Global Journal of Computer Science and Technology: G
Interdisciplinary
Volume 18 Issue 1 Version 1.0 Year 2018
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

Video Recommendation System for YouTube Considering user's Feedback

By Md. Shamim Reza Sajib, Md. Ariful Islam Malik, Md. Ashraful Islam & Sudip Kumar Halder

Bangladesh University of Business & Technology

Abstract- Youtube is the most video sharing and viewing platform in the world. As there are many people of different tastes, hundreds of categories of videos can be found on YouTube while thousands of videos of each. So, when the site recommends videos for a user it takes some issues which fill the needs of the user. Most of the time a user watches videos related to the previously watched video. But sometimes user's mood changes with time or weather. A user may not hear a song in the whole year but can search the song on a rainy day. Another case a user may watch some types of videos at day but another type of videos at night or another at midnight. In this paper, we propose a recommendation system considering some attributes like weather, time, month to understand the dynamic mood of a user. Each attribute is assigned a weight calculated by performing a survey on some YouTube users.

Keywords: youtube video recommendation system, weighted attribute based video recommendation system, youtube watch-list recommendation, youtube video suggestion.

GJCST-G Classification: H.5.2

VIDEORECOMMENDATIONSYSTEMFORYOUTUBECONSIDERINGUSERSFEEDBACK

Strictly as per the compliance and regulations of:

Video Recommendation System for YouTube Considering user's Feedback

Md. Shamim Reza Sajib [α], Md. Ariful Islam Malik [σ], Md. Ashraful Islam [ρ] & Sudip Kumar Halder [Ѡ]

Abstract- Youtube is the most video sharing and viewing platform in the world. As there are many people of different tastes, hundreds of categories of videos can be found on YouTube while thousands of videos of each. So, when the site recommends videos for a user it takes some issues which fill the needs of the user. Most of the time a user watches videos related to the previously watched video. But sometimes user's mood changes with time or weather. A user may not hear a song in the whole year but can search the song on a rainy day. Another case a user may watch some types of videos at day but another type of videos at night or another at midnight. In this paper, we propose a recommendation system considering some attributes like weather, time, month to understand the dynamic mood of a user. Each attribute is assigned a weight calculated by performing a survey on some YouTube users. Most recently viewed videos is assigned heavy weight and weather is assigned lower. This recommendation system will make YouTube more user-friendly, dynamic and acceptable.

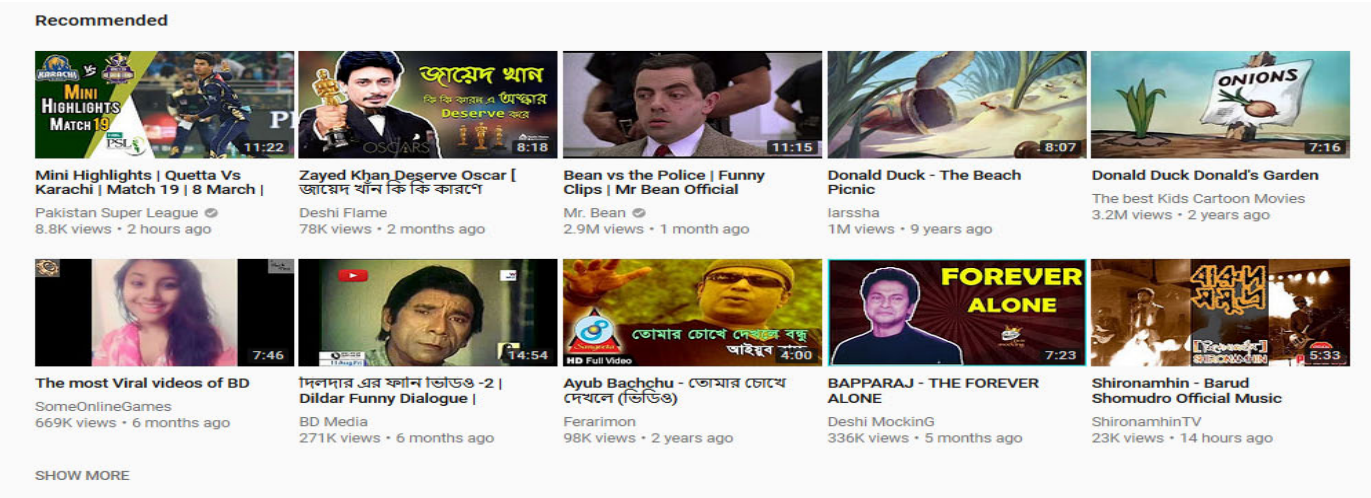
Keywords: youtube video recommendation system, weighted attribute based video recommendation system, youtube watch-list recommendation, youtube video suggestion.

