Automated clustering of video games into groups with distinctive names

Nicolas Grelier[0009−0009−8594−6113] and Stéphane Kaufmann

Pullup Entertainment, France
{nicolas.grelier,stephane.kaufmann}@pullupent.com

Abstract. When doing a study on a large number of video games, it may be difficult to cluster them into coherent groups to better study them. In this paper, we introduce a novel algorithm, that takes as input any set of games S that are released on Steam and an integer k, and cluster S into k groups. Each group is then assigned a distinctive name in the form of a Steam tag. We believe our tool to be valuable for gaining deeper insights into the video game market. Our algorithm consistently achieves high scores on an objective function that we introduce, the naming score, which assesses the quality of a clustering and how distinctive its name is.

Keywords: Steam tags · naming score · Cohen's h.

1 Introduction

In this paper, we are interested in understanding better the video game market. Let us imagine a scenario where Alice, who could be a researcher, a game developer, or working in a publishing company, faces a large number of games. She might be looking for relevant games for an academic study, or trying to find games similar to the one she is currently developing, to serve as a benchmark. The set of games could be all video games, or could be more specific: for instance all *Racing* games, or all *Puzzle* games released since 2018 that have at least 1000 reviews on the video game platform Steam.

Obviously, even as an expert, Alice only played a few of the games in the large set. She might have never heard of most of the games, especially those that did not sell well. Still, she wants to better understand this set. It may be that some games are actually not relevant for her purposes, which she could then dismiss. Perhaps she can learn that these games, which she thought were all similar, can actually be split into several coherent groups of games. This information may help Alice to do market segmentation [1], [2], since knowing the types of games that exist within a segment helps in understanding better the types of players. In this paper, we provide an algorithm for automatically splitting the games into groups with distinctive name, whatever the set of games with which Alice started. Note that our algorithm autonomously discovers meaningful ways of splitting the games. If one already knows that they are interested in a specific subgroup of the games, then it is straightforward to filter the set of games to those only containing a specified tag. However, in our scenario, Alice does not initially know which tags, among the 448 available Steam tags, would be appropriate for splitting the games into relevant groups. Our method allows for an unbiased discovery of the subgroups within a set of games without relying on predefined tags.

To the best of our knowledge, there do not exist clustering algorithms for this specific task. One issue with video games clustering is that there do not seem to be ways of splitting games into groups, such that each game would clearly belong to a unique group, even when using genres [3, 4]. This is what one hopes to do with classical clustering, but we believe that for video games, one has to relax this desired property that a game should be as different as possible from the games in the other clusters. Thus, we introduce a new clustering objective. Our objective is to find an algorithm that could split a set of games into groups, and assign a name to each group. All games within a group should relate to its name. Moreover, the name should be as distinctive as possible, so that it gives information about the group. When facing different options, meaning that the names of several clusters relate to one game, we are fine with any arbitrary choice: We do not require that a game G put into cluster C would have nothing to do with the name of cluster C'. We only require that the name of cluster C applies to G. In other words, we aim at high intra-cluster similarity, but lowering the inter-cluster similarity matters significantly less to us (we still want the name of a cluster to be distinctive enough from the rest of the games). Note that we do not require cluster names to be genres.

We want an algorithm that could partition any set of games into groups with distinctive names, using an explainable method, and that would reflect how players think. In this paper, we rely on data from the video game platform Steam, obtained through SteamSpy's API. On Steam, everyone can assign tags to games. As of November 2023, there are 448 tags assigned to the 66908 games available on Steam. Among the tags, one can find genre tags like *Adventure* or *RTS*, and many other tags like *2D*, *Funny*, *Cats* or *Medieval.* SteamSpy provides a database with for each game the twenty most assigned tags to it, along with how many players assigned those tags. We use those values for the clustering algorithms we test in this paper, and our cluster names are tags. We believe that the methodology outlined in this paper could be applied to various entertainment mediums, such as music and films. However, one would need a database of user-generated tags, which to our knowledge only Steam provides.

1.1 Related work

To the best of our knowledge, this is the first study about the problem of clustering games into groups with distinctive names, where the name of a cluster should apply to all the games it contains, but allowing that a game might have been placed in another cluster. Previous works about clustering games include clustering by genres through survey [4], by game traits through survey [5], by players behaviours [6], based on how algorithms play games [7, 8], and based on characteristics for video streaming [9]. Note that a major difference between these works and ours is that we want a program that could then be applied to cluster any set of games, whereas previous works aim at clustering all video games for good.

