SteamRec: Recommending the Recommenders in Valve's Steam Marketplace

Jacob Fischer (jtf3m8@mst.edu)
Barry Preston (rbp9bc@mst.edu)
Samuel Richter (snr359@mst.edu)

Abstract - Valve's Steam marketplace is the largest digital store for PC games. However, with it being the largest digital PC video game marketplace, there are too many titles for most consumers to easily dig through. Valve has attempted to fix this with Curators, which maintain their own list of recommended games so customers can go to them for recommendations. Because the most popular Curators have audiences in the tens to hundreds of thousands of users, getting recommended by a top Curator is incredibly important and can result in a huge boost to sales numbers . In this paper we propose a system, called SteamRec, that attempts to model their recommendations in order to predict what new games Curators would recommend. We use existing data science techniques and bring in new novel ideas, such as temporal biases and data slicing, to better predict the outcome of games in the Steam marketplace.

I. INTRODUCTION

A. Background

As digital marketplaces increase in popularity and inventory, there is an increasing problem for consumers to sift through the noise. Indeed, in Valve's own huge popular digital video game marketplace [9], Steam, that problem has come to critical mass as new games are added each year at an almost exponential rate [10]. To curtail the flood of games, Valve introduced a 'Curator" feature in 2014, where users could be their own recommendation engines to sift through all the products. With Valve shifting the responsibility of product discovery away from themselves, Curators have become an incredibly powerful tool for new games to get noticed by customers.

Because of this new paradigm, smaller games that do not have large marketing budgets, like that of the big budget titles, are increasingly dependent on the top Curators recommending their titles so the larger customer base can "discover" these titles. However, as of this writing there are over thirty four thousand unique Steam Curators, and it is a daunting proposition for game developers to parse through to find the Curators who would be most likely to recommend their game to the millions of Steam customers.

B. SteamRec

Luckily, this is a data science problem we can tackle. In this report we detail our attempt to build a system, called SteamRec, that can recommend Curators based on input games [1]. SteamRec attempts, through modified traditional data science algorithms, to reverse engineer the Steam Curators which are essentially recommendation systems unto themselves, and then model a new recommendation system around that prospect. We also take this a step further and use some of the unique aspects of the Steam data. For example we use *when* a game is recommended, to add a temporal bias. Video games are released when they are expected to sell better, such as near holidays; thus we hypothesize that bringing in time as a major bias would be important to more accurately model the data and produce more useful results.

C. Problem Statement

Our intention with SteamRec is two-fold. Firstly we intend to contribute to the data sciences by presenting and explaining a novel modification to existing cluster and recommender techniques that can be used on data sets such as Steam's marketplace. Secondly, we envision our system as being useful to the video games industry at large. If game developers can feed information about their hypothetical game(s) into SteamRec, and SteamRec can then output a manageable list of a few Curators who are highly likely to enjoy the game(s), then they have a much smaller, more manageable list of Curators to target instead of all of them.

II. METHODOLOGY

A. Dataset Gathering

Valve does expose *some* of its Steam data via a developer API. However, the API as of the time of this writing does not expose many aspects of the Steam marketplace such as any Steam Curator data. Because of this we had to write our own website scraping code to gather that data.

We used a Python script to scrape the list of Curators from Steam's AJAX API. By parsing the raw JSON response via regex, we were able to get the list of Curators from each page of the listing. The steam community has over 34,000 Curators; for this project we narrowed this list down to the top 1000. We picked this cutoff to give us a good amount of data, below the top 1000, Curators tend to only have around 40 followers. Given that the Steam community has active membership in the millions, the data from Curators small than these would have minimal usefulness to be recommended.

To scrape the individual Curators' recommendations, we use a Node.js script to query again Steam's web API, which returns HTML to build the Curator page. We parse the HTML to figure out: the appid; whether the recommendation was positive, negative, or "informative" (neutral); and the time of a Curator's recommendation. We store all this data in JSON files so SteamRec can easily parse it without having to re-scrape all the Curators, a process which can take over an hour to run.

