Analyzing Category Correlations for Recommendation System

Bernhard Scholz
School of Information Technologies
The University of Sydney, NSW 2006, Australia
scholz@it.usyd.edu.au

Sang-Min Choi, Sang-Ki Ko, Hae-Sung Eom and Yo-Sub Han*
Department of Computer Science
Yonsei University, Seoul, Republic of Korea
{jerassi, narame7, haesung, emmous}@cs.yonsei.ac.kr

ABSTRACT

Since the late 20th century, the Internet users have noticeably increased and these users have provided lots of information on the Web and searched for information from the Web. Now there are huge amount of new information on the Web everyday. However, not all data are reliable and valuable. This implies that it becomes more and more difficult to find a satisfactory result from the Web. We often iterate searching several times to find what we are looking for. Researcher suggests a recommendation system to solve this problem. Instead of searching several times, a recommendation system proposes relevant information. In the Web 2.0 era, a recommendation system often relies on the collaborative filtering from users. In general, the collaborative filtering approach works based on user information such as gender, location or preference. However, it may cause the cold-star problem or the sparsity problem since it requires initial user information. Recently, there are several attempts to tackle these collaborative filtering problems. One of such attempts is to use category correlation of contents. For instance, a movie has genre information given by movie experts and directors. We notice that these category information are more reliable compared with user ratings. Moreover, a newly created content always has category information; namely, we can avoid the cold-start problem. We consider a movie recommendation system. We revisit the previous algorithm using genre correlation and improve the algorithm. We also test the modified algorithm and analyze the results with respect to a characteristic of genre correlations.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Selection process

*Corresponding Author

ICUIMC'11, February, 21–23, 2011, Seoul, Korea

General Terms

Algorithm, Experimentation

Keywords

Recommendation system, Cold-start problem, Sparsity problem, Genre Correlation

1. INTRODUCTION

Since the late 20th century the Internet has become a useful tool for our life. Most of the Internet users search information on the Web everyday. Recently, in the Web 2.0 era, the number of Internet users and the amount of information heavily increase due to the various user participations on the Web. For instance, people use YouTube[1] to upload their own UGCs (User Generated Content) and share them with friends. They also use web service suck as Google[2] or Yahoo[3] to search image, song or video contents available on the Web. Now the Web becomes a platform to upload and share contents and, a lot of meaningful data are available on the Web. However, the huge amount of data often become an obstacle to find good information since there are lots of spam data and wrong information at the same time. Because of this problem, we often need to go through all search results to find a relevant result. Namely, the accuracy and the reliability of search results become low.

Researchers suggest recommendation techniques to resolve this problem [6, 8, 13]. In recommendation system, users do not need to go through all search results. The recommendation system filter searched results and present users relevant results only. In the Web 2.0, recommendation systems often rely on the collaborative filtering approach [1, 2, 9], which is one of the collective intelligence techniques. In general, the collaborative filtering approach works based on user information such as ratings, locations or preferences. For example, there is a A who wants a certain recommendation. The traditional way of collaborative filtering is to first select its neighbors. The neighbors are a group of users who have a similar preference with user A [4, 10]. The next step is to select items based on the preferences of neighbors and

[1]http://www.youtube.com
[2]http://www.google.com
[3]http://www.yahoo.com

suggest selected items to A. Since the traditional collaborative filtering approach is based on user information, the recommendation systems using collaborative filtering may not perform well if there are not enough user information [5]. People then develop some other approaches to avoid the problems caused by the traditional collaborative filtering. One of such approaches is to use category information of data. Note that the web pages do not have much category information. However, media contents such as movies or songs have category information. We observe that the category information is very reliable since it is given by experts. For instance, a movie category (genre) is decided by its directory or a producer. Moreover, this information is provided when a content is created. For the traditional collaborative filtering approach, it should have enough information from users to recommend contents. On the other hands, for the category information, we already have reliable information for contents to recommend. Based on this observation, we revisit the previous movie recommendation system based on genre correlation [3], improve an genre correlation algorithm and compare between the previous algorithm and the modified algorithm. Next we analyze a characteristic of genre correlations and suggest an approach using genre correlations for small-size memory devices. Finally, we illustrate the extensibility of genre correlations for recommendation systems.

