Online Friend Recommendation through Personality Matching and Collaborative Filtering

Li Bian
Media Laboratory
Massachusetts Institute of Technology
Cambridge, MA, USA
libian@media.mit.edu

Henry Holtzman
Media Laboratory
Massachusetts Institute of Technology
Cambridge, MA, USA
holtzman@media.mit.edu

***Abstract*—Most social network websites rely on people's proximity on the social graph for friend recommendation. In this paper, we present MatchMaker, a collaborative filtering friend recommendation system based on personality matching. The goal of MatchMaker is to leverage the social information and mutual understanding among people in existing social network connections, and produce friend recommendations based on rich contextual data from people's physical world interactions. MatchMaker allows users' network to match them with similar TV characters, and uses relationships in the TV programs as parallel comparison matrix to suggest to the users friends that have been voted to suit their personality the best. The system's ranking schema allows progressive improvement on the personality matching consensus and more diverse branching of users' social network connections. Lastly, our user study shows that the application can also induce more TV content consumption by driving users' curiosity in the ranking process.**

Keywords—Collaboratie filterin; Friend recommendation; Social network; Reality projection; Social TV.

I. INTRODUCTION

Online recommendation systems based upon collaborative filtering have a long history since early 1990's, ranging from applications for music suggestions [6][7] to platforms that promote new forms of online employment such as Amazon mechanical turk [1]. In recent years, with the proliferation of online social network websites such as Facebook, research projects and commercial tools that aim to encourage more TV viewing through recommendations from one's online social network have boomed. One premise for such TV viewing recommendation system to be effective is that the connections in the social networks are strong and therefore influential. Indeed the strength of connection was one of the crucial drivers for the viral growth of social network websites such as Facebook, at their beginning stages [5]. As these social networks expand, however, the connections are becoming increasingly weaker, which, in effect, reduces the influence of the recommendation in TV viewing applications. MatchMaker aims to tackle this problem by going in the reverse direction: in order to encourage more TV viewing, instead of recommending a user to watch the shows that his social network friends have watched, MatchMaker recommends him to become friends with someone whose matching TV character is friend with the user's matching TV character. If the user has not already watched the TV show, he is likely to be curious in finding out what kind of potential personality or characteristics the recommended friend has, through the TV show. Figure 1 depicts the relationship schema in a more visual way.

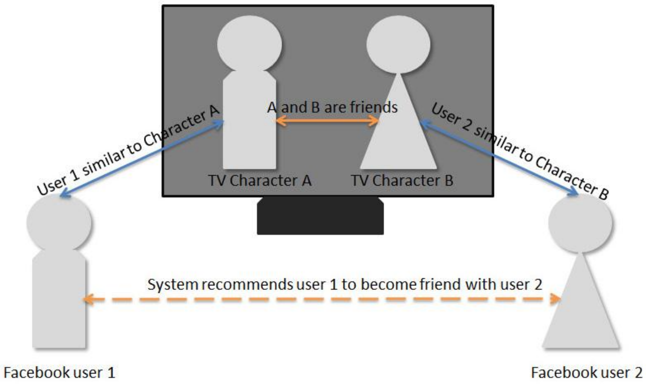

A and B are friends
User 1 similar to Character A
User 2 similar to Character B
TV Character A
TV Character B
System recommends user 1 to become friend with user 2
Facebook user 1
Facebook user 2

Figure 1. The MatchMaker system matches Facebook user 1 to TV character A and Facebook user 2 to TV character B based on their profile and social network voting, then suggests user 1 to become friends with user 2 if character A and character B are friends in the TV show.

In the following, we discuss in detail the design process as well as the implementation process of the first prototype. We will give realistic examples to demonstrate the current capability of the prototype. A short user study was conducted after the implementation of the application. We will share with readers our findings and insights from the user study. Finally, we will end with plans for future work on this project.

II. INTERACTION DESIGN AND FEATURES

In this section, we discuss the rationale behind the design decisions made through this project; we also go through the features of the current prototype. The Design Rationale part will take us through the state of art in this field and the potential advantage of MatchMaker's algorithm compared with existing algorithms used in online social media such as Facebook.

