In-SYNC:Interest Aligned Recommender System

Abhishek Kumar
Grad Student,Software Engineering
Arizona State University
Tempe,AZ,USA
akuma168@asu.edu

Aman Maheshwari
Grad Student,Software Engineering
Arizona State University
Tempe,AZ,USA
amahes10@asu.edu

Saheb Singh Johar
Grad Student,Software Engineering
Arizona State University
Tempe,AZ,USA
ssjohar@asu.edu

Shubhangi Gupta
Grad Student,Software Engineering
Arizona State University
Tempe,AZ,USA
sgupt170@asu.edu

***Abstract*— Social influence plays a significant role in deciding our selection when it comes to trying out new things. A river of "data of opinions/reviews" is flowing over the internet which can be useful to personalize and predict a person's taste of likings and disliking. However, the problem is, it is not often considered by the traditional recommender systems.**

These recommender systems operate using the popular ratings and reviews and doesn't consider the fact that a person's social influence can greatly determine what product is he/she going to try next.

The new Web based app, In-SYNC recommender system will utilize information from social networks, including user's likings, product's rating, and influence from social friends and thus, will recommend the product accordingly.

Keywords—Social Influence, Recommender System, In-SYNC, opinions, reviews, Traditional Recommender System

I. INTRODUCTION

In today's world, when most of the people are connected via internet devices with their friends and families, they never fail to ask an opinion of them while trying out a new restaurant, deciding which movie to watch on weekends, grabbing a new book for a read.

Social influence plays a significant role in deciding our selection when it comes to trying out new things. Somehow, a trend has psychologically embedded that if a new restaurant, book, movie, etc. has worked out good for my friend, I would also find it good whenever I give it a try and vice-versa. And after trying out the restaurant or a movie, which our friend mentioned, we also don't stop sharing our opinions and reviews about the same to our friends, who haven't tried out the same yet.

A river of "data of opinions/reviews" is flowing over the internet which can be useful to personalize and predict a person's taste of likings and disliking.

We see that how social influence plays a huge part in product marketing.

However, the problem is, it is not often considered by the traditional recommender systems. These recommender systems operate using the popular ratings and reviews and doesn't consider the fact that a person's social influence can greatly determine what product is he/she going to try next.

As a solution, we come up with our semantic web based recommender system," In-SYNC", which is going to take social network into consideration before recommending a new product to the person.

II. GOAL

In-SYNC recommender system will utilize information from social networks, including user's likings, product's rating, and influence from social friends.

A predictive model is developed to make personalized recommendations from such information. We extract data from a real online social network, and we see a trend in this large dataset that friends tend to select the same products and give similar reviews.

Therefore, In-SYNC will recommend a new product based on the person's social network and incase, that new product hasn't been tried at a large scale by the person's network, he will have the option to select the product from the popular-product list; the popular-product list would be a function of general anonymous reviews.

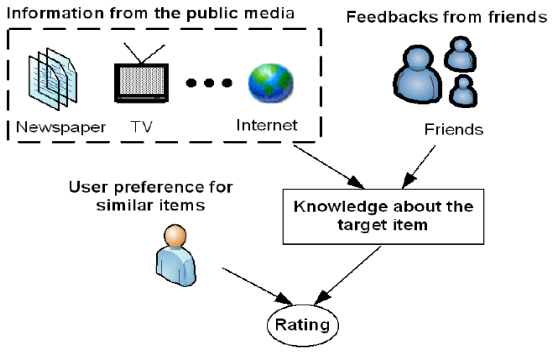


Figure 3.1: The three factors that influence a customer's buying decision: user preference for similar items, information regarding the target item from the public media, and feedbacks from friends.

The three factors that largely influence a customer's buying decision are:

- User Preference for similar items
- Information regarding the target item from the public media
- Feedbacks from friends

II. RELATED WORK IN THE DOMAIN

In doing research for the project, we found that the first factor in determining a product for consumption is person's own preference for it. If someone does not like Italian cuisine, he would be less likely to pick something like "pizza" to begin with.

The second factor is the public review of the products. If these received terrible reviews, the person would most likely lose interest and stop any further investigation. Finally, it is the recommendation from the friend, that makes one finally select a item like "Chinese cuisine". If we recall the decisions that we make in our daily life, such as finding restaurants, buying a house, and looking for jobs, many of them are influenced by these three factors.

Intuitively, a customer's buying decision or rating is decided by both his/her own preference for similar items and his/her knowledge about the characteristics of the target item.

A Knowledge about the target item can be obtained from public media such as magazines, television, and the Internet. Meanwhile, the feedbacks from friends are another source of knowledge regarding the item, and they are often more trustworthy than advertisements.

When a user starts considering the feedbacks from his/her friends, he/she is then influenced by his/her friends. With such an understanding in mind, we are going to propose an interest-based recommender system (In-SYNC).

