Determining Successful Game Features from Steam Tags

Rasmus Dyhr Larsen, *Student, ITU,* Paw Høvsgaard Laursen, *Student, ITU,* and Rasmus Bons, *Student, ITU*

***Abstract*—In this report we use data from the SteamSpy API[1] which we have processed and analyzed. Due to the lack of individual player data, we focus the analysis on the tags and score values of the games. We account for the fundamentals of the used mining algorithms - Apriori, for pattern mining and Self-Organizing Maps for unsupervised clustering. We use Apriori to identify the typical tag combinations and analyze the ratings of these combinations. The mined tags combinations are then used for two things: Firstly, we use them to generate association rules using the Apriori algorithms. These rules, not only enables us to see which combination of tags infers the occurrence of other tags, but also let us analyze which change in rating the inference of the rule creates. Then we use these mined frequent tag combinations to prioritize the Self-Organizing Maps focus areas. With the Self-Organizing Maps we are able to visualize and analyze the clusters and meta-genres of the games.**

We finally estimate the findings of the project to be helpful in the selection of focus areas within game development.

***Index Terms*—Data mining, frequent pattern mining, unsupervised learning, video games, game development, Steam.**

I. Introduction

Game development has always been a tough start-up business. To make it, you have to break through with one of the first games you make. However, exactly what to aim for to make a game that sells is not that obvious. If you are aiming for a certain genre, what kind of mechanics usually work? E.g. should you include a loot system when you are making a shooter? What art style should it be, and should there be focus on graphics? There are a lot of questions, none of them easy to answer, resulting in many games getting it wrong.

In this project, we aim to collect and analyze data in order to examine which combinations of game features that work well together. By mining game data from Steam, which is one of the most popular game distribution platforms, we are going to investigate the patterns of what feature that increases the games reception, but also which affect the review scores in a negative way.

II. Data

This project uses data from the SteamSpy API, which, at the time the project started, contained information about 13.080 games on digital distribution platform, Steam, including amount of owners, playtime, price, rating and tags. The data is extracted as .json file, and parsed into 'Data'-objects, each representing one game with its relevant attributes.

[1]reference for api and attributes found at https://steamspy.com/api.php

As the goal of the project is to mine for patterns in released games, using data from the Steam platform is an ideal choice, because of its popularity, the ease of obtaining the data from SteamSpy, and the recent rapid growth in the number of games that are available on the platform.

However, some of the attributes from SteamSpy has been reduced to mean and median in the API due to the amount of data required to store individual player information. This removes the possibility of handling extreme outliers, such as owners that has bought, but never played the game, and gives us less insight into the amount of time players are playing the game.

In order to classify and examine different genres of games, we have chosen to focus the research on the attributes: 'tags' and 'rank'. Tags can be used to provide insight into how games group up compared to tags, and the rating will give an estimate of how well the different classifications are rated. A notable feature with this data set is that there are many different tags(335), which can describe both genre (RPG), gameplay (difficult), accessibility (early access) and interaction (moddable). This result in patterns that points towards not only in game elements, but also the relevance of metagame elements.

To obtain the desired attributes, the only two pre-processing methods necessary is data removal to get rid of the unwanted attributes, and noise removal to remove null values. Additionally for one of the methods, the tags have to be transformed from a list of tags, to a vector of booleans indicating whether a game has a tag with a given index.

III. Methods

Since we are analyzing which combinations of tags are both typical and successful, we chose to use a pattern mining approach, specifically the Apriori algorithm, to be able to identify frequent combinations. Then we will take these frequent combinations and use self-organizing maps to further mine and analyze them.

A. Apriori

Apriori is a frequent pattern mining algorithm, which uses prior knowledge of properties from previous frequent itemsets(k-itemsets) to generate additional frequent itemsets (k+1)-itemsets[1].The algorithm iterates until it cannot find any frequent k-itemset. Each iteration is consisting of two steps: *join* and *prune*.

In the *join*-step, new possible k-itemsets, called candidates, are created by joining all supported (k-1)-itemsets with each other. After the joining the candidates goes through the *prune*-step. The candidates are scanned in the data storage to determine their support count. Any candidate that does not meet the minimum support threshold are discarded.