A. Design Rationale

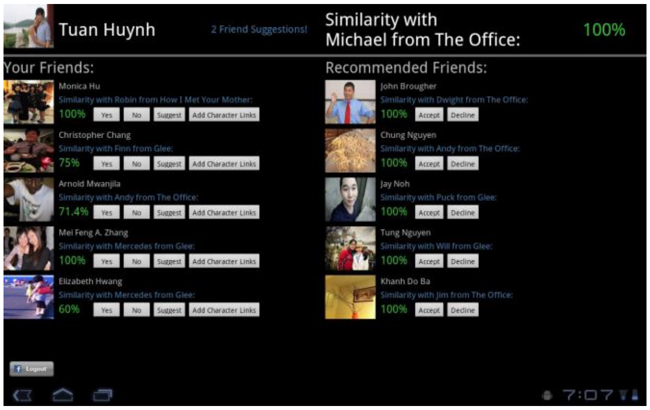
MatchMaker recommends friends to Facebook users based upon the TV characters they have been matched with. For example, if Facebook user 1 is similar to TV character A, Facebook user 2 is similar to TV character B, and character A and character B are friends in the same TV show, then the MatchMaker system recommends user 1 to become friends with user 2, if user 1 and user 2 are not already friends on Facebook. In order to calculate how similar a Facebook user is to a TV character, there are many different approaches. One approach that we initially came up was crawl the user's online profile data and compare that against a TV character's online data, such as that in International Movie Database (IMDB) or Wikipedia. However, going through many user profiles we have found that most Facebook users maintain relatively minimal profile information and there are no organized, consistent TV character profiles on IMDB or Wikipedia, either. At the same time, calculating the similarity through pure machine algorithmic techniques such as "keyword matching" using Natural Language Processing seems to be leaving out a lot of contextual information intrinsic to the social network and does not easily allow serendipitous discovery and scalable connections [3]. As a result, we decided to allow a user's 1st degree friends on Facebook to suggest and vote for characters who they believe the user is similar to. The system keeps track of the number of votes for each character that the user has been matched to, and ranks the characters in decreasing similarity order. With the same voting schema, the system asks Facebook users to add relationships among TV characters for the TV shows they have already watched. Later, when the system identifies a potential connection between two users in parallel with a TV characters' relationship, it recommends the two users to add each other as friends on Facebook.

Allowing a user's 1st degree Facebook friends to vote for his or her similar TV characters opens the door for a lot of contextual data outside of the online social media. For instance, a user might be voted to be similar to a character due to his or her looks, which, if taking a pure algorithmic approach, imposes heavy computational tasks such as image processing. A user might also be voted to be similar to a character based upon his or her personality or other features that his or her social network friends have come to know through real life interactions. Insights as such are very subjective, require a lot of common sensing judgment and are difficult to leave for machine algorithms to extract. The relationships among TV characters, however, are objective information. Since there is no good online TV character profile database, we let the users to populate the relationships into our system's database. At the same time, to ensure the accuracy of these data, we use the network ranking system to authenticate the relationships with the highest number of votes.

MatchMaker's friend recommendation system is easily compared with Facebook's existing friend recommendation system, "People you may know." While "People you may know" recommends a friend to a user based upon the number of their mutual friends, work and education information, MatchMaker recommends a friend to a user based upon the matching in personality and characteristics that their social network friends—and TV show story writers--have collectively concluded. In short, Facebook uses proximity matching whereas MatchMaker uses personality matching for friend recommendation. Intuitively, a matching personality evokes higher probability of a sustainable relationship, a technique that dating websites have been using for years. A few recent commercial platforms have been exploring various connection mapping methods, such as interest graph [8] and taste graph [9], to overcome the limitation of proximity matching given the existing social graph. In User Study section, we shall see some feedback from the users on comparing the two methods: proximity matching vs. personality matching.

B. Feature Overview

The current MatchMaker application allows a user to navigate through the following interactive stages: the Home Screen, User Info Dialog, Character Suggestion Dialog, Character Link Dialog, and the Friend Suggestion Dialog. The Home Screen, shown in Figure 2, has two parts, the left column for the signed-in user to suggest similar characters to his existing friends, and the right column showing recommended friends to the user. On the left column, given a particular friend of the user, if there is already a suggested character for this friend, the user can vote "Yes" or "No" to increase or decrease the ranking of this suggestion. If the user has another character in mind for his friend, however, he can click on "Suggest" to enter the show and the character who he believes is most similar with his friend. The user enters this information in Character Suggestion Dialog, as shown in Figure 3.


