

# 1 Video Recommendation System for YouTube Considering user's 2 Feedback

3 Md. Shamim Reza Sajib<sup>1</sup>, Md. Ariful Islam Malik<sup>2</sup> and Md. Ashraful Islam<sup>3</sup>

4 1

5 *Received: 8 December 2017 Accepted: 1 January 2018 Published: 15 January 2018*

6

---

## 7 **Abstract**

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

20

---

21 **Index terms**— youtube video recommendation system, weighted attribute based video recommendation  
22 system, youtube watch-list recommendation, youtube video suggestion

## 23 **1 Introduction**

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

39 In the paper, a new recommendation system is proposed where we consider some attributes for recommending  
40 videos along with most recently and most frequently viewed videos. The new attributes are time, month and  
41 weather. As each of them is not equally significant for deciding which video a user may watch, a weight assigned  
42 to each attribute. The weight is calculated by surveying some YouTube users. Most of the users feel that they  
43 expect a video which is related to the previously of frequently watched videos. So a high weight is assigned to

## 5 C) GENERATING RELATED VIDEOS

---

44 these two attributes. Some users feel that they watch different types of videos at the different time of the day.  
45 So a moderate weight is assigned to this attribute. A less number of users feel that they watch some videos  
46 in a particular time of the year but not in the other time like they watch rainy day song in rainy weather but  
47 not in the cold weather. So this attribute is assigned a less weight. But the highest weight is assigned to a  
48 new video of a channel that the user subscribed and watches the videos on that channel regularly. So, when  
49 the system recommends videos, the weighted sum of related videos is calculated. The highest valued videos are  
50 recommended for the user and top N videos are shown on the home page like the method [5].

## 51 2 Proposed Method

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

## 56 3 a) Input Data

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

## 67 4 b) Assigning Weight Table I: Survey Result on the Questions 68 asked to some

### 69 Youtube users

70 There may be a large number of input data for further processing. Among them, all the videos are not equally  
71 important. So, we have to find out the significant ones for further processing. For this purpose, a weight is  
72 assigned to each attribute based on a number of user's feedback. We take the feedback of the users on some  
73 questions like: Whether user's mood or taste vary at different times of the day or with the change of weather.  
74 The questions and the survey result is given in Table ?? value of the answers of the users. The equation for  
75 calculating weight is  $W_i = A_i + 0.8 * O_i - N_i$

76 Where  $W_i$  is the weight of an attribute,  $O_i$  is multiplied with .8 as its contribution of the total weight should  
77 be less than the contribution of always yes.  $N_i$  is subtracted from the weight as those users do not want those  
78 videos.  $A_{si} = 0$ ,  $A_r = 1$ ,  $A_t = 0$ ,  $A_w = 0$ .

## 79 5 c) Generating Related Videos

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

87 where  $c_i$  and  $c_j$  are the total occurrence counts across all sessions for videos  $v_i$  and  $v_j$ , respectively.  $f(v_i, v_j)$   
88 is a normalization function that takes the global popularity of both the seed video and the candidate video  
89 into account. One of the simplest normalization functions is to simply divide by the product of the videos global  
90 popularity:

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

---

## 98 6 d) Generating Recommendation Candidates

99 For computing personalized recommendations, the related videos association rules are combined with a user's  
100 personal activity on site. This can include videos that were watched recently, frequently or liked or added to  
101 playlists. The union of those videos is called seed set. There may be many videos which can come with several  
102 categories, but each video is present only one time in the seed set. Assume the generated seed set  $S$ ; we expand the  
103 related video graph  $G$  in order to find the related and connected videos. For each video  $V_i$  in the seed set, assume  
104 its related video  $R_i$ . The related video set  $C_i$  will be In many cases, computing  $C_1$  is sufficient for generating a set  
105 of candidate recommendations that is large and diverse enough to yield interesting recommendations. However,  
106 in practice the related videos for any videos tend to be quite narrow, often highlighting other videos that are  
107 very similar to the seed video. This can lead to equally narrow recommendations which can achieve the goal  
108 of recommending content close to the users interest, but fail to recommend videos which are truly new to the  
109 user. This problem can arise after generating recommendation candidates by this process. To get rid of that  
110 possibility, a distance of  $n$  will be traversed through the related video graphs to find more candidates. Due to  
111 the high branching factor of the related videos graph, we found that expanding over a small distance yielded a  
112 broad and diverse set of recommendations even for users with a small seed set. That's why the value of  $n$  should  
113 be set a smaller value. A large value of  $n$  can generate a huge candidate set which will be time consuming and  
114 unnecessary. Note that each video in the candidate set is associated with  $N_i = N$  of  $U$  users.  $N$  is the total number of users  
115 in the seed set. We keep track of these seed to candidate associations for ranking purposes  
116 and to provide explanations of the recommendations to the user. A deep neural network based method is used  
117 by [4] to generate recommended candidates. They also consider related videos for candidate generation, but  
118 they have used a deep neural network to generate the best candidates from the millions of videos. But their  
119 method need high computational resources and millions of data. In the proposed method we use the same process  
120 proposed by [1] for generating candidates.

