level access to the player performance database, these data must be collected by polling the servers like Play Overwatch for user data. Hence, the first logical step is to obtain a list of Overwatch players.

Existing works indirectly provide this information because they provide a ranking service to their users. Because these websites list players as a service, a player list can be obtained. This list can then be used to poll the Play Overwatch server.

Similarly, as there does not seem *to be* any official API for the game's player's statistics database, manual querying using the obtained player list is needed. To obtain the data, an iOS app known as *SiteSucker*[***] is utilized in

[§§] https://www.overbuff.com/
[***] http://ricks-apps.com/osx/sitesucker/

order to obtain a large list of all individual players across all platforms. Consequently, individual players on the player list can be requested and the returned web page can be processed to extract pertinent data.

*4.2. Preprocessing the data*

The data returned by the web page tends to be incomplete because some players may have records in certain statistics while others do not. For example, for the hero Reaper, some players have no record of killing other players with the attack Death Blossom. These data are also sometimes wrongly labeled. For example, the statistic "Cards" may appear on some web pages as "Card". Similarly, other fields have been encountered with similar plurality errors.

The data returned from the website should be stored in an unprocessed state until the developer is able to provide for every user a complete set of statistics and a way to replace all misnamed dimensions. Only until a complete and repetitively plural-consistent dataset is made should a developer try to mine the player data.

The data returned by the server are also not purely numeric and steps should be taken to transform such data into numeric values in preparation for mining. For example, multiple numeric values contained a comma for values over one thousand. While this may look elegant for display, it can cause issues when converting a value to a float.

To solve the aforementioned problems (e.g., missing data, wrong data, errors), the second key step of the clustering and visualization algorithm is to adapt an intelligent computational model [7, 8] for quality assurance. Here, the model captures and analyzes historical data. To deal with missing data, the model fills the gaps by finding similar sequences from historical data. To deal with noisy data (e.g., wrongly labeled data), the model first examines historical data to learn the norm and then detects any anomalies [9] that are deviated from the norm. To transform character strings into equivalent numeric formats, the model reads the strings and removes thousand separators (e.g., comma, period, or thin space between groups of thousands) or delimiters. In addition, the model is also designed and developed to handle high volumes of a wide variety of data types in such a way that it can also discretize numeric data into equi-width or equi-depth intervals. Moreover, the model is also designed and developed to handle a wide variety of data source in such a way that it can also integrate data from different sources (say, statistical data from more than one websites or servers). Through this preprocessing step (via data cleaning, data integration, data transformation and data discretization techniques), data are now cleaned, integrated, transformed and complete, and thus ready for mining.

*4.3. Mining the data*

After obtaining a complete and repetitively plural-consistent dataset, the raw data can be mined by adapting an *affinity propagation* technique [10]. Although the clustering and visualizing algorithm uses affinity propagation to group similar data, the algorithm is designed and developed in such a flexible way that other clustering techniques—such as *k*-means or *k*-medoid portioning, hierarchical, density-based, grid-based, model-based, or constraint-based techniques—can be used. Moreover, while a *scikit-learn*[†††] implementation of affinity propagation is adapted, other implementations can also be used.

Here, affinity propagation aims to group similar objects and forms clusters by sending messages between pairs of data points until they converge so that each cluster is identified by its most representative data point in the cluster. The messages sent between pairs of data points represent the suitability for one data point to be a representative of other data points in the clusters. More specifically, the following are two types of messages sent between data points:

(a) responsibility $r(i, k)$, which is the accumulated evidence that sample data point $k$ should be the representative for sample data point $i$, can be computed in Equation (1):

$$r(i, k) = s(i, k) - \max\{a(i, k') + s(i, k') \mid k' \neq k\} \tag{1}$$

where $s(i, k)$ is the similarity between sample data points $i$ and $k$; and

[†††] http://scikit-learn.org/stable/modules/clustering.html

(b) availability $a(i, k)$, which is the accumulated evidence that sample data point $i$ should choose sample data point $k$ to be its representative after taking in account the values of all other data points that consider $k$ as their representative, can be computed in Equation (2):

$$a(i, k) = \min\{0, r(k, k) + \sum_{i' \text{s.t. } i' \notin \{i,k\}} r(i', k)\}. \quad (2)$$

Initially, $r(i, k) = 0$ and $= a(i, k) = 0$ for all pairs of $i$ and $k$. Values of data points are updated in response to values from other pairs according to the above equations. This updating process is carried out iteratively until convergence. In other words, this can be considered as an iterative-refined clustering process.