Tuan Huynh
2 Friend Suggestions!
Similarity with
Michael from The Office: 100%

Your Friends:

Monica Hu
Similarity with Robin from How I Met Your Mother:
100%

Christopher Chang
Similarity with Finn from Glee:
75%

Arnold Mwanjila
Similarity with Andy from The Office:
71.4%

Mei Feng A. Zhang
Similarity with Mercedes from Glee:
100%

Elizabeth Hwang
Similarity with Mercedes from Glee:
60%

Recommended Friends:

John Brougher
Similarity with Dwight from The Office:
100%

Chung Nguyen
Similarity with Andy from The Office:
100%

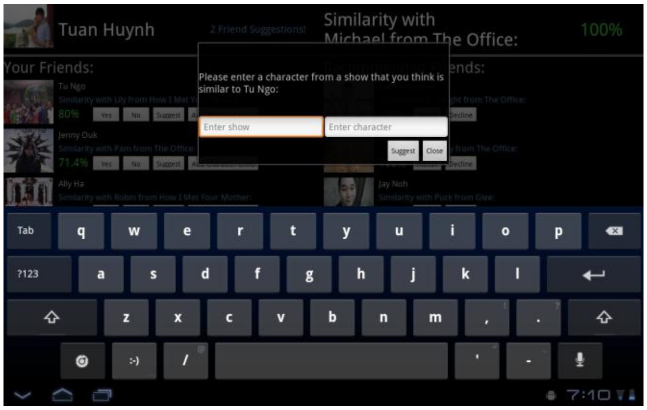
Jay Noh
Similarity with Puck from Glee:
100%

Tung Nguyen
Similarity with Will from Glee:
100%

Khanh Do Ba
Similarity with Jim from The Office:
100%

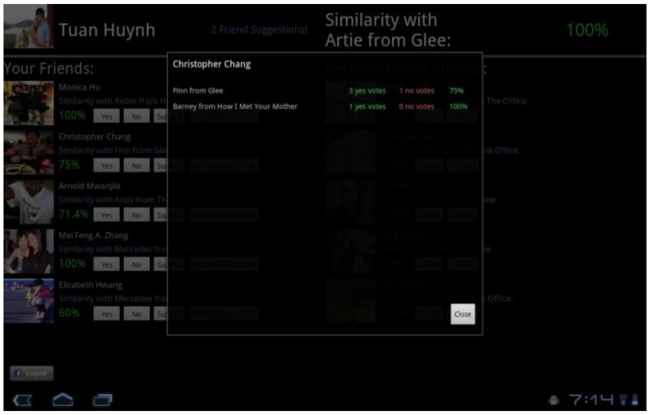
Figure 2. The Home Screen.

When the user clicks on a friend's profile picture, he gets to see the list of suggestions the network has made on which characters the friend is similar to and the respective voting percentages (Figure 4). A percentage is calculated by dividing the number of “Yes” votes over the total number of votes.


Tuan Huynh
2 Friend Suggestions!
Similarity with Michael from The Office:
100%
Your Friends:
Tu Ngo
80%
Jenny Ouk
Similarity with Pam from The Office:
71.4%
Ally Ha
Similarity with Robin from How I Met Your Mother:
Jay Noh
Similarity with Puck from Glee:
Yes
No
Suggest
Decline
Enter show
Enter character
Suggest
Close
7:10

Please enter a character from a show that you think is similar to Tu Ngo:

Figure 3. Character Suggestion Dialog.

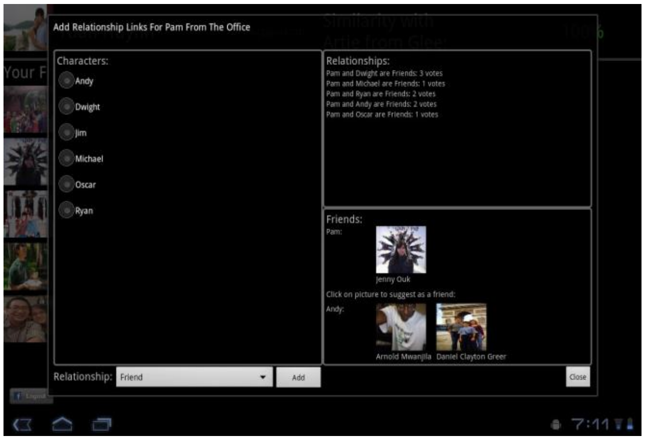


Christopher Chang

Finn from Glee	3 yes votes	1 no votes	75%
Barney from How I Met Your Mother	1 yes votes	0 no votes	100%

Figure 4. User Info Dialog.

