A Field Study of Related Video Recommendations: Newest, Most Similar, or Most Relevant?

ABSTRACT

Many video sites recommend videos related to the one a user is watching. These recommendations have been shown to influence what users end up exploring and are an important part of a recommender system. Plenty of methods have been proposed to recommend related videos, but there has been relatively little work that compares competing strategies. We describe a field study of related video recommendations, where we deploy algorithms to recommend related videos in a movie trailer viewing interface. Our results show that non-personalized algorithms yield the highest click-through rates, while the algorithm prioritizing recency is the strongest in leading to trailer-level user engagement. Our findings suggest the potential to design non-personalized yet effective related item recommendation strategies.

1 INTRODUCTION

Online video consumption has grown to tremendous volumes. YouTube, for instance, reported in early 2017 that its users consume a billion hours of video on its service each day [7].

Online video-oriented services take several forms, including sharing sites like YouTube and Vimeo, social networks like Facebook, and media companies like Comedy Central. Each of these services offers a version of an "up next" recommender that finds and displays related videos which users can click on. The algorithms that recommend these related videos impact what people end up watching [16]. Therefore, understanding what strategies work well is important for improving users' experiences with related video recommenders.

Related video recommendation involves the trade-off among relevance to the seed item, relevance to the user, and non-personalized criteria. The most similar videos might not be the most interesting videos to the user. Some methods focus on identifying similar items by analyzing their visual, audio and textual features [12]; some aim to provide recommendations tailored to users' personal preferences [2, 5]; other methods incorporate non-personalized elements to recommend the most popular items or new releases. Yet, there is comparatively little work that evaluates competing algorithms in this context to better understand this trade-off.

Therefore, we conduct a field experiment where we deploy several algorithms to recommend related items on an interface for viewing movie trailers. Based on a seed trailer that a user is watching, the algorithms recommend additional trailers to watch. All three of our experimental algorithms begin by retrieving a candidate set of similar videos. Two of the algorithms order these candidates by non-personalized criteria (recency and similarity), while the third ranks similar items based on a user-personalized predicted rating. We aim to answer the research question:

In the context of related video recommendation, how do algorithms prioritizing recency, similarity, and predicted rating affect users in terms of their propensity to click and their subsequent engagement?

Figure 1: Trailer viewing interface. The current trailer, Moana, is highlighted on the bottom left. The recommended trailers extend to the right. The screenshot is annotated with labels for the liking, disliking, and wishlisting controls.

In this paper, we present results from an 18-month field experiment, with a focus on how users respond to the different recommendation strategies. We compare two non-personalized methods and one personalized approach by measuring how they impact user engagement. Our results show that the non-personalized algorithms yields higher click-through rates. The algorithm prioritizing recency is the strongest in promoting trailer-level user engagement while the other two perform similarly. Our findings suggest the potential for designing non-personalized yet effective related item recommendation strategies.

2 RELATED WORK

Related item recommendation is an important feature in many recommendation systems. For example, Zhou et al. [16] find that the related videos on Youtube are a major source of video views. Related item recommendations are also used extensively in music recommenders [10], e-commerce [8], or news sites [1] to enhance users' experiences.

Some methods that recommend related items focus on identifying items similar to a seed item. For example, Mei et al. [12] propose a related video recommendation strategy that identifies similar videos through their visual, audio and textual features. Other methods recommend items tailored to users' personal preferences such as their search histories or viewing habits [2, 5]. Bendersky et al. [3] demonstrate that an algorithm that incorporates users' personal preferences leads to longer video watch time than a content-based information retrieval approach.

In our study, we identify similar items using a content-based metric called the tag genome, which computes similarity scores between movies by comparing vectors of latent features [15]. In addition, our personalized algorithm adopts the classic collaborative

filtering approach to compute predicted scores of each similar item and ranks them accordingly [14].

3 METHODS

To learn about related video recommendations, we deployed a field experiment on a non-commercial movie recommendation website [site name]. In this section, we describe the interface changes we made in support of this experiment, and the details of a within-subjects field experiment that compares the outcomes of three different recommendation strategies.

3.1 User Interface Modifications

[site name] offers a rich set of metadata about movies, as well as images from the movie and movie trailers which help people find movies matching their interests [13].

Prior to this study, the trailer viewing interface was a minimal full-screen modal overlay. To facilitate this study, we enhanced the overlay to include several new features: "related trailer" recommendations, "like/dislike" buttons, and the ability to wishlist and rate the movie from the trailer interface (see Figure 1). Like most video watching sites, we auto-advance through the list of recommendations as the user finishes each trailer. Users can open the trailer viewing interface by clicking a play button that we added to movie poster graphics throughout the system.

