Social Media Recommendation System – Movies

Nidhi Tattur Aravinda Kumar-(013845494)
CMPE256, San Jose State University

Abstract— Social media plays an important role in marketing any product in the recent times. In this project I present a new paradigm of recommender systems which can utilize information in social networks, including user preferences, item's general acceptance and influence from social friends. A probabilistic model is developed to make personalized recommendations from such information. I extract data from a real online social network, and my analysis of this large dataset reveals that a person's choice of movies depends on various factors like personal choice, ratings given by friends and friends tend to select the same items and give similar ratings.

Keywords—media recommendation, collaborative filtering, content-based filtering, hybrid recommendation system.

I. Introduction

Recommender Systems provide users with personalized recommendations on items such as movies, books, news and web pages. These systems predict user preferences (often represented as numeric ratings) for new items based on the user's past ratings on other items. There are typically two types of algorithms for recommender systems -- content-based filtering and collaborative filtering.

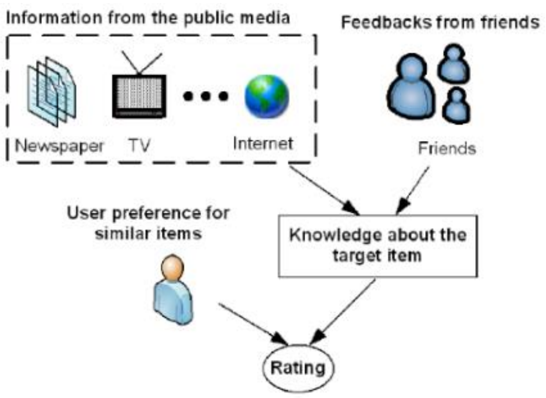
Content-based methods measure the similarity of the recommended item (target item) to the ones that a target user (i.e., user who receives recommendations) likes or dislikes on item attributes.

Collaborative filtering finds users with tastes that are like the target users based on their past ratings. Collaborative filtering will then make recommendations to the target user based on the opinions of those similar users.

On the other hand, Hybrid recommender systems combine two or more recommendation strategies in different ways to benefit from their complementary advantages.

II. Social Media BASED Recommender System

The figure below illustrates impact of three factors on customers' final buying decisions. 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 user's preference, such as interest in drama movies, is usually reflected from the user's past ratings to other similar items, e.g. the number of drama movies that user previously viewed and the average rating that user gave to those movies. 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. With the advent of social networks and accessibility of personal profile information, it is now possible to build recommendation engines that incorporate personal profiles. Aside from just information about individuals, social media also gather how they relate to each other in the social graph. Using these pieces of information, recommendation engines that can take advantage of the rich information available from social networks are built.



III. DATASET

The recommendation engine is built on data collected from websites like IMDb, bookmyshow and personalized recommendation websites like movielens.

```
credits.head()
```

	cast	crew	id
0	[{'cast_id': 14, 'character': 'Woody (voice)',...	[{'credit_id': '52fe4284c3a36847f8024f49', 'de...	862
1	[{'cast_id': 1, 'character': 'Alan Parrish', '...	[{'credit_id': '52fe44bfc3a36847f80a7cd1', 'de...	8844
2	[{'cast_id': 2, 'character': 'Max Goldman', 'c...	[{'credit_id': '52fe466a9251416c75077a89', 'de...	15602
3	[{'cast_id': 1, 'character': "Savannah 'Vannah...	[{'credit_id': '52fe44779251416c91011acb', 'de...	31357
4	[{'cast_id': 1, 'character': 'George Banks', '...	[{'credit_id': '52fe44959251416c75039ed7', 'de...	11862

The dataset consists of ratings and tag applications applied to movies by users. Users were selected at random for inclusion. All selected users had rated at least 20 movies. No demographic information included. Each user is represented by an id, and no other information is provided.

```
keywords.head()
```

	id	keywords
0	862	[{'id': 931, 'name': 'jealousy'}, {'id': 4290,...
1	8844	[{'id': 10090, 'name': 'board game'}, {'id': 1...
2	15602	[{'id': 1495, 'name': 'fishing'}, {'id': 12392...
3	31357	[{'id': 818, 'name': 'based on novel'}, {'id':...
4	11862	[{'id': 1009, 'name': 'baby'}, {'id': 1599, 'n...