For the game data, we used Steam Spy, which exposes a simple API that can, in one query, return a JSON-formatted list of all the games in steam, along with various metadata. The metadata we were most concerned with were each title's Steam Tags, which is a list of genres and attributes the community collectively agrees fit the title. In addition, the Steam Spy API exposes how many users tagged the game with each tag, so we can use their count to further bias recommendations.

It is also important to note that the official Steam API and SteamSpy both exclude some games from their output. This is entirely up to the game developer if they wish to hide it. The few titles that do this are ignored in recommendations by Curators, as we do not use the official APIs there.

B. Recommender Dead Ends

To build SteamRec, we did investigate a few dead ends before we ended up in our final algorithm. Our first algorithms made use of the Surprise recommender learning system [2]. At first we just fed into Surprise the matrix of Curator IDs to App (game) IDs, with a 1 representing a recommendation and a 0 representing no recommendation. However, this methodology would just help to predict recommendations based off the App ID, which is largely arbitrary for any given game.

Next, we decided to leverage each game's Steam Tags as a richer way to represent a game. Using Steam Tags enabled us build a matrix of Curator IDs to each Steam Tag, with the sum of each game they recommended tagged with that Steam Tag. We furthermore penalized negative recommendations by counting that game's Steam Tags as a negative score. Using this system, if a Steam Curator dislikes a genre of games, tags for that genre will have a negative score, and vice versa. However, again using Surprise's built in prediction functions just predicts what a Steam Curator would score a given tag, not a collection of tags that represent a new game.

III. ALGORITHM

A. SteamRec Algorithm

Our final SteamRec algorithm is a set of custom modifications to the K Nearest Neighbors algorithm. Our modification takes into account the temporal aspects of the data. SteamRec is KNN-based, and uses the difference between two games' normalized tag counts as distance. To predict a game, the nearest neighbors to that game each vote for every Curator that recommends them, with a weight based on the timestamp of the recommendation. This means that more recent reviews are more heavily weighted than older reviews, and thus Curators that are actively recommending games recently should be preferred. Games that are not presently recommended by any Curators are excluded from the graph, as they provide no useful data.

To validate the effectiveness of this algorithm, we built a graph of all the games on Steam recommended by at least one curated, excluding a set of 100 randomly chosen, recently-recommended games (i.e. recommended after March 1st, 2017). We then used the graph and recommender algorithm to predict *n* Curators who were likely to recommend each of the 100 games. We tested different values for *k* and *n*, and measured the percentage of games where the algorithm correctly predicted at least one Curator who actually recommends the game. The results of this validation test are shown in the results section.

B. Temporal Data Slicing

Another unique aspect of the video game market is that games are not released randomly; instead, games are released when their developers think they will sell the best. This leads to many releases occurring towards the end of the year as holiday shopping intensifies in all markets. Because of this, recommendations may vary over periods of time with game releases. To help evaluate this, we have added another unique enhancement to our temporal biasing called “Temporal Data Slicing”. Using this technique, we choose a start and end time, or a “slice” of time, and build our recommendation matrix around that time slice. New games are then evaluated against just that time slice. We then run the prediction function for the last year, to illustrate who would recommend the game at what point in time. We slice the time either by months, so 12 slices, or by fiscal quarters (4 in a year) to simulate the most popular release windows. Using this algorithm, a prospective company could better tune their release dates based on the temporal preference of Steam Curators.

IV. RESULTS

A. Validation and tuning of n and k

The results of our initial validation test, detailed in the Methodology section, are shown in Table 1 and visualized in Figure 1.