However, on evaluating the related work done in this domain ,we found that most of the apps doesn't consider the social influence of the users and thus, misses out a great opportunity to tap-in the correct interests of the customers.

Also, a survey did find that as compared to traditional recommender system, apps like In-Sync will get 17% more accurate results, which is a big margin which the companies can't afford to lose.

III. SURVEY OF TRADITIONAL RECOMMENDER SYSTEMS

A number of traditional apps are available that try to recommend the products to the users on the content based model approach which unfortunately doesn't consider the integration of social network into the recommender system.

- Identify applicable funding agency here. If none, delete this text box.

In-SYNC integrates the social network of the users and uses the same for recommending the products to the potential customers.

Some of the popular traditional recommender systems are:

A. *IMDb*

IMDb, the world's most popular and authoritative source for movie, TV and celebrity content.

B. *Spotify*

Spotify is a digital music service that gives you access to millions of songs.

C. *Yelp*

User Reviews and Recommendations of Top Restaurants, Shopping, Nightlife, Entertainment, Services

D. *Amazon*

Amazon is an ecommerce company that suggests online products to its users.

E. *RottenTomatoes*

Rotten Tomatoes, home of the Tomatometer, is the most trusted measurement of quality for Movies & TV.

IV. SEMANTIC DATA MODEL OF IN-SYNC

The application In-SYNC is a Semantic Web application which operates on a semantic data model. This model facilitates the functionalities of the application and provides a detailed view about how the datasets are consumed and linked by the application.

The basic idea of the application is to provide recommendation to our users based on their interest alignment with their friend and a generalized recommendation based on popularity of items in the respective dataset/domain. The application working model is as follows:

Our application allows our users to login into our application via Facebook, where the application fetches the details of the user interests such as the places he has been to, the movies he has watched, the music he likes, the authors he follow, etc. We also get a list of users who are friends with our main user. Based on these interests and linking of users, we recommend our user about the products they are interested in.

We first map our users and find out who all are linked as friends. Once done, we keep it in one set (RDF turtle file) along with his own interests. When the user tries to get a recommendation of a item (such as movies of genre action), firstly we fetch the details of his/her friends (movies watched by his/her friends). Based on the data we have gathered, we filter our data based on the conditions provided by the user

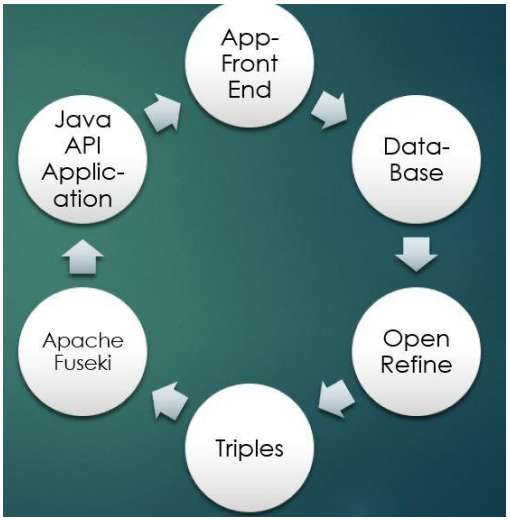
(such as Genre must be action for the movies he/she wants). Once we get the desired output, we match the count of the recommended items with our threshold value of the items which will be recommended to the user.

If the count is less than the threshold, then we fetch the data from a particular domain(dataset) on the basis of the same condition provided by the user earlier (finding movies in movies dataset for Genre Action) and the popularity of the items on general basis. Once we have gathered our data, we collaborate both the data and display it in our application to the user as his required recommendations aligned with his/her friends interests and a general view of the audience.

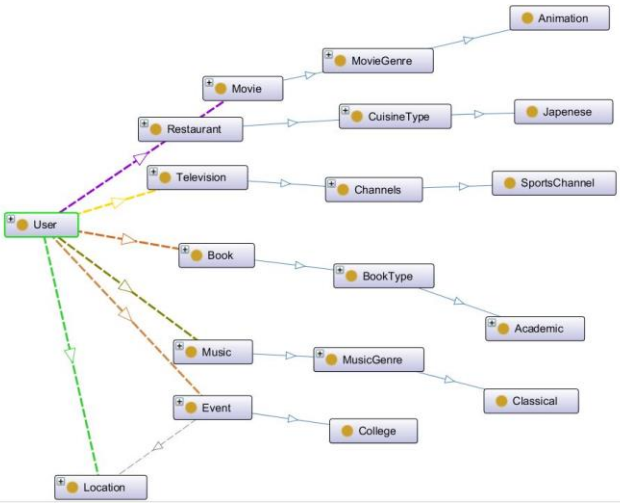
The Software used are:

- Apache Jena Fuseki
- SPARQL
- Google Refine
- Java MVC
- PHP
- MySQL

Below Figure illustrates the flow of the semantic data model:



V. ONTOLOGY USED



The Figure above depicts the brief obverview of the ontology being used in In-SYNC app.