“Add Character Links” button allows Facebook users to collectively populate and authenticate the relationships among TV characters, as shown in Figure 5.



Relationships:
Pam and Dwight are Friends: 3 votes
Pam and Michael are Friends: 1 votes
Pam and Ryan are Friends: 2 votes
Pam and Andy are Friends: 2 votes
Pam and Oscar are Friends: 1 votes

Click on picture to suggest as a friend:

Figure 5. Character Link Dialog.

To give users the incentive to populate TV character relationships into our database, the Character Link Dialog interface also provides matching friend profiles every time a new relationship link is created. For example, upon adding the relationship “Pam and Andy are friends” in The Office, the user instantly sees friend profiles matching to Andy, Arnold Mwanjila and Daniel Clayton Greer, showing up. Since the user has come to this dialog by clicking on “Add Character Links” button under the profile of his friend Jenny Ouk, he now sees that Arnold Mwanjila and Daniel Clayton Greer should be friends with Jenny Ouk. If they are not already connected on Facebook, the user can simply click on Arnold Mwanjila’s and Daniel Clayton Greer’s profiles to recommend them to Jenny Ouk. Next time when Jenny Ouk signs into her profile, she will see the two friend suggestions from the user.

The right column on the Home Screen (Figure 2) lists friends recommended by the system, based on the character similarity matching and character relationship links. The user can choose to accept or decline each recommendation. Upon clicking on “Accept”, the user will be directed to a Facebook page where he can send a friend request to the recommended person. One important feature worth noting is that all the links on the Home Screen, “Similarity with CharacterXYZ from Show123”, lead to a search on YouTube to allow the user to watch the video clips of the characters that are similar to his friends or recommended friends. This feature is especially important in the case of allowing the user to know more about the recommended friends. At present, Facebook’s “People you may know” allows a user to view the mutual friends between him and the recommended friends with proximity matching. Although “People you may know” also allows the user to view the profile of the recommended friends, our research on Facebook profiles has shown that the profile information is usually kept at a minimal level and gives not much information on the actual personality and characteristics of the recommended friends. Therefore, by allowing the user to view video clips of a character whose personality and characteristics have been voted to be similar with a recommended friend, MatchMaker provides the user with more contextual information and subsequently boosts the user’s confidence in accepting the recommended friend.

III. TECHNICAL IMPLEMENTATION

The current prototype of MatchMaker has been developed on Android 3.0 platform [2] on Motorola tablet Xoom. As a result, the layout has been customized to look nice on the tablet. Any smartphones or other mobile devices with Android OS can run the application, although the layout might not look as nice. Additionally, the user profile data and usage history has been saved locally on the tablet, which means the users cannot download the application and share with one another in a simple way yet. We plan to move the database to a server soon and adopt a client-server architecture for the implementation.

There are five major steps in the implementation process. In the following, we describe the five steps in detail.

A. Home Activity Creation

The Home Screen is defined by the HomeActivity class, which extends the Activity class provided by Android. Note that Android defines an activity as a screen that the user sees, so MatchMaker only has one activity. To set the home activity as the default activity to launch when the user opens MatchMaker, the AndroidManifest.xml file includes the following lines:

```
<activity
        android:name=".HomeActivity"
        android:label="@string/app_name"
        android:theme="@android:style/Theme.Black.
NoTitleBar">
<intent-filter>
<action android:name="android.intent.action.MAIN" />
        <category
        android:name="android.intent.category.LAUN
        CHER" />
</intent-filter>
</activity>
```

The category "android.intent.category.LAUNCHER" specifies that the HomeActivity is the activity to run when the user starts MatchMaker. Once the activity starts, the method onCreate() is called. It is in this method that we initialized our data structures, database helpers, and automatic UI event handlers.

We also set our application layout in this method by calling setContentView() on the main.xml file, which defines the contents of the layout. The Android framework takes care of initializing all the necessary view objects such as buttons, layout containers, text.