In terms of time complexity, this affinity propagation technique takes $O(N^2T)$ where $N$ is the number data points in the dataset and $T$ is the number of iterations until convergence. In terms of space complexity, when using a dense similarity matrix to represent and store the data points, this affinity propagation technique requires $O(N^2)$. This can be reduced when a sparse similarity matrix is used to represent and store the data points.

### *4.4. Finding the high win-rate clusters*

The next step is to organize the dataset into classes containing members belonging to a cluster. These classes can then be directly used knowing entries belong to a particular cluster. For instance, to find top-$K$ clusters and $m$ members (say, $K$=4 top clusters and each with at least $m$=8 members), the clustering and visualization algorithm performs the following. It first ignores clusters with fewer than $y$ members because it aims to look for high-performance teams with multiple players (where the number of players $\geq m$) exhibiting similar playstyles. This will lead to high win percentage. The idea is that, if multiple people exhibit this playstyle, it is a playstyle that can be performed by non-unique players, and can be emulated by the average user to improve their gameplay. The algorithm then ranks the resulting clusters having at least $y$ members according to the win rate. The top-$K$ clusters with high win-rate ($K$ high performance clusters) are then returned to the users.

### *4.5. Visualizing every dimension and identifying differences in playstyles*

Once the top-$K$ high performance clusters identified, the clustering and visualization algorithm generates plots for identification of differences between how high performance clusters perform for each dimension with respect to other clusters/playstyles. Specifically, for every dimension, the cluster index runs along the $x$-axis, while the current dimension runs along the $y$-axis. *Individual player values*—as indicated by the "×" markers—in the current dimension from the dataset are plotted. The *global average value* for the particular dimension is indicated by a horizontal line and can serve as a reference point for cluster averages. Cluster averages for the dimensions are indicated by the "+" markers and can be compared to the global average and the averages of other clusters to gain insight into how clusters perform relative to one another.

Note that a higher than average performance in a single dimension is not necessarily indicative of high performance. It is possible that underperforming a particular dimension may lead to higher win rates. The plots help players to think if they should lower their effort in a particular dimension for a hero, or perhaps practice to improve in that dimension. Moreover, players can also learn and avoid lower performance cluster playstyle.

## 5. Evaluation

As a case study for evaluating the clustering and visualization algorithm presented in this paper, statistics of a particular hero—namely, Zenyatta—is chosen for analysis and evaluation due to his high-skill requirement. By choosing a hero who requires a large amount of skill to play, it is easier to distinguish the poor players from the excellent ones. Two of the authors have a fair amount of personal experience playing as Zenyatta, and they can provide their expertise on validating some hypotheses.

Zenyatta is a support hero, who focuses on healing his teammates while remaining in the back where he is safer. His healing is mediocre compared to other support heroes, and his true advantage is being able to mark opponents. This mark allows all teammates to deal an additional 30% damage to the target. Zenyatta can also deal far more damage than the other supports, giving him a delicate balance of offense and support capabilities.

### *5.1. Average healing done*

Looking at the resulting data plotted in Fig. 1, the highlighted columns are clusters with notably higher win rates. With the average healing done per match evening out around 9000, it can be easily observed that all clusters with higher win rates tend to fall below the average amount of healing. With Zenyatta's high damage output and mediocre healing, it may prove more beneficial to focus on dealing damage. If this hypothesis is correct, then looking at the chart for average damage done, the clusters with a high win rate should be above average as well.

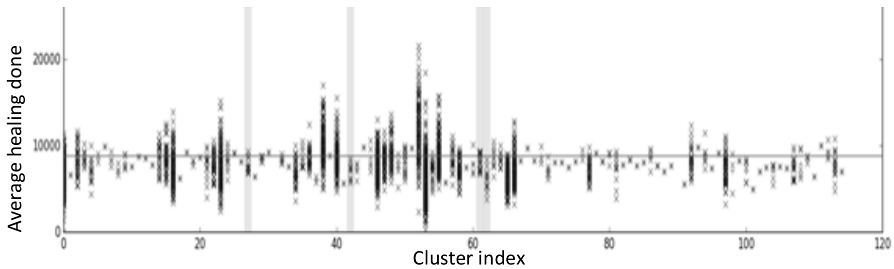


Fig. 1 Average healing done.

### *5.2. Average damage done*

It can be observed from Fig. 2, which shows the average damage done, that the clusters with higher win rates tend to place above average. It is interesting to note that there appears to be a much greater range in damage done when compared to the average healing done.

