A Content Recommendation System Based on Category Correlations

Sang-Min Choi
Department of Computer Science
Yonsei University
Seoul, Republic of Korea
jerassi@yonsei.ac.kr

Yo-Sub Han
Department of Computer Science
Yonsei University
Seoul, Republic of Korea
emmous@yonsei.ac.kr

Abstract— **In Web 2.0, there is no clear distinction between users and webmasters. Many internet users are not only information consumers but also information providers. There are lots of information in the Web and most people can find what they want by searching the Web. One problem of the large number of data in the Web is that we often spend most of our time to find a correct result from search results. Thus, people start looking for a better system that can suggest relevant information instead of letting users go through all search results: We call such systems recommendation systems. A recommendation system is often based on collaborative filtering (CF). A traditional CF approach requires lots if user data so that a recommendation system can compute the similarity of user preferences and suggest items based on the computed preferences between users. This approach has two problems: sparsity and cold start. We propose a different CF approach based on the category correlation of contents. First, we show how to compute the category correlations of contents. Then, we design a new recommendation algorithm based on the category correlations to a user with certain preferences. Note that our approach does not require computing the preference similarity between users.**

Keywords- recommendation system; collaborative filtering; category correlation

I. INTRODUCTION

We search and gain lots of information from the Web since the late 20th century. One big difference from the Web and the traditional content provides is that we can search what we are interested from the Web. For instance, from a music magazine, it is not easy to find information about a particular song. On the other hand, in the music search site such as Yahoo! Music, we can find songs by simply typing in a song title [1]. We can also obtain information about books, shops or movies from the internet [2, 3, 4, 5, 6, 7]. However, huge amount of data do not always guarantee satisfactory outcome. Because of lots of spam data and wrong information on the Web, we often spend the most of our time to search for relevant information from search results by going through all of them. Namely, the accuracy and reliability of search results become very low.

In a recommendation system, users do not need to go through all search results. The system actively suggests items that are likely interested to users based on user information and, thus, eliminate the burden of looking at all results. The recommendation systems are often based on collaborative filtering [4, 5, 6]. Collaborative filtering is one of many ways to implement collective intelligence. A traditional CF is as follows: First, users provide preferences for a set of items. Then, the system identifies groups of users with similar preference based on a similarity measure of preferences. When the system suggests an item to a user X, it first determine the group that X belongs to and suggest relevant items based on the preferences of users in the group [2, 4, 5]. This approach works well when there are enough user preference data. In other words, it is difficult to suggest good items if there are not enough data to create user groups [8].

We design a different approach for suggesting items. Instead of computing user groups based on user preferences, we use category information on items. We first compute category correlations and, then, match the correlation with user preference. We apply the proposed method to the GroupLens database that consists of 3,883 movies with category information [9].

In Section II, we describe related work and some problems that we aim to tackle. We, then, propose a new technique in Section III for recommending items and implement the algorithm and show experiment results in Section IV. We conclude the paper in Section V.

II. REATED WORK

A. Collaborative Filtering based on User Preferences

A collaborative filtering based on user preferences is defined as follows with three phases: The first step is that the system selects a user X for recommendation and calculates the Pearson correlation coefficient of X with the other users using Equation (1).

$$\rho_{xy} = \frac{\mathrm{cov}(X,Y)}{\sigma_x \sigma_y} = \frac{\Sigma(X_i - \overline{X})(Y_i - \overline{Y})}{\sqrt{\Sigma(X_i - \overline{X})^2}\sqrt{\Sigma(Y_i - \overline{Y})^2}} \qquad (1)$$

In Equation (1), X is a user selected for recommendation, and $\overline{X}$ is a mean rating of user X. Then, σ_X is the standard deviation of rating of user X. X_i is the rating for the i^{th} item by user X. Let Y be the other users. The Pearson correlation coefficient is between -1 and 1. If the Pearson correlation between X and Y is closer to 1, then it means that two users

X and Y have similar preferences. If X and Y have opposite preferences, then it becomes close to -1 [2, 4, 5].