B. Facebook Authentication and User Data Retrieval

MatchMaker has three classes that take care of Facebook authentication: SessionStore, SessionEvents, and LoginButton. These three classes were extracted from a Facebook example that was provided with the Facebook Application Programming Interface (API) for Android. The SessionStore class saves and clears session information so that the user does not have to sign in more than once if the session has not yet expired. The SessionEvents class executes events during signing in and signing out process. The LoginButton class handles both signing in and signing out. In the onCreate() method, SessionStore first tries to restore a valid session. If there is no valid session, LoginButton displays a login picture. If there is a valid session, LoginButton displays a sign-out picture instead. If the sign-in picture is visible and the user clicks on the button, the Facebook class calls the authorize() method to requests a new session, which SessionStore ultimately saves. If the sign-out picture is visible and the user clicks on the button, the application clears the current user data, and SessionStore deletes the session.

The Facebook API provides a class called AsyncFacebookRunner that does asynchronous requests to the Facebook server. If a session turns out to be valid, onCreate() calls initUserData() which adds user specific content to the initial empty layout. initUserData() is also called when a user signs in successfully. The source code of initUserDate() is below:

```
private void initUserData() {
        mProgressDialog =
        ProgressDialog.show(HomeActivity.this, "",
        "Loading...");
        mAsyncFacebookRunner.request("me", new
        UserInfoRequestListener());
}
```

We can see that the AsyncFacebookRunner does a request for information about the signed-in user using the Facebook Graph API path "me". The UserInfoRequestListener class is a callback that continues with loading user data into the application if the request was successful. Another Facebook Graph API path is "me/friends" which is also used in MatchMaker to retrieve information about the signed-in user's friends.

C. Pop-up Dialogs

The overview of MatchMaker in section II shows several pop-up dialogs. These dialogs are not created during onCreate() but are instead created dynamically during user interaction. That is, these dialogs are created immediately before they are shown for the first time to the user. Once they are created, they are kept in memory. The Activity class provides two convenient methods for creating and customizing dialogs. The first method is onCreateDialog() which is called only once during dialog creation. Thus, onCreateDialog() initializes the necessary data structures and layout of the dialog. The second method is onPrepareDialog() is called every time before a dialog is shown. This allows MatchMaker to prepare dialog to show specific information regarding what the user clicked on. For example, the user clicks on the "Suggest" button for a certain friend. The ID of this friend is passed to onPrepareDialog() so that the character suggestion dialog knows what friend the user is suggesting for. Each dialog has a unique ID defined by MatchMaker. By passing these IDs into the methods showDialog() and dismissDialog(), MatchMaker can easily choose what dialog to show.

D. Database

The database tables are created via source code instead of the execution of raw SQL queries external to MatchMaker. During the initialization of DbAdapter classes in the onCreate() method, tables are created if they do not already exist. These DbAdapter classes also provide convenient methods to retrieve, update, insert, and delete table entries. In fact, the essential feature of MatchMaker, recommendation of friends based on character profiling, is done entirely by the database which has saved user inputs of character suggestions and relationship links. The schema of the most important database table, named votes, is shown below:

User ID – A string used to identify the Facebook user. This ID is the same ID that Facebook uses to identify its users.
Show – The name of a show.
Character – The name of a character.
Yes count – The number of yes votes.
No count – The number of no votes.

A row in this table means that a user is profiled to some character from some show with some number of yes votes and some number of no votes.

E. *Video Display*

YouTube results are shown via a browser that comes with the device. This process is relatively straight forward. As an example, MatchMaker first parses "Similarity with Finn from Glee" to just "Finn+from+Glee." This new string can just be appended to a standard YouTube URL query [10]. Then all MatchMaker needs to do is to start the browser activity with the URL. Below is the source code after parsing the "Similarity ..." text:

```
Uri uri = Uri.parse("http://m.youtube.com/index?desktop_uri=%2F&gl=US#/results?q=" + query);
Intent intent = new Intent(Intent.ACTION_VIEW, uri);
startActivity(intent);
```

Once the user closes the browser, MatchMaker is comes into focus again.

IV. USER STUDY

We conducted a survey with 17 users after they have tested the MatchMaker application. The goal of the survey is to find out what users think about the MatchMaker interface and its personality matching technique. The survey asked users questions based on a five-point scale and the questions are divided into roughly three categories: the users' existing usage of Facebook's friend recommendation system—"People you may know", the users' habit of watching TV, and the users' feedback on using MatchMaker compared with using Facebook friend recommendation system.