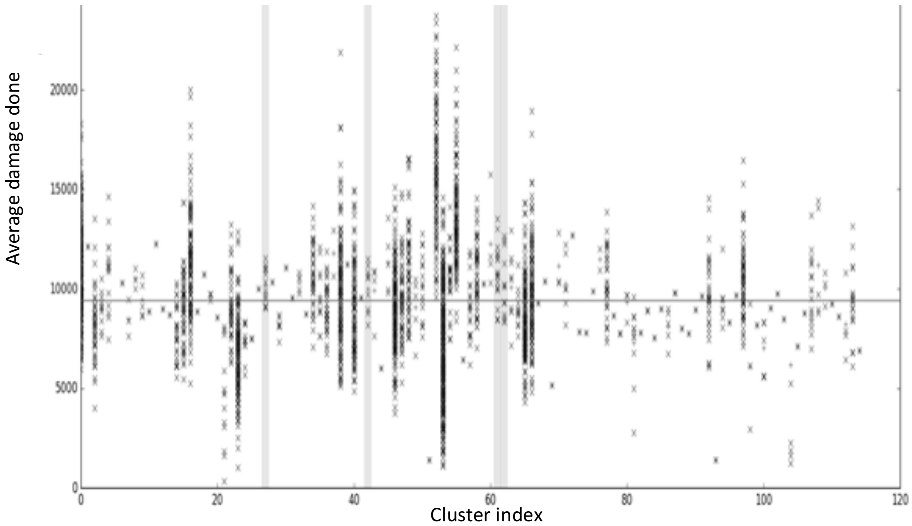


Fig. 2 Average damage done.

Experienced players note that the players with damage far above average are likely to be players who want to play a damage role but are forced to pick a support due to a poor team composition. Hence, they likely have below average healing. On the other end of the spectrum, players with below average damages may have a lower win rate because they are focusing too much on healing, or perhaps they simply have poor accuracy.

The clustering and visualization algorithm helps analysts to verify their hypothesis that "players with higher win rate tend to heal less in exchange for more damage output". This also is reflected in plots for other dimensions such as average eliminations, offensive assists, and defensive assists.

### *5.3. Average objective time*

Fig. 3 on the objective time shows that clusters with notably higher win rate fluctuate quite a bit above and below the average. The average objective time is about 75 seconds. As a support hero, Zenyatta's average is similar to the average of many offense heroes who stay on the objective when necessary, but will often leave the objective to pursue an enemy. Tank heroes also tend to spend more time on the objective and keep the support safe. Based on this information, it is beneficial for players as Zenyatta to stay on the objective to be protected by the tank heroes. Although it is not a necessity for his playstyle, player may learn that leaving the point to place range between Zenyatta and his opponent, running for cover, or attempting to finish an enemy could be an alternative strategy for game playing as Zenyatta.

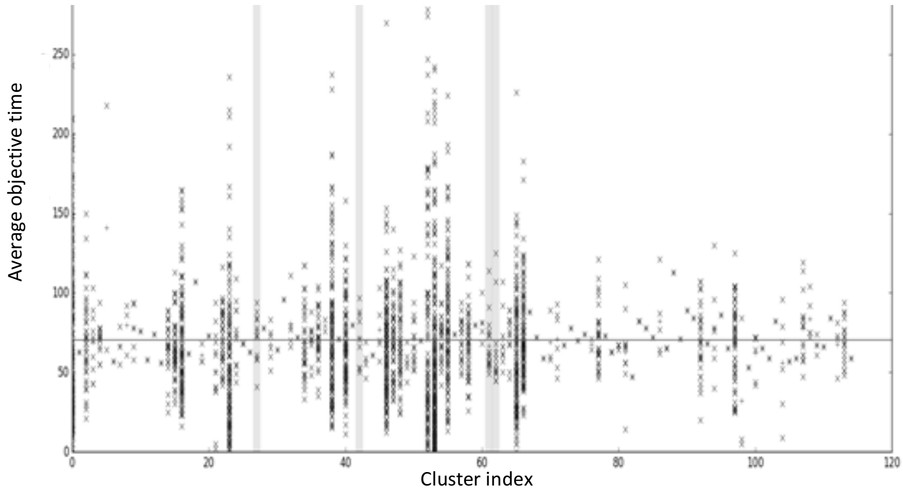


Fig. 3 Average objective time.

### *5.4. Weapon accuracy*

Fig 4—which shows the weapon accuracy—reveals that a notably higher win rate tends to have higher weapon accuracy. This strengthens an earlier finding that "higher win rate players tend to focus more on eliminations than healing". Accuracy is required in order to kill opponents. While the average accuracy is 30%, notable clusters range from 30%-40%, and a 10% difference can make a significant difference when targeting the enemy.