Table 1
Accuracy of algorithm validation for k, n values

K, *Number of Curators fetched*	*1*	*2*	*3*	*5*	*10*
1	21	24	32	43	50
2	26	32	37	42	54
3	21	31	37	41	58
5	24	34	40	54	67
10	30	42	47	58	68
20	33	46	55	61	68
50	27	43	54	62	71

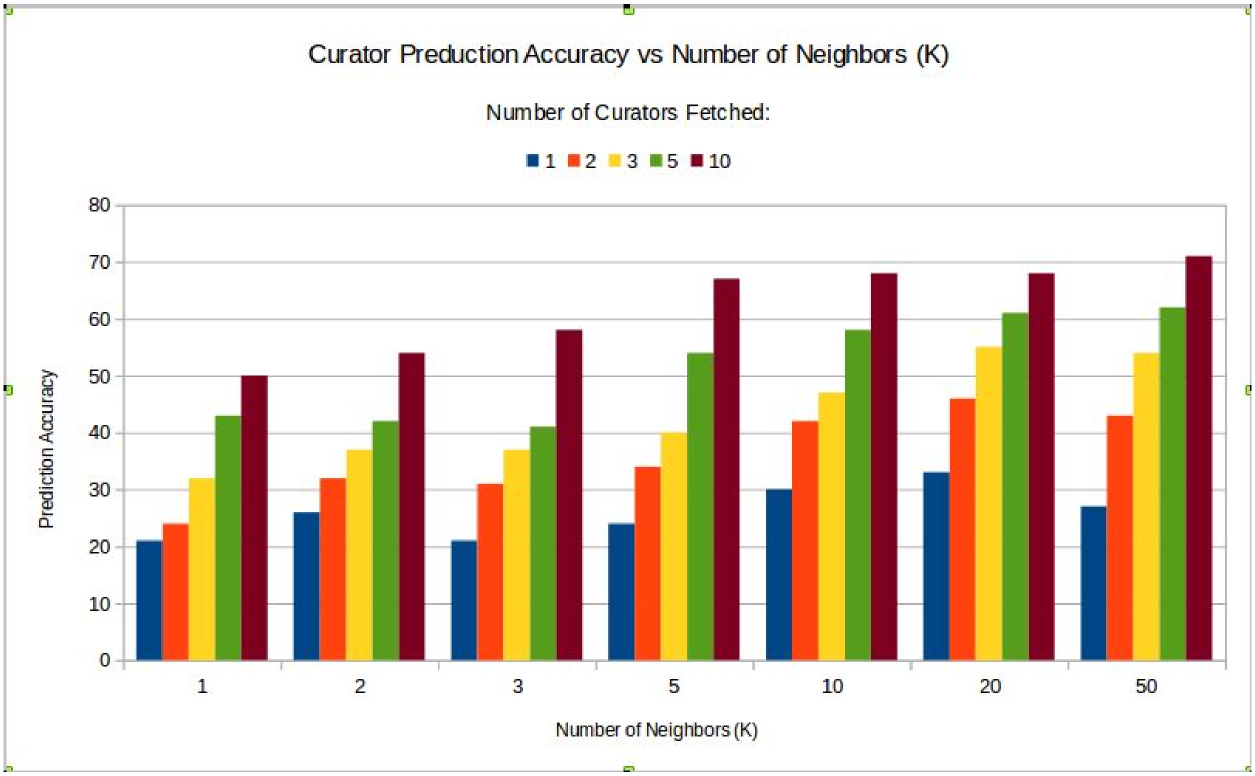

Figure 1: Accuracy of algorithm validation for k, n values

To validate our algorithm we simply removed 100 existing, recommended, games from the dataset. We then re-built the graphs without them and tried to predict what Curator(s) would recommend them. We also used this method to find optimal *k* and *n* values, where *k* is the number of nearest-neighbors used in our modified k-NN algorithm, and *n* is the number of Curators to recommend.

As you can see in Figure 1, increasing the number of Curators recommended increases the accuracy of our model, but it appears to start leveling off past 5 Curators. For this reason, we chose 10 as our default number of Curators to return. In addition, 10 feels like a good number, as it's still a manageable number for humans to manually look into those Curators for game sales.

For values of *k*, the results vary more wildly. As *k* is increased up to 5 marked improvements are observed, but past that, there are diminishing returns. 50 is only slightly better than 5. However, we still chose to stay with 50 as it produces the most accurate results for fetching 10 Curators.

It is also important to note that with an n of 10, the algorithm correctly selected at least one recommending Curator *at least 50%* of the time, with a correct result meaning that a recommended Curator in reality did recommend that game before we removed it from the validation dataset.