The set of candidates used during the *prune*-step can contain huge amount of candidates, which is expensive to iterate through. To reduce search possibilities and improve the efficiency of the algorithm, the Apriori property is introduced. This property states that all nonempty subsets of a frequent itemset must also be frequent[1]. This is used to reduce the size of the set of candidates, by discarding candidates if any (k-1)-subset of the candidate is not frequent.

We will be using Apriori to find frequent tag combinations, as well as finding association rules. We will then analyze which combinations are successful by looking at the ratings of the most frequent tag combinations.

We will also generate Association rules, which will tell us how often the occurrence of a not-empty set of tags infers the presence of another not-empty set of tags. To analyze whether the implications of the rules actually makes any difference, we will use ratings delta value, which measures the difference in rating between where the premise infers the conclusion and where the premise does not. The ratings delta will be calculated by taking the average rating of all games that contains both the premise and the conclusion and subtract the average rating of the games that contain the premise, but not the conclusion.

B. Self-organizing Map

The type of artificial neural network known as a '*Self-organizing map*' is an unsupervised algorithm that uses competitive learning to create a low-dimensional discrete representation of the high-dimensional input data. The *map* is made up of a two dimensional array of neurons, which each are trained to fit the data, thereby organizing it, hence the name.

First, each neuron of the map is initialized with low random values. Then neuron best matching the input is found for each datapoint, using euclidean distance. The weights of the best matching neuron, and those in its nearest neighborhood, are adjusted towards the tested input[3]. The adapted neurons are updated according to Kohonen's update rule, which is defined as:

$$W_j(t+1) = W_j(t) + \eta(t).h_{ij(x)}(t)(X - W_j(t)), \quad (1)$$

where $0 < \eta(t).h_{ij(x)}(t) \leq 1$, t is step index, $\eta(t)$ is learning rate function, $h_{ij(x)}(t)$ is a Gaussian neighborhood function using distance between neuron *j* and *i*, *x* is the input and *X* is weight of the best match.

IV. Results

A. Descriptive Statistics

To get more insight into the data, simple descriptive statistics were performed on the attributes, that is subjectable to calculation of mean, median and standard deviation. The null value(not reported in) from price and rating was removed, prior to the calculations. The results can be seen on *Fig. 1.*

Attribute	mean	median	std
Owners	$1.91*10^5$	17037	$1.2*10^6$
Price(cent)	912.04	599	1630
Rating	45.70	43	30.21

Fig. 1. Descriptive statistics for relevant attributes

In addition to these, a histogram of game tags were created and the most frequent are referenced in Appendix C, to give an overview of the most popular tags across all games.

The descriptive statistics does not provide much new insight to the games. However, the high standard deviation does show that there is a large spread in the data, indicating the games differs a lot in the different attributes. The low mean and median in price shows that the majority of the games are cheap, while the rating indicates that most rating are gather at the lower middle.

We have also calculated the average rating for games with each specific tag. For reference on the average ratings of for all tags in games, see the *TagRatings.xlsx* file.

B. Apriori

1) Experimentally choosing parameters: We first use the Apriori to pattern mine the tag combinations of the games to find the most frequent combinations. We experimented with using different support thresholds from 3% up to 10% incrementally. These results are included in the *ThresholdExperimentalResults.xlsx* file. When using a threshold of 10% we were only able to find a single supported 3-length tag combination, namely Action, Adventure and Indie. Since we wanted tag combinations of at least length 3, for the combinations to be more meaningful, we found a threshold of around 4% to be the most meaningful. Here we found 49 supported tag combinations of length 3 or more.

2) Pattern Mining: We find the high average ratings of the combinations containing the tag 'Great Soundtrack', to be misleading, e.g. the supported tag combination of 'Adventure', 'Great Soundtrack', and 'Singleplayer' at a rank of 64,92%, since the tag 'Great Soundtrack' not only speaks to the occurrence of a feature, but also the quality of said feature. We would likewise believe that if there existed such a tag as 'Shitty Soundtrack' (there does not), its occurrence would generally imply a lower rating. Similarly the tag 'Atmospheric' is considered to a positive cognition, and will therefore also be generally disregarded. This is since we do not just want the findings of the study to be: to get good rating make high quality games. This is (hopefully) logically implied.