I. Introduction

Since the launch of YouTube in 2005, it has become a popular destination site for users to find videos as well as share their videos. YouTube has earned worldwide popularity in the past decade. Thousands of users watch and upload millions of videos daily. So YouTube has a recommendation system for each user individually. But the mood and need of a user is very dynamic and changes dramatically. So it is the challenge of the recommendation system to understand the current mood and need of a user and suggest that types of videos that the user wants. As YouTube recommends a very few videos from thousands of videos, they are very selective for this recommendation system. The system recommends personalized sets of videos to users based on their recent and frequent activity on the site, subscribed channel, etc [1]. The recommendation made by the system is reasonably recent and fresh, as well as diverse and relevant to the users recent action. But user mood can change at any time. Let a user generally does not watch songs of the rainy day. But on a rainy day he may search for a favoured rainy day song that he watched many days ago or not at all. In another case: a user watches many videos regularly but some of those he may watch at mid of a day, some of them he mostly watches at early night and some of them at late night. So user's mood can change at different time of a day. So when the system recommends videos, it should also consider the current time and what videos mostly he watches at that time. So dealing with this dynamic mood and need of a user is the prime challenge of this recommendation system.

In the paper, a new recommendation system is proposed where we consider some attributes for recommending videos along with most recently and most frequently viewed videos. The new attributes are time, month and weather. As each of them is not equally significant for deciding which video a user may watch, a weight assigned to each attribute. The weight is calculated by surveying some YouTube users. Most of the users feel that they expect a video which is related to the previously of frequently watched videos. So a high weight is assigned to these two attributes. Some users feel that they watch different types of videos at the different time of the day. So a moderate weight is assigned to this attribute. A less number of users feel that they watch some videos in a particular time of the year but not in the other time like they watch rainy day song in rainy weather but not in the cold weather. So this attribute is assigned a less weight. But the highest weight is assigned to a new video of a channel that the user subscribed and watches the videos on that channel regularly. So, when the system recommends videos, the weighted sum of related videos is calculated. The highest valued videos are recommended for the user and top N videos are shown on the home page like the method [5].

Author α σ ρ Ѡ: Department of Computer Science & Engineering Bangladesh University of Business & Technology (BUBT).
e-mails: sajib1717@gmail.com, malikariful@gmail.com, ashacse42@gmail.com, sudiphalderruet@gmail.com


Recommended

Mini Highlights | Quetta Vs Karachi | Match 19 | 8 March |
Pakistan Super League
8.8K views • 2 hours ago

Zayed Khan Deserve Oscar [জায়েদ খাঁন কি কি কারণে
Deshi Flame
78K views • 2 months ago

Bean vs the Police | Funny Clips | Mr Bean Official
Mr. Bean
2.9M views • 1 month ago

Donald Duck - The Beach Picnic
larssha
1M views • 9 years ago

Donald Duck Donald's Garden
The best Kids Cartoon Movies
3.2M views • 2 years ago

The most Viral videos of BD
SomeOnlineGames
669K views • 6 months ago

দিলদার এর ফান ভিডিও -2 | Dildar Funny Dialogue |
BD Media
271K views • 6 months ago

Ayub Bachchu - তোমার চোখে দেখলে (ভিডিও)
Ferarimon
98K views • 2 years ago

BAPPARAJ - THE FOREVER ALONE
Deshi MockinG
336K views • 5 months ago

Shironamhin - Barud Shomudro Official Music
ShironamhinTV
23K views • 14 hours ago

SHOW MORE

Fig.1: Recommended Videos of a user

II. Proposed Method

As stated above, we do not only consider a user's recent activities, we also consider some other important attributes to make the system more dynamic and to make user understand why a video is recommended to them. The method is designed in four stages: i) Weight Calculation, ii) Generating Related Videos iii) Generating Recommended Candidates and iv) Finding recommended videos by calculating a weighted sum.

a) Input Data

During the generation of personalized video recommendations, we consider some data sources. In general, there are two broad classes of data to consider: 1) content data, such as the raw video streams and video metadata such as title, description, etc. and 2) user activity data, which we can further divide into explicit and implicit. Explicit activities include rating a video, favoriting/liking a video, or subscribing to an uploader. Implicit activities are datum generated as a result of users watching and interacting with videos. We also define some others behavior of a user as explicit data such as the specific time, date and weather when a video the user watches. But user data only captures a fraction of a users activity on the site and indirectly measures a users engagement and happiness. Because a user may watch a video for a long time, but that cannot conclude that actually he/she has liked it. The implicit activities data is generated asynchronously and can be incomplete. So it is very challenging to deal with this huge amount of discrete and noisy data.