2. RELATED WORKS

We briefly explain the general collaborative filtering approach and known problems of the collaborative filtering approach. The problems are sparsity and cold-start. Popescul [8] and Wilson [13] researched to resolve sparsity problem using various methods such as probabilistic model and case-based approach. Ishikawa and Yamaguchi [7] researched to resolve cold-start problem using information diffusion approach. We then describe the movie recommendation system based on the genre correlation suggested by Choi and Han [3].

2.1 Collaborative Filtering based on User Preference

A collaborative filtering based on user preference is described using the following three steps: The first step is to calculate correlation coefficient using user preferences. The next step is to choose neighbors for user A who wants recommendations. Neighbors are a group of users who have similar preference with A. The last step is to estimate preference for specific item based on ratings of neighbors. The detailed explanations are as follows:

2.1.1 Calculating Correlation Coefficient

This step is to calculate the correlation coefficient of user who needs recommendations with the other users using Equation (1). (It is called *Pearson correlation coefficient* [2, 9, 10].)

$$\rho_{xy} = \frac{cov(X,Y)}{\sigma_X \sigma_Y} = \frac{\sum (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum (X_i - \bar{X})^2} \sqrt{\sum (Y_i - \bar{Y})^2}}. \quad (1)$$

In Equation (1), X is user who needs recommendations and $\bar{X}$ means an average rating of X. X_i denotes the rating for the ith item by user X. Let Y be the other users. The result of the Pearson correlation coefficient is the real number between -1 and 1. If the result between X and Y is close to 1, then we say that X and Y have similar preferences. Otherwise, X and Y have different preferences.

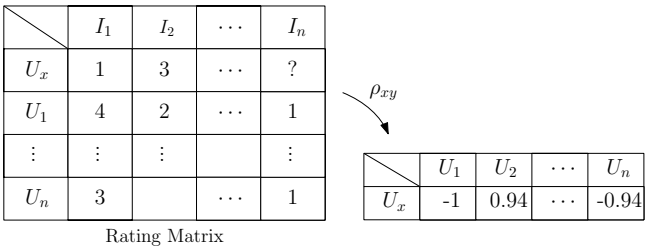

Figure 1: An example of Pearson correlation coefficient calculation

Figure 1 shows an example of the first step. In this figure, the result of Pearson correlation coefficient between U_x and U_1 is -1, and U_x and U_2 is 0.94. Thus, we can consider that the U_2 is similar user with U_x.

2.1.2 Selecting Neighbors

This step is to choose neighbors using the result of Equation (1). For this step, we first fix a certain correlation coefficient value (which is close to 1) as a threshold and select users whose correlation coefficient is more than the threshold as neighbors.

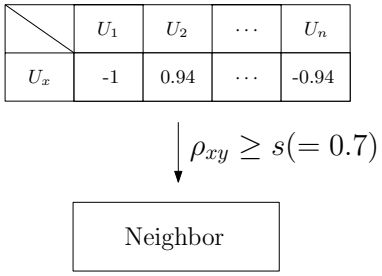

Figure 2: Selecting users whose correlation coefficient is larger than the threshold S

Figure 2 demonstrates the second step. Here S is a threshold for deciding whether or not a user is a neighbor of U_x. For example, the value of threshold is 0.7, in the Figure 2 and we can select U_2 as a neighbor of U_x.

2.1.3 Predicting Preference

The final step is to predict preference based on ratings of neighbors. This step uses Equation (2).

$$P = \bar{X} + \frac{\sum_{Y \in \text{raters}} (Y_n - \bar{Y}) \rho_{xy}}{\sum_{Y \in \text{raters}} |\rho_{xy}|}, \quad (2)$$

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

2.2 Known Issues for Collaborative Filtering based on User Preference

The collaborative filtering approach is based on user preference. It may cause some problems. First, there is the sparsity problem; it occurs where there are not enough data about user preferences. If we recommend an item using neighbors computed from a small amount of ratings, then the accuracy of recommendation is lower than using neighbors computed from a large amount of ratings [2, 9]. The sparsity problem affects the accuracy of recommendation using collaborative filtering [6, 8, 13]. The second problem is the cold-start problem. It occurs when new users or items without enough information are added [7]. The traditional collaborative filtering selects a neighbor except for uses without ratings. 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 he/she 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 [7, 11, 12].