3.2 Recommendation Algorithms

In this experiment, we developed and deployed three related item algorithms and one random baseline algorithm.

The three related item algorithms each generate trailer recommendations based on a particular trailer that a user is watching. These algorithms first identify a set of the 250 most similar movies to the current movie. To determine similarity, we use a content-based metric called the tag genome [15]. The tag genome computes similarity scores between pairs of movies based on the similarity between latent feature vectors generated by a supervised machine learning process. This process of first filtering to similar items, then applying a ranking function, is a typical approach to related item recommendations [4, 8]. Our algorithms didn't consider features that are specific to trailers; because trailers usually capture the genre and main contents of movies [9], we can identify similar trailers by finding similar movies.

The three related-item algorithms then rank the 250 most similar movies as follows:

- *TagSimilarity*. This non-personalized algorithm ranks trailers in order of their similarity to the seed movie.
- *FilmReleaseDate*. This non-personalized algorithm ranks trailers by release date, newest first.
- *PredictedRating* This personalized algorithm ranks trailers by predicted rating for the current user. We compute predicted ratings using item-item collaborative filtering [14].

The *Baseline* algorithm generates a set of random, non-personalized recommendations, drawn from all movies in the database. The recommendations are not necessarily related to the seed movie.

3.3 Field Experiment and Metrics

We designed an online within-subjects experiment to evaluate the four algorithms. Users in the experiment were randomly assigned one of the four algorithms each time they logged in to the system. The same user could be assigned to different algorithms in different sessions. We chose to design it this way because we can gain more observations for each algorithm, relying on mixed-effects statistical analysis to take account of the dependency of multiple observations from the same user.

We evaluate our algorithms by measuring their effects on user within-session engagement. We examine users' interactions with two groups of trailers: 1) all trailers that users viewed, and 2) trailers that were recommended and clicked on, which we call recommendation-click trailers or RC trailers for convenience.

We use two kinds of user engagement metrics:

- Click-through rate (CTR): the ratio of clicks on recommendations over total trailer views.
- Trailer-level user engagement: each time users watch a trailer, we measure how frequently they take the following actions: (1)*TrailerLiked, TrailerDisliked, WatchedMoreThanHalf* measure user engagement with trailers. They indicate the quality of a recommendation — whether the trailer is enjoyable to watch. TrailerLiked and TrailerDisliked are featured in Figure 1. (2)*WishlistedMovies* measures users' interests in the corresponding movie. It is featured in Figure 1.

In addition, we use three metrics to measure the output of the recommendations made by each algorithm:

- *popularityLastYear*: The number of times the movie was rated in the past year. Higher numbers indicate more popular movies.
- *avgRating*: Average rating of the movie by our users, on a 0.5-5 star scale with half-star increments.
- *ageMonth*: Difference in months between the time of measurement and the release date of the movie. Smaller numbers indicate newer movies.

We choose these metrics because they capture three different aspects of a movie and are easy to interpret and compare.

4 RESULTS

We deployed the new trailer interface and the four algorithms on [site name] on May 5th, 2016 and collected data until January 17th, 2018. Any user who logged in during this period is a potential participant; we restrict analysis to those users who viewed at least one trailer.

The dataset we consider in subsequent analysis consists of 39,400 unique users and 482,963 login sessions. These users visited the site a median of 2 times, viewed trailers 166,959 times, and clicked on recommendations 9,142 times.

4.1 Descriptive Statistics: Recommended Trailers

We gathered data of users' behaviours on the trailer viewing interface. Table 1 summarizes trailer views and trailer-level user engagement. An interesting observation is that users are more likely

Table 1: An overview of users' interactions with all trailers and with RC (recommendation-click) trailers in our experiment.

	Trailer Views	TrailerLiked	TrailerDisliked	WatchedMoreThanHalf	WishlistedMovies
Count (All)	166,959	2,260 (1.35%)	998 (0.59%)	101,059 (60.53%)	3,164 (1.89%)
Count (RC)	9,142	223 (2.44%)	53 (0.58%)	4,078 (44.60%)	572 (6.26%)

to wishlist movies in the trailer interface if they have clicked on a recommendation (6.26% vs. 1.89%).

We also notice that the first item in the recommendation list is the most likely to be clicked. About 18% of trailers that users clicked were in the first position, while items at later positions all have a lower chance. This result is consistent with the findings of several studies on YouTube related video recommendations[11, 16]

4.2 Recommendation Algorithms

The four recommendation algorithms produce different lists of items. We summarize the contents of these recommendations by three properties: avgRating, ageMonth, and popularityLastYear. Their distributions are shown in Figure 2, broken down by algorithm. Unsurprisingly, FilmReleaseDate recommends much newer items than the other algorithms. We also observe that PredictedRating tends to recommend items with higher average ratings and higher popularities than the other algorithms.