b) Assigning Weight

There may be a large number of input data for further processing. Among them, all the videos are not equally important. So, we have to find out the significant ones for further processing. For this purpose, a weight is assigned to each attribute based on a number of user's feedback. We take the feedback of the users on some questions like: Whether user's mood or taste vary at different times of the day or with the change of weather. The questions and the survey result is given in Table I. The weight is calculated considering the value of the answers of the users. The equation for calculating weight is

Table I: Survey Result on the Questions asked to some Youtube users

Question	No. Of User	Always Yes	Most Often Yes	No
Do You Want New Videos Uploaded by Subscribed Channel You Watch Regularly	250	203	38	9
Do You Want New Videos Uploaded by Subscribed Channel You Watch Irregularly	250	129	91	30
Do You Want Videos Related to Recently Watched Videos	250	147	87	16
Do You Want Videos Related to Frequently Watched Videos	250	162	77	11
Do You Watch Different Types of Videos at Different Time	250	113	78	59
Will You Be Happy if a Rain Song is Recommended on a Rainy Day	250	109	95	46

$$W_i = A_i + 0.8 * O_i - N_i$$

Where W_i is the weight of an attribute,

$$A_i = \frac{NoofUsersAnsweredAlwaysYes}{TotalNumberofUser}$$

$$O_i = \frac{NoofUsersAnsweredMostOftenYes}{TotalNumberofUser}$$

$$N_i = \frac{NoofUsersAnsweredNo}{TotalNumberofUser}$$

O_i is multiplied with .8 as its contribution of the total weight should be less than the contribution of always yes. N_i is subtracted from the weight as those users do not want those videos. So, for the first attribute which is The Videos Uploaded by Subscribed Channel That a User Watches Regularly, its weight should be W_{sr} = (203/250) + 0:8 * (38/250) - (9/250) = 0.90. Another attribute which is a new video by the channel a user follows irregularly, the weight will be W_{si}= (119/250) + . 8 * (81/250) - (50/250) = 0.54. Thus the weight is calculated for each attribute. The most significant attribute that affects the user mind mostly, gets the highest weight. The final value is calculated by multiplying the attribute value which is 0 or 1 with the corresponding weight. Suppose a video candidate is generated which is newly uploaded by a subscribed channel watched by the user regularly, the user watches that type videos at night, the user watches that type of videos recently but not frequently. The current time the user sign-in is day, and it is a hot day. Then the attribute value for $A_{sr} = 1$, $A_{si} = 0$, $A_r = 1$, $A_t = 0$, $A_r = 0$.

c) *Generating Related Videos*

For this work, we are using the method proposed by [1]. We are not proposing a new method for finding related videos. In this stage of recommendation, we have to construct a mapping from a video vi to a set of similar or related videos R_i The similar videos are defined as those that a user is likely to watch after having watched the given seed video v. For computing this mapping [1] has used a well-known technique known as association rule mining [2]. They also consider the duration of a session of a user and count for each pair of videos (v_i, v_j) how often they were co-watched within sessions. This co visitation count is denoted by *cij* and they calculate the relatedness score of v_j to a base video (v_i by the following equation.

$$r(v_i, v_j) = \frac{(c_{ij})}{f(v_i v_j)}$$

where c_i and c_j are the total occurrence counts across all sessions for videos v_i and v_j , respectively. $f(v_i , v_j)$ is a normalization function that takes the global popularity of both the seed video and the candidate video into

$$f(v_i, v_j) = c_i.c_j$$

account. One of the simplest normalization functions is to simply divide by the product of the videos global popularity:

One of the simplest normalization functions is to simply divide by the product of the videos global popularity $f(v_i, v_j) = c_i, c_j$. Other normalization functions are possible. See [6] for an overview of possible choices. [3] used a video co-view graph which represents the videos watched by some users. They then use it for generating related videos. They then pickup N videos from a number of related videos based on the value or relatedness score. N is variable depending on a threshold. If there are many videos satisfying the relatedness score, N will be larger. So this system face difficulty generating related videos which has a lower view count. There may be some additional problem like presentation bias, noisy watch data etc.