2.3 A Content Recommendation System based on Category Correlations

General collaborative filtering approach is based on user preferences. This implies that the system should wait until it has enough input data from users. Researchers proposed several methods to avoid this problem [3, 6, 7, 8, 11, 13]. One of such approaches is to use information that is reliable and available initially. Notice that we cannot always have such information available. Thus, we choose a movie recommendation system domain since a movie has a category information (called genre) given by experts. Recently, Choi and Han [3] proposed a movie recommendation system based on genre correlations. Their system does not require lots of user preferences. The system first calculates genre correlations based on the genre combinations of each movie. Then the system applies the genre combination of all movies and user-preferred genres to the average rating of each movie based on the genre correlations. Finally, it ranks movies according to the newly computed point. The details on the system is as follows:

2.3.1 Calculating Genre Correlations

This step is based on the genre combinations of each movie in database. All movies in the movie database have at least one genre. Namely, each movie has a genre combination composed of at least one genre. Genre information is decided by movie experts whereas user information such as preference or ratings is decided by user himself. The recommendation system relies on this reliable genre information. In this step, the system selects a genre and counts the number of the other genres from each movie. For example, if a movie A has genre combination of G_1, G_2 and G_5, then G_1 is selected as a criterion genre and we increase the combination counting with G_2 and G_5 by 1. Next, G_2 is selected as a criterion genre and we increase the combination counting with only G_5 by 1 again. We repeat this procedure to all movies in the database and calculate the genre correlations by percentages. For example, in Table 1, frequency of G1 with G2 divide by sum of total frequency of G1 between other genres is 0.2529. Thus, genre correlation between G1 and G2 is 25.29 by percentage. Table 1 shows the genre correlations of all genres.

2.3.2 Applying Genre Correlations

This step is to apply genre correlations. Now we assume that there are user preferred genres and average ratings of movies. If a user wants movie recommendation, then he/she will provide own preferred movie genres.

$$R_p = \frac{\sum_{i \in up} (\sum_{j \in mg} r_{i,j} M_\mu)}{|up|}. \quad (3)$$

In Equation (3), up is a set of preferred genres given by user, mg is a set of genre combination for each movie. $r_{i,j}$ means the genre correlation between genre i and j and M_μ is an average rating of a movie. Therefore the result of Equation (3) R_p is the recommendation point computed by applying the genre combination of the movie and the user preferred genres to the average rating of the movie M. If the genre i is equal to the genre j, then $r_{i,j}$ becomes one.

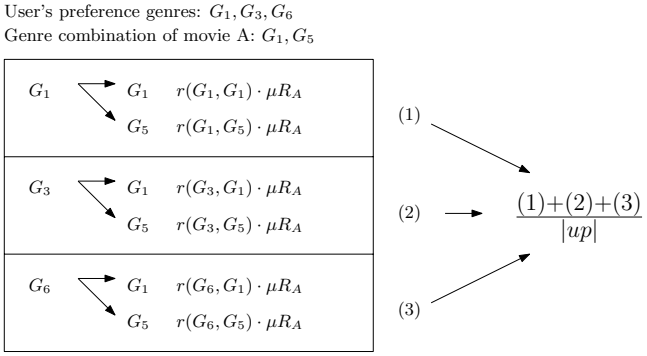

Figure 3: An example of applying genre correlations to the average rating

Figure 3 shows an example of applying genre correlations. In Figure 3, user preferred genres are G_1, G_3, and G_6, and the genre combinations of movie A are G_1 and G_5. Thus the system selects a criterion genre sequentially in user preferred genres and each criterion genre applies genre correlations to the average preference of movie A as many as the number of genres of movie A.

In summary, the system first applies the user preferred genres using genre correlations to the average rating and calculates the new recommendation points for each movie using the average rating and genre correlations. Then the system ranks movies according to newly computed recommendation points and recommends highly ranked movies.

2.3.3 Improvements of Genre Correlations

The recommendation system based on the genre correlations avoids the problems of general collaborative filtering approach. We now improve the system. In Equation (3), if genre information is increased in genre combinations of movie, then the result of Equation (3) cannot suggest the precise value when applying genre correlations. In additions, if we can find characteristics of the genre correlations, then we can compose advanced genre correlations that can be used for various devices.