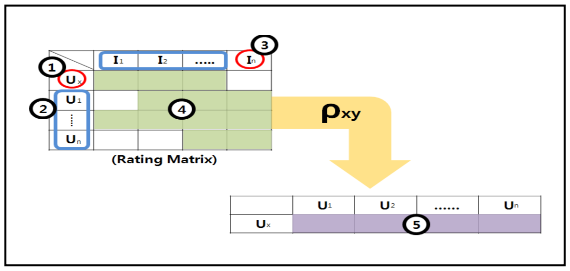

Figure 1. A course which calculates Pearson Correlation Coefficient

Figure 1 shows an example of the first step. ① is user X who receives recommendation and ② denotes all the other users. ③ is a set of items for X and ④ shows the preferences by other users in ②. Then, ⑤ is the Pearson correlation coefficient between user X and the other users.

The second step is to compose a neighbor group using ⑤ in Figure 1 Users who have more than a particular correlation coefficient are grouped as a neighbor.

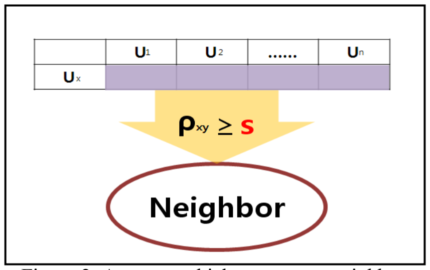

Figure 2. A course which compose a neighbor

Figure 2 shows the second step. As depicted in Figure 2, s is a threshold for determining users to be in the same group with user X.

The final step is to predict preference for specific item based on rating of a neighbor. It uses Equation (2) for collaborative filtering.

$$P = \overline{X} + \frac{\sum_{Y \in raters}(Y_n - \overline{Y})\rho_{xy}}{\sum_{Y \in raters}|\rho_{xy}|} \qquad (2)$$

$\overline{X}$ is a mean rating of a user X, and Y_n is an input rating by other users for the nth item. Then, $\overline{Y}$ is the mean rating by neighbor of user X for the current item. Finally, ρ_{xy} is the Pearson correlation coefficient between user X and the other users. The raters are a set of users who input rating for predicting item. Result P in Equation (2) is a predicted value on an item for user X.

B. Problems of Collaborative Filtering based on User Preferences

The collaborative filtering approach is based on user preference. It may cause several problems. First, there is a sparsity problem; it occurs where there are not enough inputs of user preference. If we recommend an item using a neighbor composed with a small amount of ratings, then the accuracy of recommendation is lower than using a neighbor composed with a large amount of ratings. The sparsity problem affects the accuracy of recommendation by collaborative filtering [4, 5]. The second problem is cold start. This problem occurs when new users or items are added. When new users or items are added, each one has no rating. General collaborative filtering composes a neighbor except for uses without rating. This implies that a user without rating cannot have a neighbor. In other words, the system cannot recommend an item to the new user who has no rating before user inputs more than a certain number of ratings. Thus, the system has to wait until all users rate enough number of items before recommending items.

III. Our Apporach

We propose a new recommendation system based on contents information. We compute correlations of contents category and recommend items based on the correlations between contents and users.

A. Database

We use GroupLens database, which is open in public [9]. Namely, our recommendation system suggests movies to users based on user preferences. The database is as follows:

TABLE I. Movie Database

Attribute	Meaning
MovieID	ID of each movie. Range between 1 and 3952
Title	Title of the movie
Genre	Genre of the movie

In MovieID, 69 ID of 3952 ID is empty. So, the number of all movies is 3,883.

We use genre as category of items. There are 18 genres of movies as follows:

TABLE II. Genre Database

No.	Genre	No.	Genre
G1	Action	G10	Film-Noir
G2	Adventure	G11	Horror
G3	Animation	G12	Musical
G4	Children's	G13	Mystery
G5	Comedy	G14	Romance
G6	Crime	G15	Sci-Fi
G7	Documentary	G16	Thriller
G8	Drama	G17	War
G9	Fantasy	G18	Western

The user data and the rating data are as follows:

TABLE III. USER DATABASE

Attribute	Meaning
UserID	ID of each user. Range between 1 and 6040
Gender	Gender of each user. Denoted by a 'M' or 'F'
Age	Age of each user. Express representative value
Occupation	Occupation of user
Zip-code	Address of user

TABLE IV. RATING DATABASE

Attribute	Meaning
UserID	ID of each user. Range between 1 and 6040
MovieID	ID of each movie. Range between 1 and 3952
Rating	User preference about movie
TimeStamp	Input time of rating

B. Computing Genre Correlation

Calculating genre correlation is based on genre combination in movie database. Genre is defined at the time of film production by experts such as directors or producers and, thus, shows the nature of the film very well. For example, the movie 'Toy Story' has combination of 'Animation', 'Children's', 'Comedy'. This means that the movie has characteristic of these three genres.

The 3,883 movies in database have genre combination composed of more than one genre. In this genre combination, our approach selects a genre and count number of the other genres for each movie. For example, if genre combination is G1 | G2 | G5, then G1 is selected as a criterion genre first and increase by one between a criterion genre G1 and another G2 and G5. Next, G2 is selected as a criterion genre, and increase by one between G2 and G5.

The genre counting in a movie with n-genres is as follows: For the first genre, we count all the following genres and move to the second genre. Then, from the third genre, we increase the counts and move to the next genre. We repeat this until the last genre. Figure 3 demonstrates the genre counting.

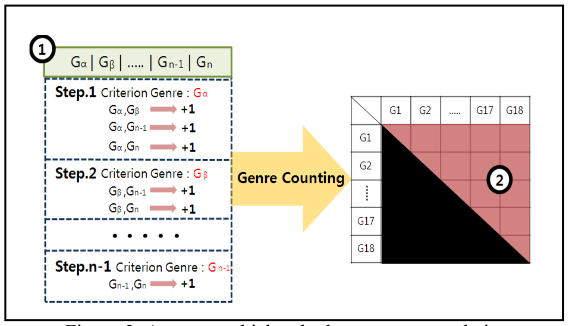
1
Gα | Gβ | ..... | Gn-1 | Gn
Step.1 Criterion Genre : Gα
Gα ,Gβ +1
Gα ,Gn-1 +1
Gα ,Gn +1
Step.2 Criterion Genre : Gβ
Gβ ,Gn-1 +1
Gβ ,Gn +1
Step.n-1 Criterion Genre : Gn-1
Gn-1 ,Gn +1
Genre Counting
G1
G2
.....
G17
G18
2

Figure 3. A course which calculates genre correlation

In Figure 3, ① is genre combination for movie M and ② is result of genre counting for all movies. table VII is the genre correlation in percentage.

C. Appling Genre Correlation

Once we have computed the genre correlations of movies, we now use the following information for recommending movies. First is the mean rating of all movies obtained from the database and second is the user preference of genre for recommending movies. The second information is provided by a user who wants to receive movie recommendation.

Using Equation (3), we calculate the new score for each movie using the mean rating and the genre correlation. Then, we sort the scores in descending order.

$$\mathrm{Tr}_1 = \frac{\sum_{i\in up}\left(\sum_{j\in mg} r_{i_n j_m}\ \mu R_M\right)}{n(up)} \qquad (3)$$

In Equation (3), the parameter up is a set of user preference genres and the parameter mg is a set of genres for each movie. The value $r_{i_n j_m}$ is a genre correlation between genre i_n and genre j_m. Then, μR_M is mean rating for movie M. Therefore, Tr_1 is the score computed by applying the genre combination of movie and user preference genres to the mean rating of movie M. If genre i_n is equal to genre j_m, then $r_{i_n j_m}$ becomes 1.

.

IV. EXPERIMENT AND ANALYSIS

We test our system with 10 users who have various movie preferences. Once a user inputs his genre preferences, the system suggests 10 movies for the user. For the purpose of comparison, we also implement another straight-forward recommendation system based on the matching between movie genres and user preference genres. This system uses a different approach with the way of Sections 2 or 3. Namely, we use user preference genres to filter movies and suggest the high rated moves that have the same or similar genres to users. Equation (4) is the formula for the comparison approach.

$$\mathrm{Tr}_2 = \frac{n(up \cap mg)}{n(up)}\mu R_M \qquad (4)$$

Note that in Equation (4), the parameter up is a set of user preference genres and the parameter mg is a set of genres for movie M. μR_M is the mean rating of movie M.