d) *Generating Recommendation Candidates*

For computing personalized recommendations, the related videos association rules are combined with a user's personal activity on site. This can include videos that were watched recently, frequently or liked or added to playlists. The union of those videos is called seed set. There may be many videos which can come with several categories, but each video is present only one time in the seed set. Assume the generated seed set S; we expand the related video graph G in order to find the related and connected videos. For each video Vi in the seed set, assume its related video Ri. The related video set Ci will be

$$C_i(S) = \bigcup_{vi \in S} R_i$$

In many cases, computing C1 is sufficient for generating a set of candidate recommendations that is large and diverse enough to yield interesting recommendations. However, in practice the related videos for any videos tend to be quite narrow, often highlighting other videos that are very similar to the seed video. This can lead to equally narrow recommendations which can achieve the goal of recommending content close to the users interest, but fail to recommend videos which are truly new to the user. This problem can arise after generating recommendation candidates by this process. To get rid of that possibility, a distance of n will be traversed through the related video graphs to find more candidates. Due to the high branching factor of the related videos graph, we found that expanding over a small distance yielded a broad and diverse set of recommendations even for users with a small seed set. That's why the value of n should be set a smaller value. A large value of n can generate a huge candidate set which will be time consuming and unnecessary. Note that each video in the candidate set is associated with one or more videos in the seed set. We keep track of these seed to candidate associations for ranking purposes and to provide explanations of the recommendations to the user. A deep neural network based method is used by [4] to generate recommended candidates. They also consider related videos for candidate generation, but they have used a deep neural network to generate the best candidates from the millions of videos. But their method need high computational resources and millions of data. In the proposed method we use the same process proposed by [1] for generating candidates.

e) Recommended Videos

After generating recommendation candidates, the recommended set may contain many videos. But the designed user interface shows only some of them. So the question is how they should be selected. After the generation step has produced a set of candidate videos they are scored and ranked using a variety of signals. [1] Considers three different signals i) Quality, ii) user specificity, iii) diversification.

The proposed method also uses these signals with considering some other attributes. For video quality, the proposed method considers view count (the total number of times a video has been watched), the ratings of the video, commenting, favoring and sharing activity around the video, and upload time. Considering all these things, the proposed method calculates the value of quality Q_i like the method [1].

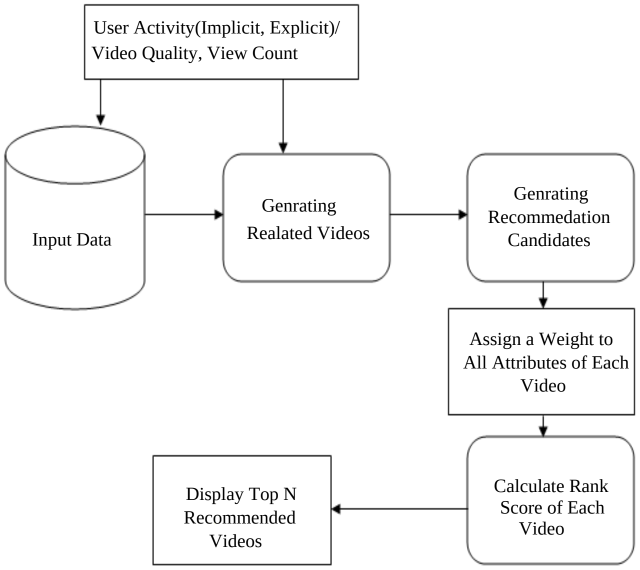


Fig. 2: System Architecture of Proposed Method

User specificity is a unique user's taste and preferences. For this the current method only considers users watch history, such as view count and time of watch. But these attributes are not enough to detect user's mood and recommend the desired videos. For this we propose to consider some other attributes described earlier. We propose to consider subscribed channel videos, recently watched videos, specific time when a video has been watched. The value of user specification can be generated by equation. Considering all these things the proposed method calculates the value of user specificity of a video v_i is:

$$U_i = W_{sr}.A_{sr} + W_{si}.A_{si} + W_r.A_r + W_t.A_t$$

Using a linear combination of these signals we generate a ranked list of the candidate videos. As YouTube only displays a small number of recommendations between 4 to 60, we have to generate a recommendation lists. In this stage diversity is considered. Since a user generally has interest in multiple different topics at differing times, videos that are too similar to each other are removed at this stage to further increase diversity. For this diversity we consider weather information. In a rainy day a video of rainy song may be recommended or a snowy video may be recommended on a snow falling evening though the user does watch this types of videos very often. After generating those videos the value of W_w is assigned to a video v_i Considering all these issues that can affect a user mind, we generate an equation that calculates the rank of a video from the video set of recommended candidates. The equation is the sum of all three signals considering all the attributes described. If the system shows N videos from the set, the highest ranked videos will be displayed. The rank of a video v_i from the candidate set can be calculated by the following equation:

$$R_{vi} = Q_{vi} + U_{vi} + D_{vi}$$

Then the top N scored videos will be displayed in the user interface.