The dataset has additional information about cast, crew, movie-id, keywords/tags for the movie, genre, average rating of the movie, number of votes by users.

```
ratings.head()
```

	userId	movieId	rating	timestamp
0	1	31	2.5	1260759144
1	1	1029	3.0	1260759179
2	1	1061	3.0	1260759182
3	1	1129	2.0	1260759185
4	1	1172	4.0	1260759205

Data Cleaning

Below are the data cleaning techniques involved:

Irrelevant data, Duplicates, Type Conversion, Syntax Errors, Transformation, Normalization and Missing Values are removed from data set.

IV. RECOMMENDATION ENGINES

1) Simple Recommendation System

The Simple Recommender offers generalized recommendations to every user based on movie popularity and (sometimes) genre. The basic idea behind this recommender is that movies that are more popular and more critically acclaimed will have a higher probability of being liked by the average audience. This model does not give personalized recommendations based on the user. Movies are sorted based on ratings and popularity and top movies of the list are displayed. We can also get the top movies of a genre.

```
build_chart('Romance').head(15)
```

	title	year	vote_count	vote_average	popularity	wr
10309	Dilwale Dulhania Le Jayenge	1995	661	9	34.457	8.565285
351	Forrest Gump	1994	8147	8	48.3072	7.971357
876	Vertigo	1958	1162	8	18.2082	7.811667
40251	Your Name.	2016	1030	8	34.461252	7.789489
883	Some Like It Hot	1959	835	8	11.8451	7.745154
1132	Cinema Paradiso	1988	834	8	14.177	7.744878
19901	Paperman	2012	734	8	7.19863	7.713951

2)Content Based Recommendation System

Content-Based recommender uses movie information and users' viewing profile. In a content-based method each user and each movie are uniquely characterized. The viewing history of a user for each movie is converted to an implicit rating. We consider feature sets, such as actor, director, keyword etc. that describe a movie. For each feature in a feature set, based on the user's past viewed movies and the user's rating for each movie, we compute a feature weight. Each feature weight is calculated separately for each user. If a user watched a movie completely or much of it, then, the features extracted from this movie are important, and their weights will be assigned accordingly. Once the user profile is generated, we calculate the similarity of the user profile with all the items in the dataset, which is calculated using cosine similarity between the user profile and item profile.

```
get_recommendations('The Dark Knight').head(10)

8031         The Dark Knight Rises
6218                 Batman Begins
6623                  The Prestige
2085                     Following
7648                     Inception
4145                      Insomnia
3381                       Memento
8613                  Interstellar
7659    Batman: Under the Red Hood
1134                Batman Returns
Name: title, dtype: object
```

Recommendations can be improvised by considering more attributes than just the description of the movie. Eg: Someone who liked The Dark Knight probably likes it more because of director and would hate Batman Forever and every other substandard movie in the Batman Franchise.

Advantages of Content Based approach is that data of other users is not required, and the recommender engine can recommend new items which are not rated currently, but the recommender algorithm doesn't recommend the items outside the category of items the user has rated.

```
improved_recommendations('The Dark Knight')
```

	title	vote_count	vote_average	year	wr
7648	Inception	14075	8	2010	7.917588
8613	Interstellar	11187	8	2014	7.897107
6623	The Prestige	4510	8	2006	7.758148
3381	Memento	4168	8	2000	7.740175
8031	The Dark Knight Rises	9263	7	2012	6.921448
6218	Batman Begins	7511	7	2005	6.904127
1532	The French Connection	435	7	1971	6.123458

3) Collaborative Recommendation System

The Content-Based engine that we built doesn't capture the personal tastes and biases of a user. Anyone querying the engine for recommendations based on a movie will receive the same recommendations for that movie, regardless of who (s)he is.