1.2 Our contributions

We introduce in Section 2 a new function to assess the quality of a clustering of video games. To maximise this function, the clustering must first have high intra-cluster similarity, and with less importance have low inter-cluster similarity, such that it is possible to give a name to each cluster of games that is distinctive from the other games. We call this function the *naming score* of a clustering.

We provide in Section 3 a clustering algorithm that has significantly larger naming scores on several testing sets than simply applying a K-means algorithm, with and without Principal Component Analysis (PCA).

2 The naming score of a clustering

We are given a set of games S that we want to partition into k clusters. Each cluster should have a distinctive name. Our idea is that we should be able to assign a tag t to each cluster C, such that t is assigned to most games in C and to few games in $S \setminus C$.

It is important to note a distinction between our clustering objective, and what is generally desired when doing clustering of some data points. Usually, it is wanted that a point should belong well to its cluster, and should be different from the points in the other clusters. This leads for instance to the well-known silhouette method to determine the number of clusters [10]. However, in our case, it is not an issue that a game might have been put in another cluster. We just want that the name of its cluster does strongly relate to the game. This is why we introduce a new objective function.

2.1 Intuitive definition

Let us consider one cluster C of a clustering of a set of games S. We have access to the tags that are assigned by players to these games. Our idea is to find the most over-represented tag t among those, and to name C as the cluster of games that have the tag t. However, it is a priori not clear how one can say whether a tag is over-represented.

For this purpose, we use Cohen's h [11]: a measure of difference between two proportions, in this case the proportion of games in cluster C that have the tag t compared to the proportion of games in the set S that have the tag t. The exact formula for computing Cohen's h is given below. The reason why we use Cohen's h instead of, for instance, a simple ratio between the two proportions is as follows: Imagine there is a tag t that is assigned to 0.001% of the games in S, and to 0.1% of the games in C. The increase in proportion is a multiplicative factor of 100, but still the proportion of games in C that have t is only 0.1%.

Thus, it would be inappropriate to name C as "the group of games that have the tag t". Cohen's h solves this issue by only giving a large value when the new proportion is both larger than the previous one, and the new proportion is in itself large.

With Cohen's h, we have a method of assessing whether a tag is over-represented in a group of games C. Therefore, we name C as the "the group of games that have the tag t", where t is the tag with highest Cohen's h value. We define the naming score of C as the Cohen's h value of this tag t. Finally, we define the naming score of a clustering of S as the weighted average naming score over all clusters, where the weight of a cluster C is $|C|/|S|$. We do weighted average, for otherwise one could obtain a clustering with an extremely high naming score by putting nearly all games in one cluster (which would have a naming score close to 0) and only one game in each other cluster (with a naming score close to π, i.e. the highest value Cohen's h may take). However, we want clusters to be roughly of the same size. By taking a weighted average, our naming score function may be large only if there is no large cluster. We give below a formal definition of this naming score.

2.2 Formal definition

Let us consider a set S of video games and an integer k. The problem consists in partitioning S into k subsets, such that for each subset C in S, there exists a tag t that is distinctive to C. By "distinctive", we mean that t is assigned to almost all games in C, and to few games in $S \setminus C$.

Let C be a subset of video games in S (potentially S itself). We denote by T_C the set of tags that are assigned to at least one game in C. For a tag t in T_C, we denote by $p(t, C)$ the proportion of games in C to which t is assigned. We denote by $h(t, C)$ Cohen's h applied to the proportions $p(t, C)$ and $p(t, S)$, i.e. $h(t, C)$ is a measure of difference between the proportion of games in C that have the tag t, and the proportion of games in S that have the tag t. It is defined as $h(t, C) := \phi(p(t, C)) - \phi(p(t, S))$, where $\phi(x)$ equals $2 \arcsin \sqrt{x}$ [11]. Observe that the maximal value of Cohen's h is π. A greater Cohen's h value indicates a greater over-representation of the tag t among the games in C compared to the games in S.

We define the naming score $n(C)$ of a subset C of video games in S as $n(C) := \max_{t \in T_C} h(t, C)$. Let $\mathcal{C} = \{C_i\}_{1 \leq i \leq k}$ be a partition of S. We make an abuse of notation and denote by $n(\mathcal{C})$ the naming score of $\mathcal{C}$, which we define as $n(\mathcal{C}) := \frac{1}{|S|} \sum_{1 \leq i \leq k} n(C_i) \cdot |C_i|$.