VI. CHALLENGES FACED IN INTEGRATING THE DATA

Large Datasets were used from various sources such as

- Spotify
- IMDb
- Facebook
- Yelp
- Kaggle

Some of the challenges faced while integrating the data were:

- While integrating FUSEKI with java application, we faced a challenge of FUSEKI getting down after certain number of queries being hit (specifically 5). The same error was persisting for all the API's with different version of FUSEKI as well.
- Since the users are main entities of In-SYNC application, it was required to have large datasets linked with users show casing their interests on different products.
  As this data was not easily available, we had to manually create the same which took lots of efforts and time.
- And whatever existing users were present, those weren't mapped. So, we had to map them, on the basis of their ids ,with different fields.

VII. QUERYING THE LINKED DATA

We have used SPARQL to query our datasets and vet the desired output. The API's developed in java hit the query endpoint provided by the datasets uploaded on the FUSEKI

server and sends the SPARQL query to the server. The query gets executed on the FUSEKI and the result is returned back to the API's in form of JSON. Let us understand this based on an working API in the project. The user wants to get recommendation for movies based on a certain Genre 'Action'. Let us see the working scenario of the application and the queries involved in it.

Step1: Finding the friends associated with the current user.
Query involved:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX InSync:
<http://www.semanticweb.org/abhishek/ontologies/2017/9/insync#>

SELECT ?person ?friend
WHERE {
        ?person InSync:hasFriends ?friend .
        FILTER (?person = <http://127.0.0.1:3333/0>)
}
```

The above query provides all the resources of the friends for the current user.

Step 2: Finding the movies of all the friends received from the above query.
Query Involved:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX InSync:
<http://www.semanticweb.org/abhishek/ontologies/2017/9/insync#>

SELECT ?person ?movies
WHERE {
        ?person InSync:watchesMovie ?movies .
        ?movies InSync:hasGenre ?Genre
        FILTER (?person in (<http://127.0.0.1:3333/68>, <
<http://127.0.0.1:3333/578>, <http://127.0.0.1:3333/758>) &&
contains(str(?Genre),'Action') }
```

Step 3: Finding the movies from the movie dataset if the movie count is less than the threshold value.
Query Involved:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX InSync:
<http://www.semanticweb.org/abhishek/ontologies/2017/9/ins
ync#>

SELECT ?movie ?Genre ?name ?popular
WHERE {
        ?movie InSync:hasName ?name.
        ?movie InSync:hasGenre ?Genre .
        ?movie InSync:hasPopularity ?popular .
  FILTER(contains(str(?Genre),'Action') && ?name not in
('Tangled', 'Monsters University', 'Cars 2'))
}
ORDER BY DESC(?popular)
LIMIT 10
```

VIII. KEY FUNCTIONALITY OF IN-SYNC

The key functionality of the application is to provide recommendation of an item based on the alignment of interest of a person with his/her friends. The application focuses on providing the recommendation by prioritizing the user's interest alignment first and then getting the general recommendation from the general views. The application focuses on providing better recommendation based on social media influence and interest alignment between the friends of the user. Also, although the IN-SYNC is using the Facebook Login for the authentication purpose ,the data which includes the interests of users are being picked up from the triples used by In-SYNC rather than that from Facebook.

IX. FUTURE WORK

- Currently, the Triples are being created by manually uploading the JSON Files to Open Refine. A step further would be automating this by running a script or by using some API provided by Open Refine or some other Tool.

- Besides Restaurants, Movies, Books we can have In-SYNC extending its reach into domains of other required interests such as Buying a property, Getting the best Doctors and Health-Clinics in Town.

ACKNOWLEDGMENT *(Heading 5)*

We would like to thank our Professor Dr. Srividya Bansal and, Mr. Jaydeep Chakraborty for their unconditional support and guidance throughout the project.

REFERENCES

[1] R. Zheng, F. Provost and A. Ghose. Social Network Collaborative Filtering: Prelimi-nary Results. In Proceedings of the Sixth Workshop on eBusiness (WEB2007), De-cember 2007.

[2] D. Billsus and M. Pazzani. Learning Collaborative Information Filters. In Proceedings of International Conference on Machine Learning, 1998.

[3] P. Bonhard and M. A. Sasse. "Knowing me, knowing you" - Using Profiles and Social Networking to Improve Recommender Systems. BT Technology Journal, Vol 24 No 3, July 2006.

[4] J. S. Breese, D. Heckerman, and C. Kadie. Empirical Analysis of Predictive Algo-rithms for Collaborative Filtering. In Proceedings of the 14th Conference on Uncer-tainty in Artificial Intelligence, pp. 43-52, 1998.