We first compare our algorithms based on click-through rate (CTR). According to Figure 3, FilmReleaseDate has the highest CTR, closely followed by TagSimilarity.

Because users can be assigned to different algorithms in different sessions, more active users could dominate our data and skew our observations. Therefore, we perform a statistical test to understand the variations among the similarity-based algorithms more accurately. Specifically, we build a mixed-effect logistic model to predict whether each recommendation in the list will be clicked. This model excludes within-users effects as random effects. The independent variables are the similarity-based algorithms, properties (avgRating, popularityLastYear, age) of the seed video, and position in the list (left-to-right). We exclude the properties of recommendations because they are correlated with the algorithms. Table 2 displays a summary of the model.

Our model has a good fit with AUC = 0.8879. With PredictedRating as the reference group, TagSimilarity and FilmReleaseDate are both more likely to generate recommendations that users would click when users' random effects are excluded. In a pairwise comparison, FilmReleaseDate is not statistically different compared with TagSimilarity in CTR. The model also confirms that position has a significant impact on click throughs. The properties of seed videos only have a slight impact on the result.

We further measure the algorithms based on trailer-level engagement metrics (TrailerLiked, TrailerDisliked, WatchedMoreThanHalf, and WishlistedMovies). According to Figure 4, we observe that both FilmReleaseDate and PredictedRating did a better job in recommending enjoyable trailers and encouraging users to wishlist movies than TagSimilarity. To confirm our observations, we conduct similar statistical tests by building four mixed effects logistic regression models. The four trailer-level metrics are the dependent variables, the algorithms and properties of seed trailers are fixed effects, userID is the random effect, and we vary which algorithm we use as the reference condition. The results are shown in Table 3. In summary, we find that FilmReleaseDate performs better than PredictedRating (p<0.05) and TagSimilarity(p<0.05) in terms of WatchedMoreThanHalf and WishlistedMovies. TagSimilarity has negative coefficients with respect to PredictedRating, although this result is not statistically significant. We don't find significant results for TrailerLiked and TrailerDisliked because the data is limited, so we don't include them in Table 3.

Table 2: Summary of the mixed effect logistic model. PredictedRating is the reference group. Note that the last three predictors are properties of seed videos and their coefficients are standardized. SE stands for standard error. $^{***}p < 0.001$, $^{**}p < 0.01$, $^{*}p < 0.05$

Predictor	Coefficient	SE	P Value
Intercept	-8.1700	0.1503	$< 0.0001^{***}$
TagSimilarity	0.1892	0.0487	0.0078^{**}
FilmReleaseDate	0.2202	0.0495	0.0124^{***}
position	-0.1093	0.0045	$< 0.0001^{***}$
age_seedmovie	0.0001	0.0207	0.9935
avgRating_seedmovie	-0.0479	0.0210	0.0227^{*}
popularity_seedmovie	-0.0194	0.0219	0.3771

Table 3: Coefficients of the mixed effect logistic models that compare algorithms in pairs. Standard errors are included in parentheses. We use algorithm abbreviations: PR (PredictedRating), TS (TagSimilarity), and FRD (FilmReleaseDate). $^{**}p < 0.01$, $^{*}p < 0.05$.

Predictor	WatchedMoreThanHalf	WishlistedMovies
PR(Base) vs TS	-0.0666 (0.0902)	-0.1205 (0.2760)
PR(Base) vs FRD	$0.2496\ (0.0906)^{**}$	$0.4675\ (0.2216)^{*}$
TS(Base) vs FRD	$0.2671\ (0.0915)^{**}$	$0.5186\ (0.2207)^{*}$

5 DISCUSSION AND CONCLUSION

In this research, we conduct a field study to learn which of three approaches to ranking similar movie trailers yields the most click-throughs and trailer-level actions on the clicked-on item.