## 122 7 e) Recommended Videos

123 After generating recommendation candidates, the recommended set may contain many videos. But the designed  
124 user interface shows only some of them. So the question is how they should be selected. After the generation  
125 step has produced a set of candidate videos they are scored and ranked using a variety of signals.

126 [1] Considers three different signals i) Quality, ii) user specificity, iii) diversification.

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

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

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

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

## 150 8 III.

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

## 11 CONCLUSION

158 input data from many users manually from their YouTube logs and store those in a big table [7]. Then we select  
159 the top N videos by the system described in section II.U i = W sr .A sr + W si .A si + W r .A r + W t .A t R  
160 vi = Q vi + U vi + D vi

## 161 9 IV.

## 162 10 Experimental Result

163 A large number of user data is experimented by the method. User data are collected from the watch history of  
164 a large number of users for a period of three weeks (21 days). The data then processed for each individual users  
165 and recommended videos are generated by the proposed method. The result then analysed by the feedback of  
166 the users. As we cannot experiment the result by the random users of YouTube, we manually generate result for  
167 each individual users and ask which video he/she may click if the video appeared in the recommendation sector of  
168 YouTube home page. Based on some user's feedback, some Since we cannot implement our method in YouTube,  
169 we calculate our result manually considering user's feedback and their feedback on current recommendation  
170 system. There may be different result in real case. As the recommendation system is designed considering user's  
171 feedback, there may be many users who do not think in the same way. It is very difficult to understand user's  
172 need as millions of user's do not think the same way. But this recommendation system is accepted by most of  
173 the users we experimented.

174 V.

## 175 11 Conclusion

176 Recommending suitable video to a user is a very challenging task as the mood of the user is very dynamic. In this  
177 paper, we consider almost every attribute that can affect user mood. It makes our recommendation system more  
178 friendly, reliable and dynamic. But all the values of the attributes depend on the previous activities of a user. So  
179 it may not perform well while recommending videos to a new user or the users who are not signed in. A user's  
180 mood can change rapidly on some incident, our system may fail to understand that. But our recommendation  
181 system can deal with almost every other possible cases. Though we consider five attributes, all of them are  
182 not equally important identifying the rank of a video. So we assign a weight to each attribute according to the  
183 significance of that attribute to the user. After that a final value is calculated for a video considering all these  
184 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.

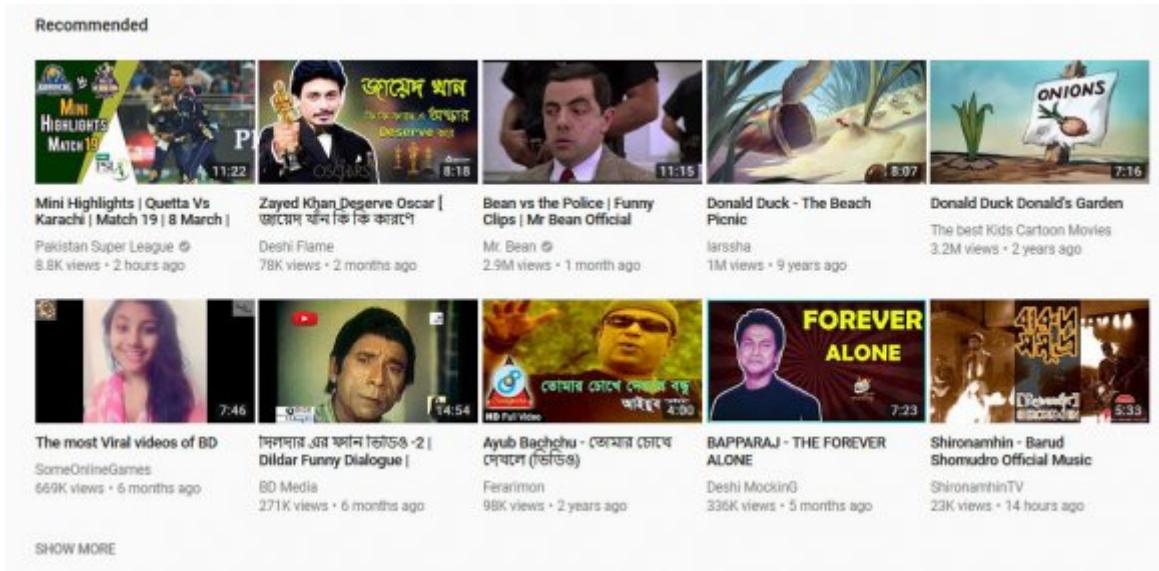


Figure 1: Fig. 1 :