3. OUR APPROACH

Table 1: Genre correlation matrix

	G_1	G_2	G_3	G_4	G_5	G_6	G_7	G_8	G_9	G_{10}	G_{11}	G_{12}	G_{13}	G_{14}	G_{15}	G_{16}	G_{17}	G_{18}
G_1	-	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
G_2	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
G_3	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
G_4	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
G_5	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
G_6	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
G_7	0	0	0	0	0.44	0	-	0.41	0	0	0	1.2	0	0	0	0	0.49	0
G_8	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
G_9	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
G_{10}	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
G_{11}	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
G_{12}	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
G_{13}	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
G_{14}	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
G_{15}	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
G_{16}	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
G_{17}	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
G_{18}	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	-

3.1 Revision of the Previous Algorithm

The previous recommendation system based on genre correlations has a problem in the step when applying genre correlations to average rating. If genre combination has many genres, then we cannot expect the precise result. The purpose of Equation (3) is to choose movies that have high correlation between users preferred genres and genre combination of each movie. However, we cannot expect precise results about genre combination which have lots of genres. For example, in the Figure 3, the number of preference genres is three, and the number of movie genres is two. In this case, first, the genre correlation between G_1 and G_1 is applied to average rating of movie, and the genre correlation between G_1 and G_5 is applied to the average rating of movie. Next, this ratings which applied each genre correlation are added. However, Equation (3) does not divide this sum. If we do not divide this sum using the number of movie genres, the step of the applying genre correlations add the sum value as much as the number of movie genres. Because of this reason, the result value becomes higher as the movie has more genres. Namely, the results of Equation (3) may suggest a movie which does not have high correlation with user preferred genre. Because of this reason we propose a new equation to solve the problem of Equation (3).

$$R_{p1} = \frac{\sum_{i \in up} \sum_{j \in mg} (r_{i=j} + \frac{r_{i \neq j}}{|mg| - 1})}{|up|} \cdot M_{\mu}. \quad (4)$$

$$R_{p2} = \frac{\sum_{i \in up} \sum_{j \in mg} r_{i \neq j}}{|up| \cdot |mg|} \cdot M_{\mu}. \quad (5)$$

Equations (4) and (5) sovle the problem of Equation (3). When we calculate the recommendation point, if the selected criterion genre in user preffered genres exists in the genre combination of specific movie, we use Equation (4). Otherwise, we use Equation (5). The difference between Equations (4) and (5) is the existence of same genre between criterion genre and one of the genre in genre combination of specific movie. In Equations (4) and (5), *up* means a set of user preffered genre and *mg* means a set of genre combination of specific movie. $r_{i=j}$ is the genre correlation when genre i is equal to genre j. Thus, the value of $r_{i=j}$ is one. $r_{i \neq j}$ is the genre correlations when genre i is not equal to genre j.

When we use Equation (3) to calculate recommendation point, the movies with a large number of genres in genre combination can obtain lower points than the movies with a small number of genres. It is because the recommendation point of a movie is divided with the number of genres of each movie. For example, suppose that a movie A has genre combination of G_1, G_2 and a movie B has G_1, G_2, G_3, G_4 and the average ratings of two movies are the same. Then, if a user inputs G_1 and G_2 as preferred genres, movie A will obtain much higher recommendation point than the movie B. In order to solve this problem, we revised this equation. Equations (4) and (5) are the revised equations. If a movie has genres that coincide with user preferred genres, we preserve the value. We only divide the correlation values between two different genres. Then, if we apply the revised equation to the previous example, movie B will obtain higher point than movie A, since movie B has genres that are preferred by the user and also has possibly related genres with them.

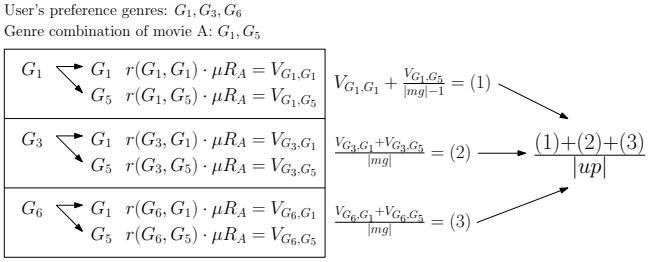

Figure 4: An example of recommendation point calculation by the revised method

The Figure 4 shows the calculating recommendation point using the Equations (4) and (5). In Figure 4, user preffered

genres are G_1, G_3 and G_6, and genre combination of movie A is G_1 and G_5. When the criterion genre is selected as G_1, we use Equation (4) since movie A has a genre G_1. The other cases (2) and (3) use Equation (5) since the selected criterion genres are not equal to the genres of movie A.