The figure also reveals that players with a higher win rate seem to be below average critical hit accuracy (which requires hitting the opponent in the head). This means that these players focus less on hitting head shots instead of focusing simply on hitting the opponent.

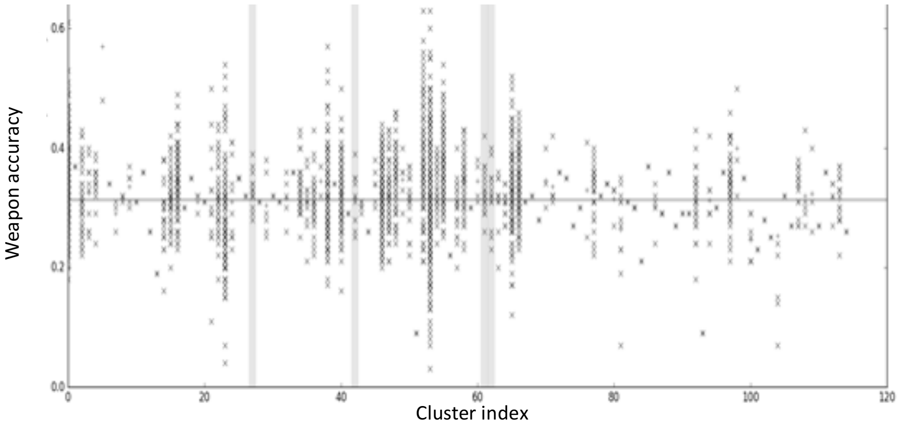


Fig. 4 Weapon accuracy.

### 5.5. Most solo kills in a game

Fig. 5—which shows the most solo kills in a game—reveals that players with a higher win rate seem to show little difference from other players, ranging between three to six kills on average. With such a low average, this only emphasizes that, as a support hero, Zenyatta tends to stay near his teammates for protection and to provide support. He is not a hero who should be circling around the enemy and harassing them from behind, as he likely will not survive the encounter.

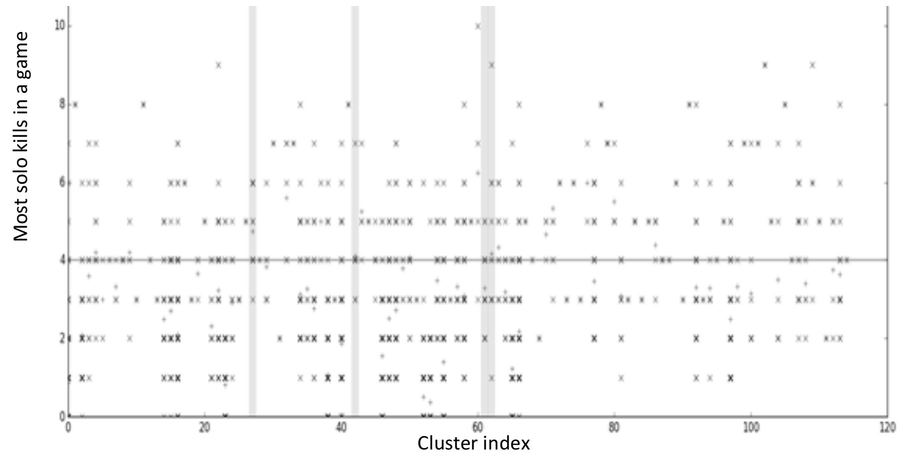


Fig. 5 Most solo kills in a game.

### 5.6. Evaluation summary

By using the clustering and visualization algorithm presented in this paper, analysts and users get more insights about game playing strategies. Specifically, clustering gives a proper summary of how players with above average win rates tend to play Zenyatta. Most high-ranked players tend to stay close to their teammates and attack the opponents from the backlines, providing healing support when necessary. Being in the back lines, they prioritize hitting the enemy over hitting head shots, as head shots become increasingly difficult with distance. Due to Zenyatta's lack of mobility options, players with higher win rates will stay on the objective when they are likely supported by teammates, but will not hesitate to step off of the objective in order to duck under cover, away from enemy projectiles.

## 6. Conclusions

While previous works were able to calculate the win rate of a hero, they were designed to make proper correlations between hero statistics and win rate. In this paper, a data mining algorithm was designed and developed to analyze the Overwatch data for the discovery of new knowledge about current players and clusters data for each hero. This knowledge allows analysts to analyze individual clusters and visualize those statistics have a high correlation with a winning rate. These statistics are expected to have a large influence on how a character is played, and thus can aid new players in learning their priorities as each hero. In other words, the algorithm helps analyze the online game playing data, get insight about the grouping or clusters of players, and offer suggestions to new players of the game.