B. Predicting hypothetical games

For the following sections, we used the game as a series of Steam Tags with normalized counts as represented in Table 2. The values for *k* and *n* were the above selected values (50 and 10 respectively) for all runs.

Table 2
Hypothetical RPG game used for evaluation

Steam Tag	**Normalized Count**
Open World	0.20
RPG	0.20
Adventure	0.20
Fantasy	0.15
Singleplayer	0.15
Atmospheric	0.10
Character Customization	0.10

If we feed just this game into SteamRec, with the *k* and *n* values in Section IV-A, we get following recommended Curators (in order):

1. Fluent's RPG World
2. best games ever made
3. TRIPLE K ANIME MAFIA
4. Verified Official
5. AccidentlyReviewed
6. Completing the Backlog
7. Cubbes
8. LGBT+ Games
9. LGBTQ+ Inclusive Gaming
10. Cancercaurus

As we will see in future results many of these curators, especially “Fluent's RPG World” come up frequently.

C. Temporal Data Results

For temporal data slicing, we did two separate runs: one with 12 months to simulate 1 year, and one with four 4 month quarters, to simulate fiscal quarters a company would be concerned with.

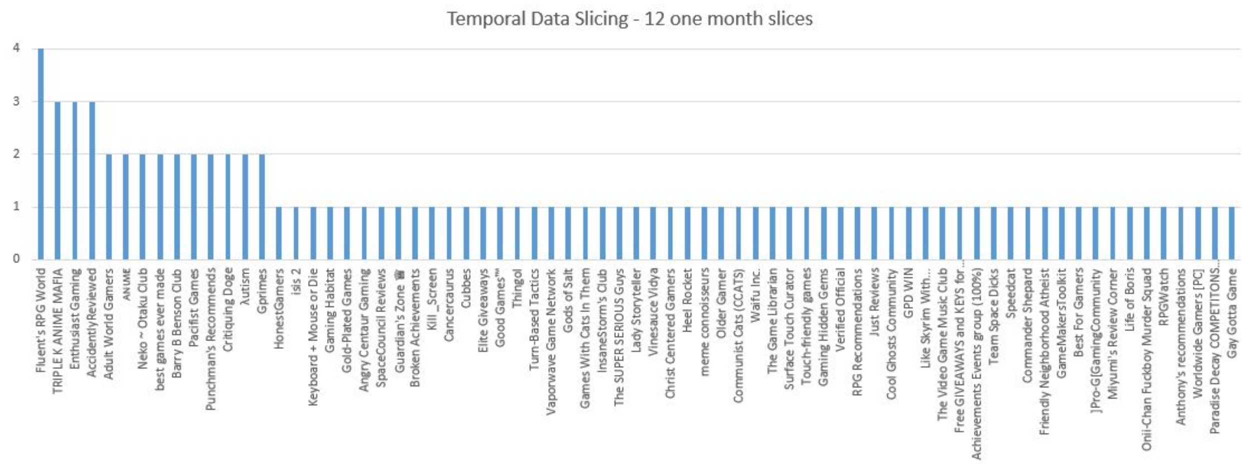

Figure 2: Twelve slices of one month intervals for Temporal Data Slicing

As you can see in Figure 2 above, the vast majority of Curators only came up once; they can almost be viewed as noise. The Curators that came up more than once would be the more interesting Curators to target, as their game tasks seem to overlap with our hypothetical game over the course of multiple quarters quite often. In addition, the fact that "Fluent's RPG World" is the only Steam Curator to be recommended in four separate temporal slices indicates that Curator would probably strongly enjoy our game.