Of frequent tag combinations with interesting ratings should be mentioned:

Action Multiplayer Open World - 39,62 average - 35 median

This is an extremely low rating, with an even lower median, meaning that most games are even lower. In the tag rating reference file, we see even though each of the individual tags averages are around 45%, the combination of the three gets the rating even lower. We believe that making a multiplayer action game with an open world seems extremely ambitious, which could be a reason why so many games fail here. This serves a another caution for game development companies to think of scoping.

2D Action Indie Singleplayer - 64,39 average - 69 median
Action Indie Simulation - 41,95 average - 39 median

Here we find a extremely successful frequent tag combination of '2D', 'Indie', 'Action' and 'Singleplayer'. With an average of 64,39 and a median of 69 this is one of the highest ranked combinations. That '2D' and 'Singleplayer' is combined with the 'Indie' tag, could serve as another reminder of the importance of scoping, since 'Indie' typically would mean low-budget and especially '2D', could be considered be a more achievable feature, than the alternatives. Serving as an example of how a well-functioning combination can still be rated highly by gamers, despite (or even because of) modest features.

It is also interesting to see how that the combination of 'Indie', 'Action' and 'Simulation', ranks as one of the lowest combinations, despite sharing many tags, with the previous combination.

3) Association Rules Mining: For mining association rules, we lowered the threshold to 2% and considered frequent tag combinations of all lengths, this is because we wanted to be able to use a confidence of a significant level (70%) and still be able to produce some rules, which we experimentally found to not be the case when using a higher threshold.

As mentioned earlier, we used wanted to consider the actual impact on rating of the rule, by measuring a ratings delta value. In the attached document *APRIORI ASSOCIATION RULES-Thres2-Confidence70.txt*, all the mined association rules can be found.

C. Self-organizing Map

The self-organizing map has a long training time, so in order to produce results within a reasonable time-frame, the map was trained with the data points from the frequent tag combinations found in the results of the Apriori algorithm. Doing this significantly reduces the number of items from 13000 to a few thousands, depending on the specific thresholds. Additionally only a subset of the resulting items were picked at random to train the map with.

Specifically the map was trained with a support threshold of 3% and about 25% of the resulting items were used. This resulted in a training time of a few hours, depending on the hardware.

1) Coloring the Map: The algorithm is able to produce several kinds of visualizations. For this project, it generated a color coded landscape with the data points painted on in black, and a black and white landscape with the data points painted on in various colors.

The visualization with the colored tag landscape gives an overview of how the tags group together. Each tag has its own color, and for each neuron the color is blended based on the weights of the neuron. The image is most useful in confirming that groups of tags indeed exist. It is expected that there will be areas of certain colors that blend into other areas smoothly. The large downside of this visualization is that it is impossible to extract any concrete data from the image itself. The colors of the image cannot be traced back to their original tags because they are a blend of multiple tags based on weights that have some error.

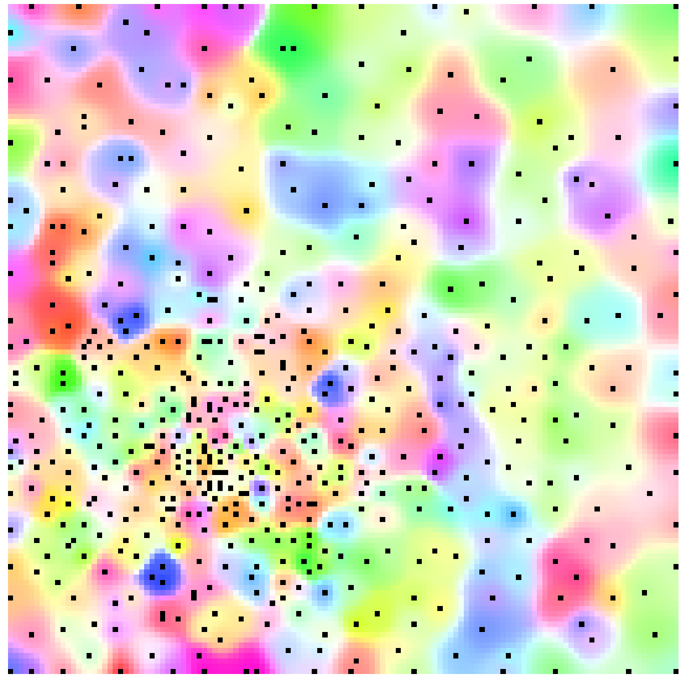

Fig. 2. The product of the self-organizing map with colors based on tags.