TABLE V. EXAMPLE OF RECOMMENDATION RESULT. WAY 1 DENOTES THE PROPOSED APPRAOCH BASED ON THE GENRE CORRELATIONS AND WAY 2 DENOTES THE SIMPLE RECOMMENDATION SYSTEM

Way 1.		Way 2.	
No.	Title	No.	Title
1	L.A. Confidential	1	Bluebeard
2	Alien	2	Baby, The
3	Key Largo	3	Sunset Blvd.

4	M	4	Double Indemnity
5	Bluebeard	5	Maltese Falcon, The
6	Touch of Evil	6	Chinatown
7	Chinatown	7	Manchurian Candidate, The
8	Thing, The	8	Big Sleep, The
9	Devil in a blue Dress	9	Strangers on a Train
10	Dark City	10	M

Table V is a top 10 movie list for two different recommendation systems based on the same user preference genres G10(Film-Noir) and G11(Horror). The left (way 1) is the results by the proposed method based on genre correlations and the right (way 2) is the simple recommendation approach based on the matching of genres between user and movie.

As shows in the example in table V, only three movies are the same by two approaches. Furthermore, the ranking scores are different ('M', 'Bluebeard' and 'Chinatown' are the same and 'M' is ranked at 4th in way 1 whereas it is ranked at 10th in way 2).

TABLE VI. TEST RESULT COMPARISON

User No.	Way 1.	Way 2.	Genre
1	9	8	G3|G4|G5
2	6	4	G3|G14|G15
3	10	8	G4|G5
4	9	6	G11|G13|G16|G17
5	5	4	G1
6	8	4	G10|G11
7	7	8	G1|G5|G8
8	8	5	G6|G9|G15|G16
9	7	7	G2|G12|G13
10	8	7	G5|G15|G18

Table VI shows the number of satisfactory results among top 10 movies suggested by two approaches. As shown in the result, our proposed approach (way1) shows more satisfactory results than the simple recommendation approach (way 2).

V. CONCLUSION

We have proposed an algorithm that computes contents correlation and applied the proposed algorithm for GroupLens movie database that gives rise to movie genre correlations [9]. Note that movie genres are described by experts on the content itself such as director or producer. Thus, it is more reliable than genres defined by ordinary users. Furthermore, since we compute the correlations only based on movies, we do not need to worry about the cold start problem [4, 5]. The traditional recommendation system needs to have enough user preference data so that the system can find groups of users and recommend items based on the groups. If there are not enough data, then the system becomes very unreliable: This is a cold start problem in recommendation systems. We have designed a different approach. Instead of using the comparisons of user preferences, we use contents correlations for suggesting items to users. Namely, if we have enough contents, then we can recommend contents without having lots of user preferences. Notice that in the modern Web, there are lots of contents whereas it is not easy to have enough user preference information.

In the future, we aim to apply our approach to more large size of open database and conduct more experiments with more users. We plan to utilize Open API of content sharing sites with category information such as Yahoo Music or YouTube [10].

ACKNOWLEDGMENTS

This research was supported by Basic Science Research Program through NRF funded by MEST (2010-0009168).

REFERENCES

[1] http://new.music.yahoo.com [last access: May, 03, 2010]

[2] Badrul Sarwar, George Karypis, Joseph Konstan, and John Rie, "Item-based Collaborative Filtering Recommendation Algorithms", Accepted for publication at the WWW10 Conference, pp. 285-295, May, 2001.

[3] Jonathan Lee Herlocker, Joseph Konstan, Al Borchers, and John Riedl, "An Algorithm Framework for Peforming Collaborative Filtering", Proceedings of the 1999 Conference on Research and Development in Information Retrieval, pp. 230-237, 1999.

[4] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl, "Analysis of Recommendation Algorithms for E-Commerce", The ACM E-Commerce 2000 Conference, pp. 158-167, 2000.