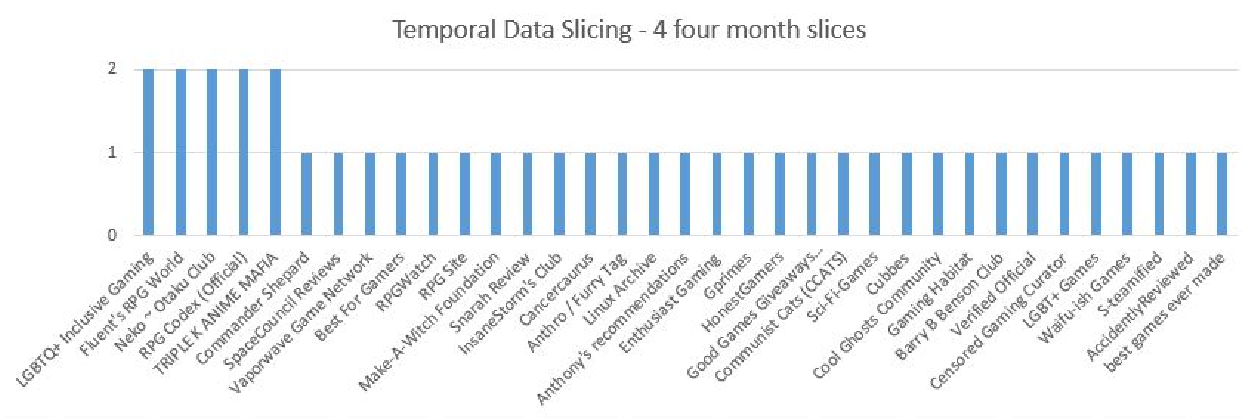

Figure 3: Four slices of four month intervals for Temporal Data Slicing

Figure 3, on the other hand, shows the results of simulating fiscal quarters, where we have fewer time spans, but their length is increased to 4 months (a quarter of a year). Because there are less temporal slices here, there will be fewer Steam Curators recommended, but with more time to model against the recommendations should be richer.

We can see with the quarterly slicing that the results are not as clear. Once again "Fluent's RPG World" comes up in the top, but this time its lead is shared among 5 Steam Curators. However, with only 4 total slices, this means that "Fluent's RPG World" and the other double recommended Steam Curators came up 50% of the time.

Overall, when considering Temporal Data Slicing in SteamRec, it may be richer to control your time span(s) as you see fit. If a company is trying to decide between a few months, then many different short one-month spans may lead to clearer top Steam Curators to target for advertising.

V. RELATED WORKS

Valve themselves have already applied several methods of game recommendation to their own Steam storefront, beginning with the Discovery Update in 2014 that, among other things, introduced Steam Curators in the first place [3]. In 2016, additional functionality was added to the storefront with the Discovery 2.0 update [4]. These two updates, together, enable the store page to present curated games to users based on their own preferences, trends among other users, and game content. Like our system, the Discovery system uses a game's community-driven tags to select and recommend similar games. However, Valve's efforts to recommend games are, so far, directed squarely at Steam's users, and although the largest Steam Curators have a significant impact in the Steam marketplace, there exist no special tools within Steam specifically for finding potential games for Curators to recommend to their audience. At the moment, Valve seems to be relying on their user-directed tool to serve that purpose.

Other video game storefronts have also been utilizing recommendation systems for some time now. In 2010, Sony's Playstation 3 received a system for recommending digital games for its users which uses data from other purchasers on the Playstation network [5]. In 2012, Koenigstein et al. developed a recommender system for the Xbox Live online service, which provides a marketplace for Xbox users to buy digital games [6]. However, to date, neither service has something analogous to Steam's Curators, which are human-driven recommenders which are themselves lucrative targets for automated recommendation; thus, neither service has any higher stakes in its recommender system than persuading each single user to purchase a video game. Steam, on the other hand, can significantly boost the sales of a video game by getting an influential Curator to recommend it positively.

In Taxi, Please! [7], Chucre and the other authors demonstrate a system to recommend taxi locations based on spatial data with a time element. While our dataset has no physical spatial attributes, we do use similar custom kNN modeling and graph building techniques. Both of our approaches are especially similar in the temporal aspect, where they model neighbors based on time sensitive data. While Taxi, Please! is more concerned with immediate time responsiveness for Taxis, they still deal with time slices; whereas SteamRec cares more about longer term time slices to use for biases.