To solve this we use another visualization, where each data point has a color. The image does not show the landscape of tags in any way, but only how the data points are positioned in relation to each other. Because each data point has a color, it can be looked up, which can give an overview of what games are grouped together.

2) Resulting images: Figure 2 and figure 3 are the two images that were produced by the self-organizing map algorithm.

We find it notable of figure 2 that there are a lot of areas with a similar color, arising from the games in that area sharing some tags. The colors themselves does not mean anything, as they are just random colors that are blended together. Some of the areas also blend together in a fairly smooth transition, for example the yellow area near the top left of the image that transitions into the purple area to the south, and the orange area to the northwest.

The opposite case also happens a lot, such as near the top right corner, where there is a green area next to a purple area, with no transition at all in between. In extreme cases there are also tag combinations that are completely isolated from their environment, such as in the bottom left corner with the blue area, that contains three data points. These completely isolated sets of tags are hard to identify due to the nature of the random colors, so they are expected to be more frequent than they immediately appear.

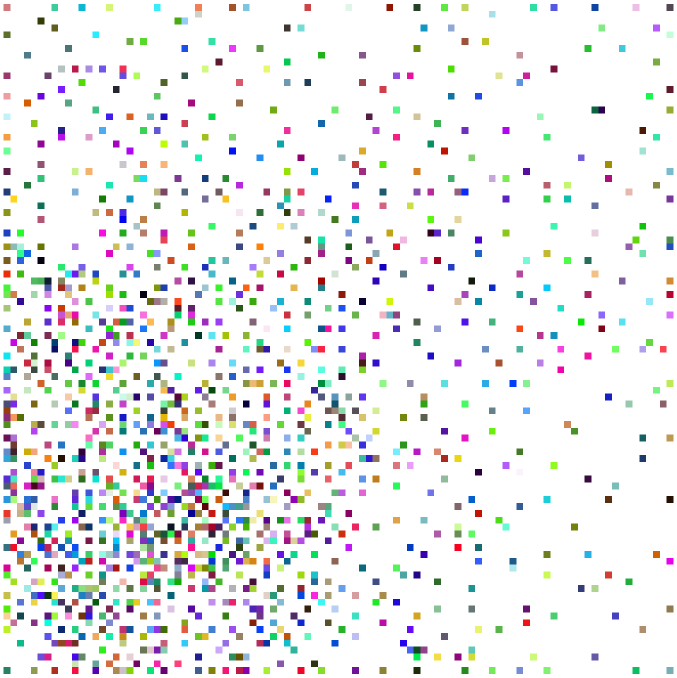

Fig. 3. The product of the self-organizing map with colored data points.

As mentioned, it is difficult for us to extract exact data from this image, but it does confirm that tags does group together in some way. This can be seen both in the huge cluster of data points, that probably share several tags, and in the distinct areas that are produced.

Figure 3 tries to provide some specific data points at the loss of the colored landscape of tags. Since we have not made a tool to analyze the data, extracting information form the image is a process of sampling the color from a specific point, looking up what game it corresponds to, and looking up what tags the game has. We have found two broad observations when sampling various points around the map.

First, the big cluster of data points is made up of mostly indie games in the action, adventure or strategy genres. We suspect that this is due to the overwhelming amount of indie games that have been released on steam recently. This pattern is also visible in the descriptive statistics analysis just by looking at the histogram of tag frequency found in Appendix B. In conjunction with this, non-indie games and indie games from other genres all lie in the provinces around the central cluster. Some specific examples include Half-Life, a non-indie game, and Super Arcade Football, an indie sports game.

Finally we have observed that the Self-Organizing Maps may not always be continuous. Games with identical tags can be found in different areas. This is to be expected based on the fact that the algorithm has no mechanism to prevent this. In figure 2 the three blue areas could be an example of this phenomenon, although there is no way to confirm it on that specific image. This problem does not interfere with the other observations though, as tags still group together in the expected way. It simply means that the map is not perfect, and does not provide the very most accurate representation of the landscape of game tags.

V. Discussion

The findings of the association rules mined with the Apriori algorithm along with the change in rating the inference of the rule creates are especially interesting.