3 Finding the best clustering algorithm

In this paper, we use the classical K-means algorithm from the scikit-learn Python library to do the clusterings. We show that a modification of the Steam database (selecting only a few tags, weighting tags differently) significantly improves the naming score. We show step by step how we were able to improve the naming score by doing one modification after another. To speed up the running time of the K-means program, we only consider the 4065 games that have at least 2000 reviews on Steam.

3.1 Priority helps

In [12], a notion of priority of a Steam tag was introduced by Grelier and Kaufmann. For a game G, where the most assigned tag $t_{\max}$ was assigned by $n_{\max}$ players, the priority of a tag t that was assigned to G by n players is defined as $n/n_{\max}$. If t is not assigned to G, then its priority is 0. The authors argued that the priority is a measure of essentiality of tags to describe games. They indicate that for a game G, tags with high priority are essential to describe G, and tags with low priority give information of less importance [12].

In this subsection, we show that priority helps to cluster games into groups with distinctive names. We compare two databases. In both, there is one entry per game, and as many columns as there are Steam tags. In the first database $\mathcal{D}_1$, there is a 1 in the column of the tag t for the game G if the tag t is assigned to G, and a 0 otherwise. In the second database $\mathcal{D}_2$, the value at column t for entry G is the priority of the tag t with respect to the game G.

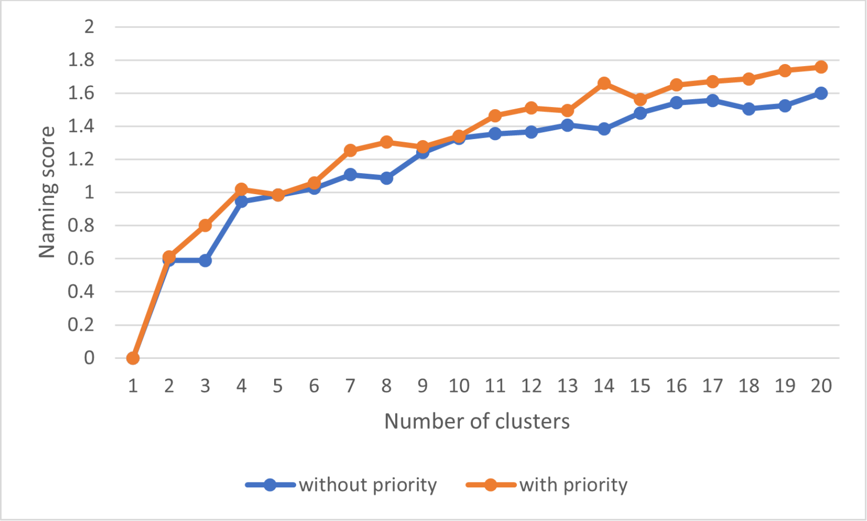

Fig. 1. Comparison of the naming scores of the clusterings depending on whether the priority is taken into account.

We compare the results of the K-means algorithm from the scikit-learn Python library applied to the two databases $\mathcal{D}_1$ and $\mathcal{D}_2$. Surprisingly, doing a Principal Component Analysis (PCA) [13] only lessened the naming scores. Thus, we apply the K-means algorithm directly to the databases for the rest of this paper. Figure 1 indicates that the naming scores of the clusterings are always larger using $\mathcal{D}_2$ than when using $\mathcal{D}_1$. Therefore, throughout the rest of the paper, we keep using the database $\mathcal{D}_2$.

3.2 Restricting the set of tags

We have the intuition that the information contained in the Steam tags can be summarised using only a few tags. Our working hypothesis is as follows:

Working Hypothesis 1 *Having too many tags for the clustering brings noise. There exists a small set of tags that summarise all the important information contained in the Steam tags, which leads to better clusterings.*

We build upon [14], where the authors presented a set of seven tags, that they call the *capital* tags, that encompass all others. Those tags are *Singleplayer*, *Multiplayer*, *Action*, *Casual*, *Adventure*, *Strategy* and *Anime*. Those tags where obtained by considering the tags that were correlated with many other tags, and that are assigned the most frequently. In [14], the authors say that a group of tags $\mathcal{T}$ *covers* a set of games S if for each game G in S there is at least one tag in $\mathcal{T}$ assigned to G. They observed that the capital tags cover 94% of all the games on Steam.