We find that non-personalized recommendation strategies that rank related items by similarity and recency lead to more user clicks than a personalized strategy that ranks based on predicted rating. This is a surprising result, given the historical reliance on predicted ratings to order recommendations in collaborative filtering systems [14]. One implication is that (at least in this domain) users

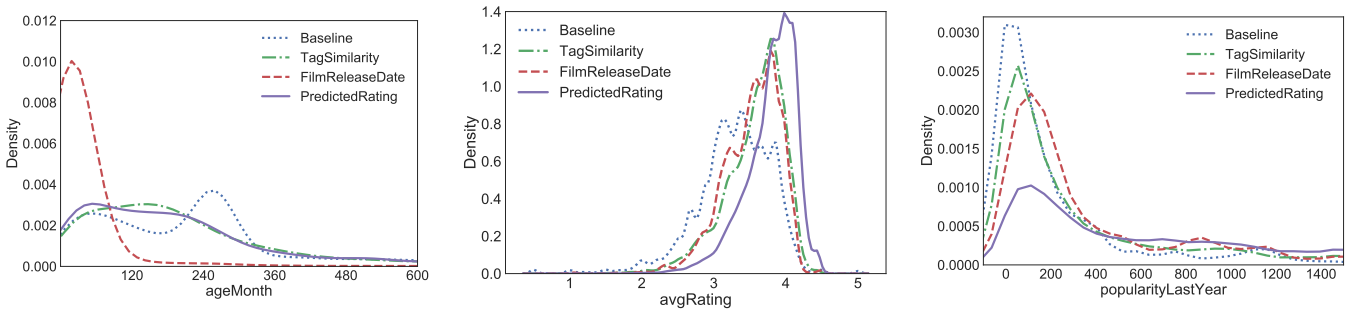

Figure 2: Distributions of the three properties of recommendations: average rating, age (in months), popularity last year, from left to right. The four lines represent the four algorithms. The y axis represents kernel density — the probability of X falling into a certain range is the area under the curve within this range.

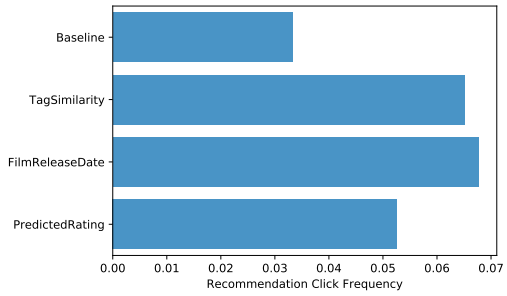

Figure 3: How often the recommendations from each algorithm were clicked. The x axis represents the ratio of clicks on recommendations over total trailer views.

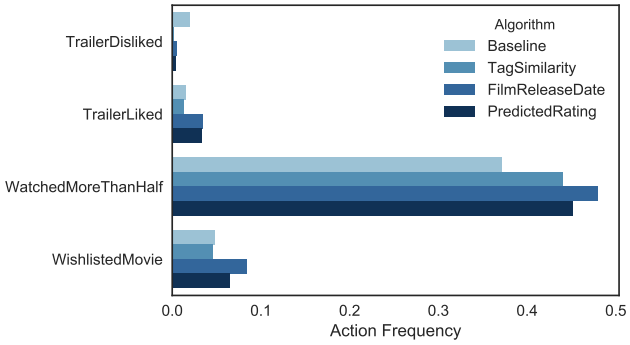

Figure 4: Several measures of utility of the different recommendations, based on the frequency of different actions a user may take after clicking on a recommended trailer. The action frequency measures the percentage of time that users take the given action in this context.

prioritize item relevance and recency over personal relevance. It also seems that the higher average ratings or popularities from PredictedRating (see Figure 2) do not necessarily make its recommendations more appealing to users. Davidson et al. [5] contributed a similar finding in YouTube, showing that related videos based on similarity have higher click-through rates than top rated ones. The success of ranking by recency may be domain-specific: this may be the result of industry efforts to optimize the appealing qualities of modern movie trailers [6], or because users of our experimental system are simply most interested in using movie trailers to learn about new releases. This result speaks to the importance of incorporating domain-specific insights in the development of recommendation algorithms. The success of ranking by similarity is also surprising. In prior research, similar items are usually identified so that they can be passed to a more sophisticated, user-personalized ranking method (e.g., [5, 8]).

Looking one step past click-through rates, we see a somewhat different picture. Users are more likely to watch more than half of the trailer or wishlist the movie when the recommendation comes from the FilmReleaseDate algorithm. PredictedRating and TagSimilarity perform similarly with respect to these metrics in the statistical test (though PredictedRating has an edge in activity counts). Therefore, ranking by similarity may be successful only superficially: it appears to be better at boosting the click-through rate than at finding interesting content.

The results of this study suggest the potential of building non-personalized yet effective related item recommenders. There are many ways to build on this work. The algorithms we test are simple; it is future work to test more sophisticated similarity or ranking algorithms. Our results may be domain specific; it is future work to test the effectiveness of prioritizing recency in other domains. We think the results presented here contribute an interesting data point to the investigation of related item recommendation algorithms, and we look forward to seeing more empirical results in this research space.