185

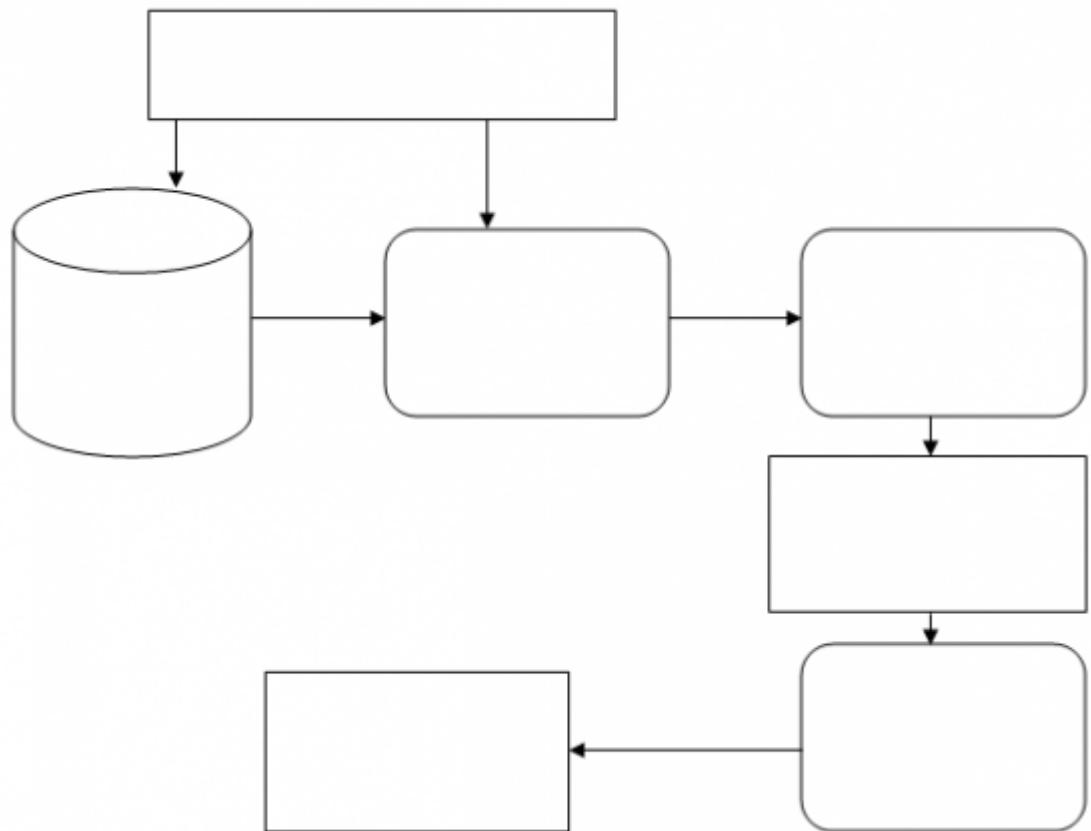


Figure 2:

## 11 CONCLUSION

---

Question	No. Of User	Always Yes	Most Yes	No Of- ten
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

Figure 3:

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  
 $W$

[Note:  $si = (119/250) + .8 * (81/250) - (50/250) = 0.54$ . 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  $sr = 1$ ,]

Figure 4:

---

## II

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 recommendation system.

the Proposed Method

63%video recommended by current

Figure 5: Table II :

## 11 CONCLUSION

---

---

186 [Chang et al. ()] 'Bigtable: A distributed storage system for structured data'. F Chang , J Dean , S Ghemawat ,  
187 W C Hsieh , D A Wallach , M Burrows , T Chandra , A Fikes , R E Gruber . *USENIX 07*, 2006. p. 205218.

188 [Chang et al. ()] 'Bigtable: A distributed storage system for structured data'. F Chang , J Dean , S Ghemawat ,  
189 W C Hsieh , D A Wallach , M Burrows , T Chandra , A Fikes , R E Gruber . *USENIX 07*, 2006. p. 205218.

190 [Covington et al. ()] 'Deep neural networks for youtube recommendations'. Paul Covington , Jay Adams , Emre  
191 Sargin . *Proceedings of the 10th ACM Conference on Recommender Systems*, (the 10th ACM Conference on  
192 Recommender Systems) 2016. ACM.

193 [Spertus et al. ()] 'Evaluating similarity measures: a large-scale study in the orkut social network'. E Spertus ,  
194 M Sahami , O Buyukkokten . *KDD 05*, (New York, NY, USA) 2005. ACM. 678684.

195 [Deshpande and Karypis (200)] 'Item-based top-n recommendation algorithms'. M Deshpande , G Karypis . *ACM  
196 Trans. Inf. Syst.* 200. 22 (1) p. 143177.

197 [Davidson et al. ()] *The YouTube video recommendation system*, James Davidson , Benjamin Liebald , Junning  
198 Liu , Palash Nandy , Taylor Van Vleet , Ullas Gargi , Sujoy Gupta , Yu He , Michel Lambert , Blake  
199 Livingston , Sampath , Dasarathi . 10.1145/1864708.1864770. 2010. p. .

200 [Baluja ()] 'Video suggestion and discovery for youtube: taking random walks through the view graph'. Shumeet  
201 Baluja . *Proceedings of the 17 th international conference on World Wide Web*, (the 17 th international  
202 conference on World Wide Web) 2008. ACM.