2D, Action→Indie ; conf.: 88,40 % dRating: 8,78

This points back to the successful combination of 2D, Action, Indie, and Singleplayer from the frequent combination mining part of the algorithm. Here we see that not only is does the 2D and Action combination often infer the Indie occurence, but also that this combination is actually higher rank as an indie game.

Open World→Adventure ; conf.: 76,45 % dRating: 9,45

It seems logical that a Open World game would be a good candidate for Adventure elements. Here we also see that focusing on adventure elements for an Open World game increases the rating of the game.

VR→VR Only ; conf.: 74,44 % dRating: 10,95

This one is interesting because it speaks to the effect of focus in a game. When VR games are only for VR, it not only increases the focus of the game design, but also increases the chance that players actually experience the game in its optimal play setting, which would also increase the player's impression of the game.

It is important note that even if the game does not have the 'Singleplayer' tag, it could still be a singleplayer game, it has just not been tagged as such sufficiently by the users. We can however analyze that it is something the users find or do not find as an important element of the game. It is therefore applicable when mining with the goal of weighing the necessity of different features.

As found in the attached TagRatings.xml file, it is shown that the 'Singleplayer' tag has a very average rating of 53.56, meaning that it is not just the occurrence of a singleplayer mode that makes a a game increase or decrease drastically in rating, but we have certainly found it to be very impactful in specific tag combinations. This also helps show the validity of the findings, as a simple rating average of a tag, could be considered misleading, if not considering the other features of the game in development.

The self-organizing map was a bit lacking in producing findings. It acts more as a tool to confirm our assumptions about the relations between tags, than a useful tool for discovering relations. Some of this is due to our implementation that does not have any tools for analysing the data, but in the best case the map could have found the exact clusters that tags belong in, which in turn could be used to classify exact meta-genres.

Finding clusters in the self-organizing map assumes that there are clusters to be found. In the images produced there appear to be one large group of games, that we identified as mostly indie games, but apart from that the landscape of games is pretty evenly spread. That is not to say that clusters of tags does not exist, just that using the self-organizing map to find them is insufficient. We expect that the areas of the map that are clearly made up of tags grouped together are the meta-genres of games, and we would almost certainly be able to identity the clusters exactly either using a different algorithm, that is more specialized to find clusters.

The main reason we do not expect further analysis of the self-organizing map to yield exact results, is because of the observation we made, that it is not always coherent, and sets of tags that are similar may be great distances apart. It may be a result from the even spread of tags among games, the initial random state of the map, a combination of the two or something else entirely.

In general the data in these findings should be quite viable in helping to determine game development companies in which features to prioritize, when making a particular game.

VI. Conclusion

In part because of the limitations of our data, it was not possible to say something directly about a video game's success, other than the user rating of the game, but the measurements of success could of course also be defined in any number of ways.

One of the general tendencies we found in the ratings of the typical tag combinations, were that ambitious feature combinations, like Action, Multiplayer and Open World were unsuccessful, while more modest combinations, like 2D, Action and Indie, were successful, especially for indie companies.

The association rules mining gave some very interesting results. These make it possible for game developers to determine which tag, are higher rated by players for certain types of games.

The self-organizing map showed clear clusters of games, which could be further mined to estimate what meta-genres games belong to. The clearest of the clusters is the large group of indie games, which presumably is that large amount of games that have been released on steam in recent times. Unfortunately the tags are not consistent enough for the resulting map to be completely coherent, but the clusters of tags where found regardless, even if some of them are scattered throughout the landscape.

It should be possible using this data to make a game design planning tool, where developers could look up a combination of tag they know they are already planning to have, and then see which other features would detract or add to the experience of the users. Additionally the tool could provide games that belong to the same meta-genre, to give specific examples of games for inspiration on how to implement the suggested features. This can be done a lot earlier than any actual user testing or QA, which could save a lot of time and money by avoiding unnecessary elements and feature creep.

Play session data like session time and time of day could be interesting for future work, as well as user data, like age and gender. This would enable mining, that could help tell which target group or target play session length, game developers should aim for, when creating a new game.

Appendix A
Apriori frequent tagsets

Appendix one text goes here.

Appendix B
Histogram of most frequent tags

Histogram can be seen on its own page after the biblography.

Acknowledgment

The authors would like to thank Sergey Galyonkin for his amazing work on SteamSpy.