3.2 Calculating Genre Correlations with Different Numbers of Movies

We consider a characteristic of genre correlations in different numbers of movies. This case is to determine whether or not a small number of movie data can affect the genre correlations compared with a large number of movie data.

4. EXPERIMENTS AND ANALYSIS

4.1 Database

We use an open movie database called *GroupLens database*[4]. The GroupLens database has three sub-databases: movie database, user database and rating database. Table 2 is the movie database.

Table 2: Movie database

Attribute	Meaning
MovieID	ID of each movie. Between 1 and 10681
Title	Title of the movie
Genre	Genre of the movie

This database provides IDs, titles and genre combinations of each movie. The total number of movies is 10,681. Table 3 is the list of genres and Table 4 is the user database. This database provides IDs, genders, ages, occupations and zip-codes of each user. Table 5 is the rating database. This database provides user IDs, movie IDs and timestamps of each rating.

Table 3: Genre Number

No	Genre	No	Genre
G_1	Action	G_{10}	Film-Noir
G_2	Adventure	G_{11}	Horror
G_3	Animation	G_{12}	Musical
G_4	Children's	G_{13}	Mystery
G_5	Comedy	G_{14}	Romance
G_6	Crime	G_{15}	Sci-Fi
G_7	Documentary	G_{16}	Thriller
G_8	Drama	G_{17}	War
G_9	Fantasy	G_{18}	Western

Table 4: User database

Attribute	Meaning
UserID	ID of each user.
Gender	Gender of each user. 'M' or 'F'
Age	Age of each user.
Occupation	Occupation of user
Zip-code	Address of user

[4]http://www.grouplens.org/node/12

Table 5: Rating database

Attribute	Meaning
UserID	ID of each user.
MovieID	ID of each movie.
Rating	User preference about movie
TimeStamp	Input time of rating

4.2 Comparison of previous method and revised method

Table 6 shows the top ten recommended movies by the previous method and the revised method. We input 'Drama, Romance' as genre combination. In the previous method, if a movie has more genres than other movies, the movie gets more points than other movies in the condition of the same average ratings since we just add up genre correlation values in the previous method. We can obtain more precisely recommended movies by the revised method because we average the genre correlation values. Although a movie has many genres including genres suggested by users, it would not be preferred by the revised method. Table 6 shows that with results. Stunt Man, The (1980) receives the highest score among all movies by the previous method. (The genre combination is 'Action, Adventure, Comedy, Romance, Thriller'.) The right side of Table 6 shows the result by the revised method. Shadows of Forgotten Ancestors (1964) is the top movie by the revised method. The genre combination of this movie is exactly the same with input genre combination 'Drama, Romance'. City Lights (1931) is the fourth movie by the revised method whose genre combination is 'Comedy, Drama, Romance'. This means 'Comedy' genre has high correlations with 'Drama' and 'Romance'. As shown in Table 1, 'Comedy' genre has high correlations with 'Drama' and 'Romance'. Therefore, City Lights (1931) is a desirable recommendation.

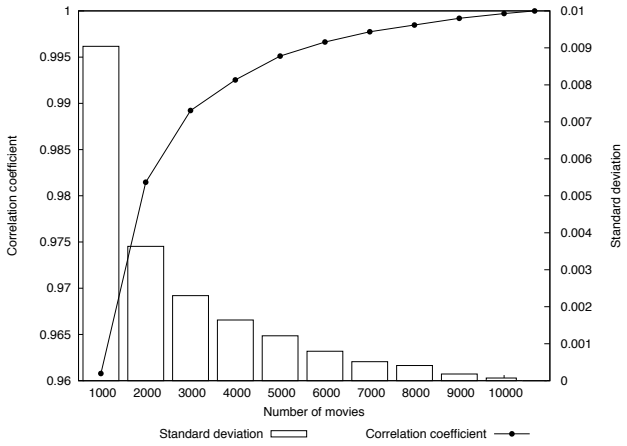

Figure 5: Comparisons between genre correlation matrices with different numbers of movies