REFERENCES

[1] Deepak Agarwal, Bee-Chung Chen, Pradheep Elango, and Raghu Ramakrishnan. 2013. Content Recommendation on Web Portals. *Commun. ACM* 56, 6 (June 2013), 92–101. https://doi.org/10.1145/2461256.2461277

[2] Shumeet Baluja, Rohan Seth, D. Sivakumar, Yushi Jing, Jay Yagnik, Shankar Kumar, Deepak Ravichandran, and Mohamed Aly. 2008. Video Suggestion and Discovery for Youtube: Taking Random Walks Through the View Graph. In *Proceedings of the 17th International Conference on World Wide Web (WWW '08)*. ACM, New York, NY, USA, 895–904. https://doi.org/10.1145/1367497.1367618

[3] Michael Bendersky, Lluis Garcia Pueyo, Jeremiah J. Harmsen, Vanja Josifovski, and Dima Lepikhin. 2014. Up next: retrieval methods for large scale related video suggestion. In *KDD*.

[4] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep Neural Networks for YouTube Recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems (RecSys '16)*. ACM, New York, NY, USA, 191–198. https://doi.org/10.1145/2959100.2959190

[5] James Davidson, Benjamin Liebald, Junning Liu, Palash Nandy, Taylor Van Vleet, Ullas Gargi, Sujoy Gupta, Yu He, Mike Lambert, Blake Livingston, and Dasarathi Sampath. 2010. The YouTube Video Recommendation System. In *Proceedings of the Fourth ACM Conference on Recommender Systems (RecSys '10)*. ACM, New York, NY, USA, 293–296. https://doi.org/10.1145/1864708.1864770

[6] Josh Dzieza. 2015. The Star Wars history of trailers. https://www.theverge.com/2015/12/10/9882404/star-wars-trailers-movie-marketing-youtube-disney

[7] Cristos Goodrow. 2017. You know what's cool? A billion hours. https://youtube.googleblog.com/2017/02/you-know-whats-cool-billion-hours.html

[8] J. Katukuri, T. Könik, R. Mukherjee, and S. Kolay. 2014. Recommending similar items in large-scale online marketplaces. In *2014 IEEE International Conference on Big Data (Big Data)*. 868–876. https://doi.org/10.1109/BigData.2014.7004317

[9] Lisa Kernan. 2004. *Coming Attractions: Reading American Movie Trailers*. University of Texas Press. Google-Books-ID: 7gu0ZU7K834C.

[10] Peter Knees and Markus Schedl. 2013. A Survey of Music Similarity and Recommendation from Music Context Data. *ACM Trans. Multimedia Comput. Commun. Appl.* 10, 1 (Dec. 2013), 2:1–2:21. https://doi.org/10.1145/2542205.2542206

[11] Dilip Kumar Krishnappa, Michael Zink, Carsten Griwodz, and P\a al Halvorsen. 2015. Cache-Centric Video Recommendation: An Approach to Improve the Efficiency of YouTube Caches. *ACM Trans. Multimedia Comput. Commun. Appl.* 11, 4 (June 2015), 48:1–48:20. https://doi.org/10.1145/2716310

[12] Tao Mei, Bo Yang, Xian-Sheng Hua, and Shipeng Li. 2011. Contextual Video Recommendation by Multimodal Relevance and User Feedback. *ACM Trans. Inf. Syst.* 29, 2 (April 2011), 10:1–10:24. https://doi.org/10.1145/1961209.1961213

[13] Theodora Nanou, George Lekakos, and Konstantinos Fouskas. 2010. The effects of recommendations' presentation on persuasion and satisfaction in a movie recommender system. *Multimedia Systems* 16, 4-5 (Aug. 2010), 219–230. https://doi.org/10.1007/s00530-010-0190-0

[14] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based Collaborative Filtering Recommendation Algorithms. In *Proceedings of the 10th International Conference on World Wide Web (WWW '01)*. ACM, New York, NY, USA, 285–295. https://doi.org/10.1145/371920.372071

[15] Jesse Vig, Shilad Sen, and John Riedl. 2012. The Tag Genome: Encoding Community Knowledge to Support Novel Interaction. *ACM Trans. Interact. Intell. Syst.* 2, 3 (Sept. 2012), 13:1–13:44. https://doi.org/10.1145/2362394.2362395

[16] Renjie Zhou, Samamon Khemmarat, and Lixin Gao. 2010. The Impact of YouTube Recommendation System on Video Views. In *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement (IMC '10)*. ACM, New York, NY, USA, 404–410. https://doi.org/10.1145/1879141.1879193