References

[1] J. Jiawei, J. Pei and Micheline Kamber, *Data mining: concepts and techniques*, 2nd ed. Amsterdam, Dutchland: Elsevier/Morgan Kaufmann, 2012.

[2] Wikipedia, *Self-organizing map*, 2017.

[3] T. Kohonen, *Self-organized formation of topologically correct feature maps*. Biological cybernetics, 43(1), 59-69, 1982

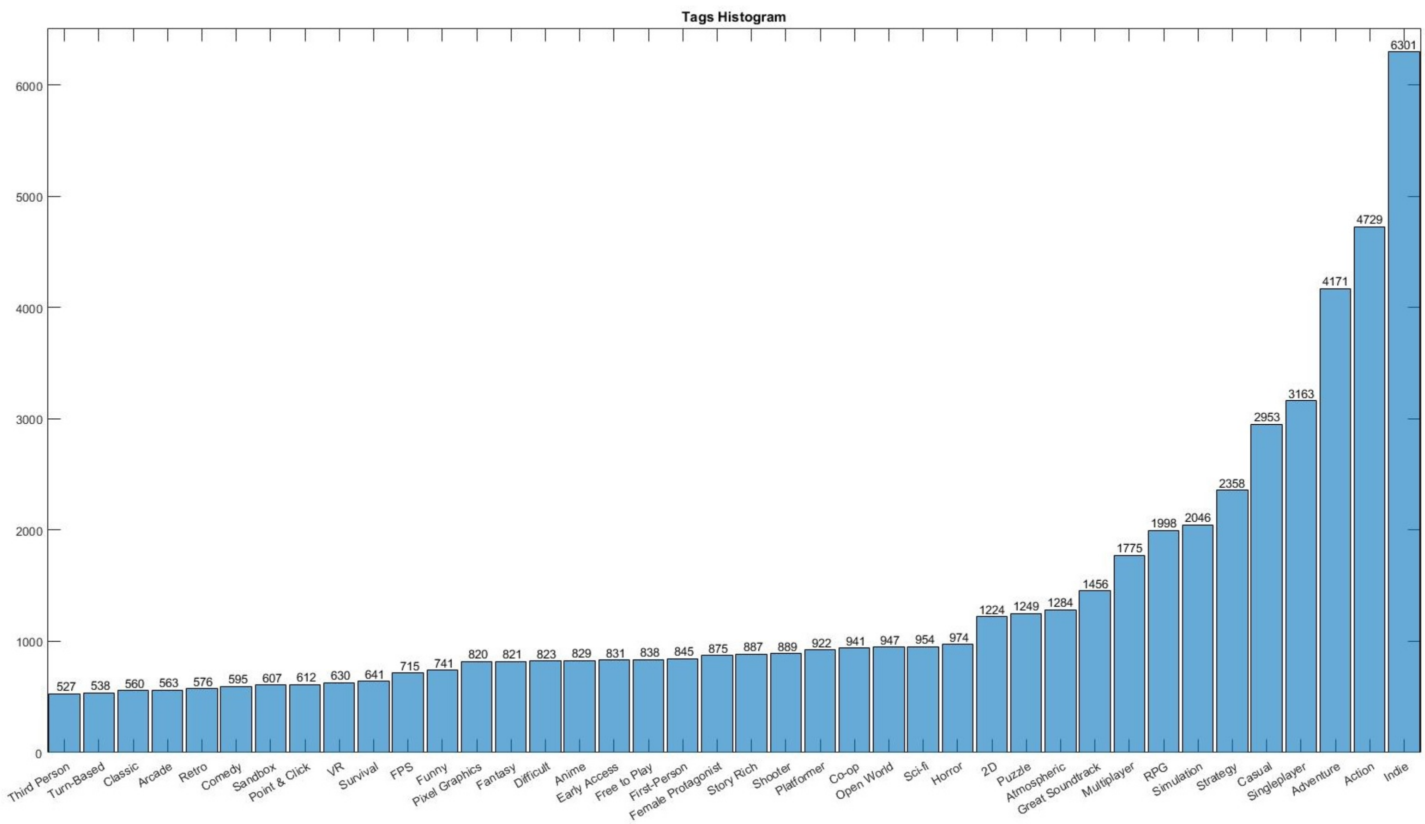

Fig. 4. Histogram of the tags that is included in at least 4% of the games