We refer now to those capital tags as being of rank 1. To obtain the capital tags of rank 2 and 3, we first removed the capital tags of rank 1 from the database and then iterated the same process. Thus, to obtain the capital tags of rank 2, we did an exhaustive search to find how to cover as many games as possible while using only a few tags. The number of games was chosen empirically, when we deemed that adding a new tag would not increase significantly the number of games covered. After one iteration of this process, we obtained the capital tags of rank 2: *RPG*, *2D*, *3D*, *Atmospheric*, *Simulation*, *Colorful* and *Puzzle*. After a second iteration, we obtained the capital tags of rank 3: *Pixel Graphics*, *Funny*, *Story Rich*, *Fantasy*, *Arcade*, *Relaxing*, *Shooter*, *Management*, *Horror*, *Sci-fi*, *Platformer*, *Co-op*, *Third Person*, *Open World*, *Rogue-like*, *Exploration* and *Sports*. We observe that out of the 4065 games with at least 2000 reviews, 98% have a capital tag of rank 1, 96% have a capital tag of rank 2, and 97% have a capital tag of rank 3. Moreover, 99.7% have at least one capital tag (of any rank).

Figure 2 shows the naming scores when the K-means algorithm is applied to a variation of $\mathcal{D}_2$ where we keep only on the capital tags of rank 1, 2 and 3, compared to the previous results with the whole of $\mathcal{D}_2$. Note that when we compute the naming score of a clustering, we do it on all Steam tags, not only on the capital tags.

There are 14 games out of the 4065 that do not have any capital tags, and thus were not put into any cluster. Since we are missing only about 0.3% of the considered games, this difference is not significant.

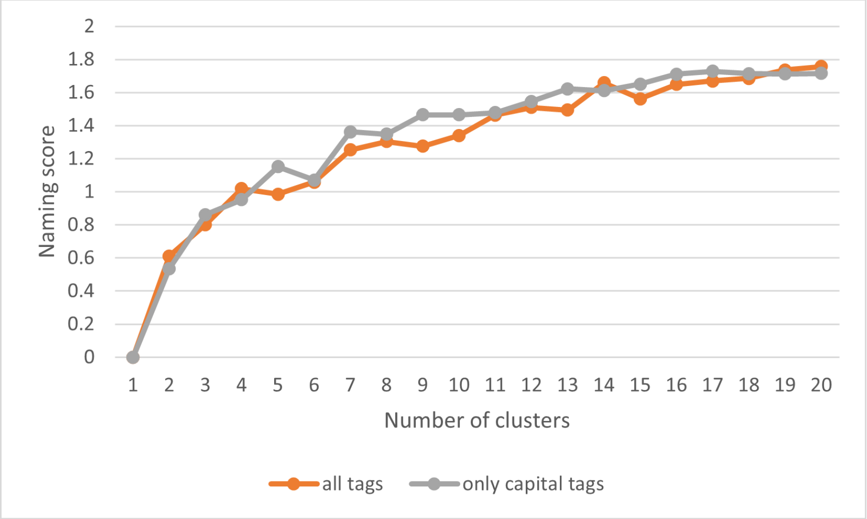

Fig. 2. Comparison of the naming scores of the clusterings depending on whether they are computed on all tags or only on the capital tags.

3.3 Using the ranks of the capital tags

We have seen in Figure 2 that, for clustering the games into groups with distinctive names, using only the capital tags does not deteriorate the results. However, in Working Hypothesis 1, we stated the belief that capital tags do not only preserve the quality of the naming, but even improve it. We had the following intuition:

Working Hypothesis 2 *To obtain the best naming scores, capital tags should be treated differently according to their ranks.*

Let us consider a game G. As said in Subsection 3.2, G is very likely to have at least one capital tag of each rank. Let us consider only the capital tags of rank i assigned to G. We tried several functions to apply to the priorities of the tags, we present here the one that gave the best naming scores. If G has no capital tag of rank i, we do nothing. Otherwise, we can associate G to a vector in $[0,1]^{d_i}$, where d_i is the number of capital tags of rank i. The coordinate along axis j is the priority of the capital tag j of rank i assigned to G. First, we normalise each vector by its ℓ_2 norm. Secondly, we multiply it by a scaling factor denoted as λ_i, where $\lambda_1 = 0.25$, $\lambda_2 = 0.7$ and $\lambda_3 = 1$. Those values are the ones that maximised the naming scores in our manual experimentations.

Figure 3 shows a significant improvement of this new solution compared to the previous ones in terms of naming score, thereby confirming Working Hypotheses 1 and 2. This algorithm, using the database with priority and only keeping the normalised and weighted capital tags, is the algorithm that maximises the most the naming score among all the methods we tried.