III. Implementation

[1] Choose a batch-oriented pre-computation approach rather than on-demand calculation of recommendations. The proposed method does the on-demand calculation of recommendation. As there are millions of data in the logs, the most significant downside of this approach is the delay between generating and serving a particular recommendation data set. To reduce the problem, we propose to use a pre-calculated recommendations. This recommenddations are updated regularly so there is no chance of recommending same videos again and again. The actual implementation of YouTubes recommendation system can be divided into three main parts: 1) data collection, 2) recommendation generation and 3) recommendation serving. We collect input data from many users manually from their YouTube logs and store

those in a big table [7]. Then we select the top N videos by the system described in section II.

IV. Experimental Result

A large number of user data is experimented by the method. User data are collected from the watch history of a large number of users for a period of three weeks (21 days). The data then processed for each individual users and recommended videos are generated by the proposed method. The result then analysed by the feedback of the users. As we cannot experiment the result by the random users of YouTube, we manually generate result for each individual users and ask which video he/she may click if the video appeared in the recommendation sector of YouTube home page. Based on some user's feedback, some results are shown in. The proposed method has been experimented on more than 100 users. According to their feedback they would click around 75% of the recommended video. At the same time they would click only 63% video recommended by current recommendation system. Since we cannot implement our method in YouTube, we calculate our result manually considering user's feedback and their feedback on current recommendation system. There may be different result in real case. As the recommendation system is designed considering user's feedback, there may be many users who do not think in the same way. It is very difficult to understand user's need as millions of user's do not think the same way. But this recommendation system is accepted by most of the users we experimented.

Table II: User's Feedback on Recommended Videos by the Proposed Method

User No	Recommended Videos	Videos He May Watch	Videos He May Ignore	Success Rate
User 1	43	28	15	65.11%
User 2	52	40	12	76.92%
User 3	33	25	8	76.76%
User 4	55	43	12	78.18%

V. Conclusion

Recommending suitable video to a user is a very challenging task as the mood of the user is very dynamic. In this paper, we consider almost every attribute that can affect user mood. It makes our recommendation system more friendly, reliable and dynamic. But all the values of the attributes depend on the previous activities of a user. So it may not perform well while recommending videos to a new user or the users who are not signed in. A user's mood can change rapidly on some incident, our system may fail to understand that. But our recommendation system can deal with almost every other possible cases. Though we consider five attributes, all of them are not equally important identifying the rank of a video. So we assign a weight to each attribute according to the significance of that attribute to the user. After that a final value is calculated for a video considering all these facts. The highest valued videos will be recommended to the user. Selection of attributes that take care of the dynamic behavior and the calculating process makes our proposed system more robust, dynamic and reliable.

References Références Referencias

1. Davidson, James Liebald, Benjamin Liu, Junning Nandy, Palash Van Vleet, Taylor Gargi, Ullas Gupta, Sujoy He, Yu Lambert, Michel Livingston, Blake Sampath, Dasarathi. (2010). The YouTube video recommendation system. 293-296.10.1145/1864708. 1864770.
2. F. Chang, J. Dean, S. Ghemawat, W. C. Hsieh, D. A. Wallach, M. Burrows, T. Chandra, A. Fikes, and R. E. Gruber. Bigtable: A distributed storage system for structured data. In USENIX 07, pages 205218, 2006.
3. Baluja, Shumeet, et al."Video suggestion and discovery for youtube: taking random walks through the view graph." Proceedings of the 17th international conference on World Wide Web. ACM, 2008.
4. Covington, Paul, Jay Adams, and Emre Sargin. "Deep neural networks for youtube recommendations." Proceedings of the 10th ACM Conference on Recommender Systems. ACM, 2016.
5. M. Deshpande and G. Karypis. Item-based top-n recommendation algorithms. ACM Trans. Inf. Syst., 22 (1):143177, 200.
6. E. Spertus, M. Sahami, and O. Buyukkokten. Evaluating similarity measures: a large-scale study in the orkut social network. In KDD 05, pages 678684, New York, NY, USA, 2005. ACM.
7. F. Chang, J. Dean, S. Ghemawat, W. C. Hsieh, D. A.Wallach, M. Burrows, T. Chandra, A. Fikes, and R. E. Gruber. Bigtable: A distributed storage system for structured data. In USENIX 07, pages 205218, 2006.