Therefore, we implemented Collaborative Filtering to make recommendations to Movie Watchers. Collaborative Filtering is based on the idea that users similar to a person can be used to predict how much the person will like a particular product or service those users have used/experienced, but the person has not.
User Based-CF uses the logic of recommending items by finding similar users to the *active user* (to whom we are trying to recommend a movie).
Item Based-CF focus on items from all the options that are more similar to each other and the person's likes.

```
ratings[ratings['userId'] == 1]
```

	userId	movieId	rating	timestamp
0	1	31	2.5	1260759144
1	1	1029	3.0	1260759179
2	1	1061	3.0	1260759182
3	1	1129	2.0	1260759185
4	1	1172	4.0	1260759205
5	1	1263	2.0	1260759151
6	1	1287	2.0	1260759187

4) Hybrid Recommendation System

Both content-based filtering and collaborative filtering have their strengths and weaknesses. Three specific problems can be distinguished for content-based filtering:

- *Content description*. In some domains generating a useful description of the content can be very difficult.
- *Over-specialization*. A content-based filtering system will not select items if the previous user behavior does not provide evidence for this.
- *Subjective domain problem*. Content-based filtering techniques have difficulty in distinguishing between subjective information such as points of views and humor.

A collaborative filtering system doesn't have these shortcomings. Because there is no need for a description of the items being recommended, the system can deal with any kind of information. Furthermore, the system is able to recommend items to the user which may have a very different content from what the user has indicated to be interested in before. Finally, because recommendations are based on the opinions of others it is well suited for subjective domains like art. However, collaborative filtering does introduce certain problems of its own:

- *Early rater problem*. Collaborative filtering systems cannot provide recommendations for new items since there are no user ratings on which to base a prediction.
- *Sparsity problem*. In many information domains the existing number of items exceeds the amount a person is able (and willing) to explore by far. This makes it hard to find items that are rated by enough people on which to base predictions.
- *Gray sheep*. Groups of users are needed with overlapping characteristics. Even if such groups exist, individuals who do not consistently agree or disagree with any group of people will receive inaccurate recommendations.

A system that combines content-based filtering and collaborative filtering could take advantage from both the representation of the content as well as the similarities among users. Although there are several ways in which to combine the two techniques a distinction can be made between two basis approaches. A hybrid approach combines the two types of information while it is also possible to use the recommendations of the two filtering techniques independently.

Tn this project, will try to build a simple hybrid recommender that brings together techniques I have implemented in the content based and collaborative filter-based engines. This is how it will work:
Input: User ID and the Title of a Movie
Output: Similar movies sorted on the basis of expected ratings by that particular user.

```
hybrid(1, 'Avatar')
```

	title	vote_count	vote_average	release_date	id	est
974	Aliens	3282.0	7.7	1986-07-18	679	3.343526
1011	The Terminator	4208.0	7.4	1984-10-26	218	3.165331
8658	X-Men: Days of Future Past	6155.0	7.5	2014-05-15	127585	3.095831
522	Terminator 2: Judgment Day	4274.0	7.7	1991-07-01	280	3.005560
8401	Star Trek Into Darkness	4479.0	7.4	2013-05-05	54138	2.845466
344	True Lies	1138.0	6.8	1994-07-14	36955	2.840542
2014	Fantastic Planet	140.0	7.6	1973-05-01	16306	2.820183

V. CONCLUSION

Social media provide an important source of information regarding users and their interactions. This is especially valuable to recommender systems. In this project I implemented a social network-based recommender system (SNRS) which makes recommendations by considering a user's own preference, an item's general acceptance and influence from friends. The influences from distant friends are also considered in an iterative classification. In addition, data was collected data from a real online social network. The analysis on this dataset reveals that friends tend to review the same movies and give similar ratings. We compared the performance of SNRS with other methods, such as collaborative filtering (CF), content filtering, hybrid filtering and achieved best results using hybrid model.

REFERENCES

[1] *Social Media Recommender Systems: Review and Open Research Issues. Anitha Anandhan, Liyana Shiub,Ghulam Muztaba 2169-3536*

[2] *Q. Gao, L. Xin, "Products Recommend Algorithm Based on Customer Preference Model and Affective Computing",* Proc. CCC'2010, *pp. 2891-2896, 2010.*

[3] *Introduction to Recommender Systems: Collaborative Filtering; http://blog.ethanrosenthal.com/2015/11/02/intro-to-collaborative_-filtering/*

[4] *MMDS course slides; Jeffrey Ullman; http://infolab.stanford.edu/ ullman/mmds/ch9.pdf*

[5] *Recommending items to more than a billion people; Kabiljo et al; https://code.facebook.com/posts/861999383875667/recommending-items-to-more-than-a-billion-people/*

[6] *Movie Recommendation https://github.com/rounakbanik/movies*

[7] *Social Media Movie Recommendation System https://github.com/jalajthanaki/Movie_recommendation_engine*