Evaluation results from the case study gives a broad understanding of how each hero is being played, what results in higher win rates, and what an average player does in comparison to a player with a notably higher win rate. From observing Zenyatta's data, some surprising information (e.g., head shots having a low priority) are found. In comparison, they are a necessity for heroes such as Jesse McCree and Widowmaker to succeed.

While our current study helps *individual* players to maximize their wins, ongoing work is to incorporate the mined knowledge to maximize the winning of a *team*. Another ongoing work is to explore the relationships between players' social network data [11-13] and uncertainty or predictions of their winning rates [14-16]. Finally, *big data issues* [17, 18], for instance via machine learning and clustering, are planned to be considered in the near future.

## Acknowledgements

This project is partially supported by NSERC (Canada) and University of Manitoba.

## References

[1] Leung CK, Joseph KW. Sports data mining: predicting results for the college football games. *Procedia Computer Science* 2014; **35**: 710-719.

[2] Ferrari S. From generative to conventional play: MOBA and League of Legends. In: *Proceedings of the DiGRA 2013*, DiGRA; 2013.

[3] Kou Y, Gui X. Playing with strangers: understanding temporary teams in league of legends. In: *Proceedings of the ACM CHI PLAY 2014*, ACM; 2014, p. 161-169.

[4] Lee C, Ramler I. Investigating the impact of game features on champion usage in League of Legends. In: *Proceedings of the FDG 2015*, SASDG; 2015, paper 15.

[5] Kica A, La Manna A, O'Donnell L, Paolillo T, Claypool M. Nerfs, buffs and bugs - analysis of the impact of patching on League of Legends. In: *Proceedings of the CTS 2016*, IEEE; 2016, p. 128-135.

[6] Kim YJ, Engel D, Woolley AW, Lin JY, McArthur N, Malone TW. What makes a strong team? Using collective intelligence to predict team performance in League of Legends. In: *Proceedings of the ACM CSCW 2017*, ACM; 2017, p. 2316-2329.

[7] Leung CK, Mateo MAF, Nadler AJ. CAMEL: an intelligent computational model for agro-meteorological data. In: *Proceedings of the ICMLC 2007*, IEEE; 2007, p. 1960-1965.

[8] Mateo MAF, Leung CK. CHARIOT: a comprehensive data integration and quality assurance model for agro-meteorological data. In: *Data Quality and High-Dimensional Data Analysis*, WSPC; 2009, p. 21-41.

[9] Leung CK, Thulasiram RK, Bondarenko DA. An efficient system for detecting outliers from financial time series. In: *Proceedings of the BNCOD 2006*, Springer; 2006, p. 190-198.

[10] Frey BJ, Dueck D. Clustering by passing messages between data points. *Science* 2007; **315**(8514), p. 972-976.

[11] Braun P, Cuzzocrea A, Leung CK, Pazdor AGM, Tran K. Knowledge discovery from social graph data. *Procedia Computer Science* 2016; **96**: 682-691.

[12] Kawagoe K, Leung CK. Similarities of frequent following patterns and social entities. *Procedia Computer Science* 2015; **60**: 642-651.

[13] Leung CK, Tanbeer SK, Cuzzocrea A, Braun P, MacKinnon RK. Interactive mining of diverse social entities. *International Journal of Knowledge-based and Intelligent Engineering Systems* (*KES Journal*) 2016; **20**(2): 97-111.

[14] Budhia BP, Cuzzocrea A, Leung CK. Vertical frequent pattern mining from uncertain data. In: *Proceedings of the KES 2012*, IOS Press; 2012, p. 1273-1282.

[15] Chowdhury NK, Leung CK. Improved travel time prediction algorithms for intelligent transportation systems. In: *Proceedings of the KES 2011, Part II*, Springer; 2011, p. 355-365.

[16] Leung CK, Mateo MAF, Brajczuk DA. A tree-based approach for frequent pattern mining from uncertain data. In: *Proceedings of the PAKDD 2008*, Springer; 2008, p. 653-661.

[17] Cuzzocrea A. Accuracy control in compressed multidimensional data cubes for quality of answer-based OLAP tools. In: *Proceedings of SSDBM 2006*, IEEE; 2006, p. 301-310.

[18] Cuzzocrea A. Privacy and security of big data: current challenges and future research perspectives. In: *Proceedings of CIKM-PSBD 2014*, ACM; 2014, p. 45-47.