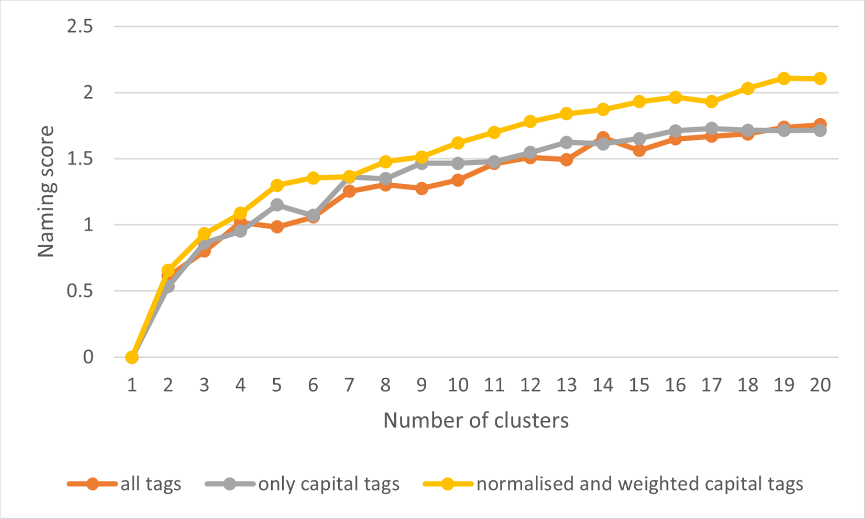

Fig. 3. Comparison of the naming scores of the clusterings depending on whether they are computed on all tags, only on the capital tags or only on the normalised and weighted capital tags.

4 Conclusion

We have introduced a new optimisation function for clustering games into groups with distinctive names: the naming score. We showed that using the priority notion on normalised and weighted capital tags improves the naming score of the clusterings when using the K-means algorithm.

Our clustering algorithm works only on games that have at least one capital tag. As 99.7% of the games with at least 2000 reviews have at least one capital tag, this is not too problematic. Still, it would be interesting to try to guess for the remaining games what their capital tags could have been. We believe this could be done using traditional machine learning methods.

References

1. S. Dolnicar, "A review of unquestioned standards in using cluster analysis for data-driven market segmentation," 2002.
2. J. A. Saunders, "Cluster analysis for market segmentation," *European Journal of marketing*, vol. 14, no. 7, pp. 422–435, 1980.
3. D. Arsenault, "Video game genre, evolution and innovation," *Eludamos: Journal for computer game culture*, vol. 3, no. 2, pp. 149–176, 2009.
4. S. Heintz and E. L.-C. Law, "The game genre map: A revised game classification," in *Proceedings of the 2015 annual Symposium on computer-human Interaction in play*, 2015, pp. 175–184.
5. X. Fang, S. S. Chan, and C. Nair, "A lexical approach to classifying computer games," 2009.
6. C. Bauckhage, A. Drachen, and R. Sifa, "Clustering game behavior data," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 7, no. 3, pp. 266–278, 2014.
7. D. Ashlock, D. Perez-Liebana, and A. Saunders, "General video game playing escapes the no free lunch theorem," in *2017 IEEE Conference on Computational Intelligence and Games (CIG)*. IEEE, 2017, pp. 17–24.
8. P. Bontrager, A. Khalifa, A. Mendes, and J. Togelius, "Matching games and algorithms for general video game playing," in *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 2016, pp. 122–128.
9. S. Zadtootaghaj, S. Schmidt, N. Barman, S. Möller, and M. G. Martini, "A classification of video games based on game characteristics linked to video coding complexity," in *2018 16th Annual workshop on network and systems support for games (NetGames)*. IEEE, 2018, pp. 1–6.
10. P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of computational and applied mathematics*, vol. 20, pp. 53–65, 1987.
11. J. Cohen, *Statistical power analysis for the behavioral sciences.* Academic press, 2013.
12. N. Grelier and S. Kaufmann, "A data-driven classification of video game vocabulary," in *Entertainment Computing - ICEC 2023 - 22nd IFIP TC 14 International Conference*, ser. Lecture Notes in Computer Science, vol. 14455. Springer, 2023, pp. 17–30. [Online]. Available: https://doi.org/10.1007/978-981-99-8248-6_2
13. K. Pearson, "On lines and planes of closest fit to systems of points in space," *The London, Edinburgh, and Dublin philosophical magazine and journal of science*, vol. 2, no. 11, pp. 559–572, 1901.
14. N. Grelier and S. Kaufmann, "Data-driven classifications of video game vocabulary," *arXiv preprint arXiv:2303.07179*, 2023.