The survey found that Facebook users usually do not add friends from Facebook's "People you may know", with 58.8% saying that they never do and 35.3% saying that they occasionally do. When they do add someone from "People you may know", 23.5% indicated high dependence on the number of mutual friends, 23.5% indicated some dependence and another 23.5% indicated no dependence, with the rest of the users in between the spectrum. Due to this equally-spread distribution, it is hard to tell whether the number of mutual friends alone has any significant impact on the users' decision making process. Among the 17 users, the majority also stated that they watched very little TV. However, this might be due to the limitation on the user study participant selection, since all of them are undergraduate students at MIT with intense course work.

When asked about the likelihood of adding friends from MatchMaker's recommendation, 29.4% of the users said they would never do, 23.5% said sometimes, and 41.2% fell in-between never do and sometimes. Although this feedback is not as positive as we had hoped, it does give relatively higher probability of users adding recommended friends compared with that in Facebook. Indeed, one of the suggestions from 64.7% of the users was to combine the personality matching and proximity matching techniques to give even more context for the recommendations.

One of the goals of MatchMaker, besides recommending friends in a more contextual manner, is to encourage more TV content viewing. In the survey, we asked the users how likely they would watch the TV shows which had characters similar to the friends recommended to them. Over three quarters of the users indicated that they would watch the shows both before and after they had added the friends, in order to get a better understanding of the friends' possible personality and characteristics. The users also seemed to be satisfied with viewing the TV content through YouTube. As the users also needed to suggest similar characters to their existing network friends, we asked them how often their suggestions were based upon the friends' personality vs. appearance. While 47.1% of the users indicated both, 35.2% focused on personality and 17.6% focused on appearance. Lastly, although some people have compared MatchMaker with various dating websites, through the survey we found out that 41.1% users preferred using it for adding new friends, 35.3% were neutral and 23.5% preferred using it for finding dates.

In short, this survey has provided us with a few helpful insights. It confirmed our initial hypothesis that Facebook's current friend recommendation system alone, "People you may know", does not have significant impact on users' decision making process in adding new friends. It shines light on a promising future usage of personality-based friend recommendation system, but also leaves us with space of improvement on such system, i.e. adding the proximity matching on top of the personality matching.

V. CONCLUSION

The exponential growth of online social networking platforms such as Facebook has captivated our attention in recent years. As a result, the emerging field of NIT (Network and Information Ecology) has brought many new research efforts into the study of social graphs. As we dive deeper and start utilizing the social graphs in more and more applications that benefits from collaborative filtering, we realize, however, that the social graphs are not always a good model for matching data and drawing connections. One of the shortcomings of existing social graphs is that its proximity matching schema does not necessarily provide enough context. MatchMaker is an attempt to address this problem by trying out a different approach: personality matching for friend recommendation. We designed and implemented the MatchMaker prototype on Android tablet, and had some users test it in order to draw feedback for further improvement. The feedback from users has suggested that personality matching does provide the users with more contextual information about recommended friends, comparing with proximity matching. However, it also suggests that a combination of personality matching and proximity matching will work even better in terms of giving the users more information and confidence to add a new friend online.

VI. FUTURE WORK

In the next iteration of the prototype, we will combine both personality matching and proximity matching in the MatchMaker application. We also plan to adopt client-server architecture for future implementation. In addition, we hope to conduct the next round user study on a larger scale and with participants of more diverse background.

ACKNOWLEDGMENT

We thank Tuan Huynh for his contribution in the implementation process of the application, Marie-Jose Montpetit for her insightful advice. We also thank all the user study participants for their kind feedback.

REFERENCES

[1] Amazon Mechanical Turk. https://www.mturk.com/mturk/welcome

[2] Android Developer's Guide. http://developer.android.com/guide/index.html

[3] E. Chang, Scalable Collaborative Filtering for Mining Social Networks, Neural Information Processing Systems, December 2008.

[4] Facebook Graph API Reference http://developers.facebook.com/docs/reference/api/

[5] D. Kirkpatrick, The Facebook Effect, Simon & Schuster Adult Publishing Group, June 2010.

[6] M. Metral, MotorMouth: A Generic Engine for Large-Scale, Real-Time Automated Collaborative Filtering. Master Thesis, Program in Media Arts and Sciences, Massachusetts Institutue of Technology, June 1995.

[7] U. Shardanand, Social Information Filtering for Music Recommendation, Master's Thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, June 1994.

[8] The Interest Graph. http://www.theinterestgraph.org/

[9] Taste Graph, Hunch. http://blog.hunch.com/?p=47384

[10] YouTube API Documentation. http://code.google.com/apis/youtube/overview.html