VI. CONCLUSIONS

Steam is an ever growing marketplace that it is becoming increasingly hard to get noticed in without help. In this paper, we propose a new system, SteamRec, that leverages Steam's Curators feature to build a model that can predict which Steam Curators would likely enjoy and recommend a new game based on historical and temporal data. We explore a variety of metrics to tune the k-Nearest-Neighbors clustering based process to find the best values that result in the most accurate predictions with manageable results returned. We add a variety of temporal biases into SteamRec to influence the Curator recommendations based upon when the Curators recommend certain types of games via their Steam Tags. We finally explore a feature we call "Temporal Data Slicing" where we do multiple prediction runs on different time spans to find which Curators come up the most during desired time spans. This simulates release windows for games and gives game publishers more fine-grained recommendations that normal kNN could not produce.

A. Future Work

Steam is an ever evolving marketplace, and as it changes, the methods we laid out in the paper could be expanded upon and updated. Valve has recently stated that they will expand upon the Curator idea with Steam Explorers who will be encouraged to review lesser known titles and recommend them accordingly [8]. We do not know how this will impact our SteamRec system, but future work could explore that aspect biasing towards smaller indie titles.

Although SteamRec is built for new games entering the Steam ecosystem, there still remains a very large number of games already for sale on Steam but not presently being recommended by any of the top Steam Curators. SteamRec could be leveraged to recommend these games to Curators to increase their exposure in the same way as with a newly released Steam game.

Furthermore, the game-tag distance metric used in SteamRec's KNN algorithm could also be applied directly to the tag system, to recommend fitting tags for users to tag games with. This could be done by analyzing games that are already a small distance apart, games that similar users play, or new games that do not yet have a significant number of tags.

In addition, our SteamRec system could be applied to non game specific data sets. Similar media such as music, movies, and television could be tagged and recommended just as Steam games are currently. This is already a reality with music playlists which are essentially a list of recommended songs. A system based off SteamRec could do temporal data management like we do in this paper.

REFERENCES

[1] 'CS6001 Group Project for a Steam Recommendation Engine', 2017. [Online]. Available: https://github.com/JacobFischer/CS6001-SteamRec. [Accessed: 7-May-2017].

[2] 'Surprise: A Python Scikit for Recommender Systems', 2015. [Online]. Available: https://github.com/NicolasHug/Surprise. [Accessed: 14-Apr-2017].

[3] 'A Smarter Storefront, Just for You', 2014. [Online]. Available: http://store.steampowered.com/about/newstore. [Accessed: 7- May- 2017].

[4] Your Steam Store, Updated and Evolving', 2016. [Online]. Available: http://store.steampowered.com/about/newstore2016. [Accessed: 7- May- 2017].

[5] Next PS3 Firmware Update Adds PlayStation Store Recommendations', 2010. [Online]. Available: https://blog.us.playstation.com/2010/07/26/next-ps3-firmware-update-adds-playstation-store-recommendations/. [Accessed: 7- May- 2017].

[6] N. Koenigstein, N. Nice, U. Paquet, and N. Schleyen, "The Xbox recommender system," Proceedings of the sixth ACM conference on Recommender systems - RecSys '12, 2012.

[7] M. R. R. B. Chucre, S. M. Nascimento, J. A. F. D. Macêdo, J. M. D. S. M. Filho and M. A. Casanova, "Taxi, Please! A Nearest Neighbor Query in Time-Dependent Road Networks," 2016 17th IEEE International Conference on Mobile Data Management (MDM), Porto, 2016, pp. 180-185.

[8] John Bain, *I will Now Talk About My Visit to Valve and the Future of Steam For About an Hour*. 2017. [Online]. Available: https://www.youtube.com/watch?v=DeEyCYv_QDI. [Accessed: 7- May- 2017]

[9] 'Stardock Reveals Impulse, Steam Market Share Estimates', 2009. [Online]. Available: http://www.gamasutra.com/view/news/117102/Stardock_Reveals_Impulse_Steam_Market_Share_Estimates.php. [Accessed: 29-Apr-2017].

[10] 'Steam Spy on Twitter: "38% of all Steam games were released in 2016', 2016. [Online]. Available: https://twitter.com/Steam_Spy/status/804072335997358084. [Accessed: 29-Apr-2017].