[5] Daniel Bill and Michael Pazzani, "Learning Collaborative Information filters", Proceedings of ICML, pp. 46-54, 1998.

[6] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstorm, and John Riedl, "GroupLens: An Open Architecture for Collaborative Filtering of Netnews", Proceedings of ACM CSCW94 Conference on Computer Supported Cooperative Work, pp. 175-186, 1994.

[7] Ben Schafer, Joseph Konstan, and John Riedl, "Recommender Systems in E-Commerce", Proceedings of the 1st ACM conference on Electronic commerce, pp. 158-166, 1999.

[8] Katsuhiro Honda, Akira Notsu, and Hidetomo Ichihashi, "Collaborative filtering by sequential extraction of user-item clusters based on structural balancing approach", Fuzzy Systems, 2009. FUZZ-IEEE 2009. IEEE International Conference on, pp. 1540-1545, 2009

[9] http://www.grouplens.org/node/12 [last access: May, 03, 2010]

[10] http://www.youtube.com [last access: May, 03, 2010]

TABLE VII. GENRE CORRELATIONS OF MOVIES FROM GROUPLENS [8] IN %

	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16	G17	G18
G1	-	25.29	2.13	3.23	7.15	18.45	0	10.19	10.2	0	11.21	1.81	7.64	5.89	25	21.83	22.88	18.51
G2	17.04	-	7.48	20.14	4.84	3.02	0	3.36	23.81	1.78	3.58	5.42	1.91	4.54	15.65	5.09	5.97	7.4
G3	0.53	2.76	-	20.89	2.75	0	0	0.1	4.081	1.78	0.44	19.27	0.63	0.67	1.86	0.65	0.99	0
G4	1.73	16.01	44.91	-	10.23	0	0	2.75	25.85	0	0.44	22.28	1.27	1.17	3.27	0.16	0.99	3.7
G5	8.65	8.69	13.36	23.13	-	12.08	36.36	23.03	12.92	1.78	18.38	24.69	8.28	34.34	7.24	5.09	8.95	31.48
G6	7.32	1.77	0	0	3.96	-	0	9.17	0.68	26.78	2.69	0	8.28	1.51	1.4	9.52	0	0
G7	0	0	0	0	0.44	0	-	0.41	0	0	0	1.2	0	0	0	0	0.49	0
G8	13.31	6.52	0.53	6.71	24.86	30.2	36.36	-	6.12	10.71	5.38	9.03	20.38	34.34	5.37	18.06	37.81	24.07
G9	1.99	6.91	3.21	9.45	2.09	0.33	0	0.91	-	0	0	1.2	0	1.17	3.037	0.16	0.49	0
G10	0	0.19	0.53	0	0.11	5.03	0	0.61	0	-	0.44	0	5.09	0.16	0.46	3.28	0	0
G11	3.32	1.58	0.53	0.24	4.51	2.01	0	1.22	0	1.78	-	1.2	3.82	0.5	13.55	9.68	0	0
G12	0.399	1.77	17.11	9.2	4.51	0	18.18	1.52	1.36	0	0.89	-	0	3.03	0.46	0	1.49	0
G13	1.59	0.59	0.53	0.49	1.43	4.36	0	3.26	0	14.28	2.69	0	-	2.02	1.4	8.04	0	0
G14	4.66	5.33	2.13	1.74	22.44	3.02	0	20.79	4.76	1.78	1.345	10.84	7.64	-	1.63	5.41	9.95	5.55
G15	14.24	13.24	4.27	3.48	3.41	2.01	0	2.34	8.84	3.57	26	1.2	3.82	1.17	-	11.49	5.47	5.55
G16	17.7	6.12	2.13	0.24	3.41	19.46	0	11.21	0.68	35.71	26.45	0	31.21	5.55	16.35	-	3.98	1.85
G17	6.12	2.37	1.06	0.49	1.98	0	9.09	7.74	0.68	0	0	1.81	0	3.36	2.57	1.31	-	1.85
G18	1.33	0.79	0	0.49	1.87	0	0	1.32	0	0	0	0	0	0.5	0.7	0.16	0.49	-