4.3 Comparison of correlation matrices with different numbers of movies

For the comparison of the recommendation results from different numbers of movies, we use fourteen subsets with different numbers of movies: 100, 200, 500, 1000, 2000, 3000,

Table 6: The top 10 recommended movies by previous method and revised method

	By the previous method		By the revised method
1	Stunt Man, The (1980)	1	Shadows of Forgotten Ancestors (1964)
2	Life Is Beautiful (La Vita Ãİ bella) (1997)	2	Casablanca (1942)
3	Band of Outsiders (Bande Ãă part) (1964)	3	Children of Paradise (Les enfants du paradis) (1945)
4	City Lights (1931)	4	City Lights (1931)
5	Slumdog Millionaire (2008)	5	Slumdog Millionaire (2008)
6	Cinema Paradiso (Nuovo cinema Paradiso) (1989)	6	Cinema Paradiso (Nuovo cinema Paradiso) (1989)
7	Shadows of Forgotten Ancestors (1964)	7	Bad Blood (Mauvais sang) (1986)
8	Eternal Sunshine of the Spotless Mind (2004)	8	Dodsworth (1936)
9	Another Thin Man (1939)	9	Persuasion (1995)
10	Forrest Gump (1994)	10	Graduate, The (1967)

$\cdots$, 9000, 10000 and 10681. We construct genre correlation matrices 100 times per each subset. Since the same set of movies gives the same correlation matrix we use movie sets that are selected randomly. Since there are all fourteen subsets, we construct correlation matrix 1400 times as a result. See Figure 5: this is a graph that has two y axes with one axis representing the correlation coefficient and the other axis representing the standard deviation. We omit the values for 100, 200, 500 and 10681 for better presentation in Figure 5.

When we calculate the correlation between each number of movies, we first extract subsets from the total set. Next we calculate correlations between the total set and each subset using the Pearson correlation coefficient.

$$R = \frac{\sum_{i=1}^{|G_n|} \rho_{x_i y_i}}{|G_n|}. \quad (6)$$

Note that R in Equation (6) is an average of correlation coefficients. Lastly, we repeat this for 100 times and obtain the average correlation coefficient for each subset. The histogram in Figure 5 shows the standard deviation of 100 correlation coefficients.

The standard deviation for 1000-movie subset is slightly lower than 0.01. For the other subsets, the standard deviations are much lower than 0.01. The histogram shows a sharp decrease between 1000 and 2000 movies. The correlation coefficients form the reverse of the standard deviations. The correlation coefficient increases rapidly between 1000 and 2000 movies too.

We then apply the proposed algorithm in Section 3.1 to the subsets of movies that we have used, and verify the usefulness of the algorithm. We compare the top 10 movies from each subset of movies and the top 10 movies from the total set 10 times. Figure 6 shows the index of coincidence: It shows if the number of movies is bigger than 2000, then the result is almost same with the case of the total set. This implies that we can compute the genre correlation with a certain number of movies (in here 2000) instead of whole set and recommend movies.

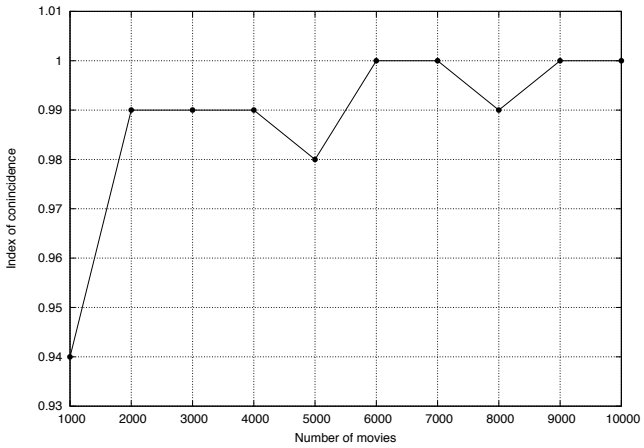

Figure 6: The graph shows the index of coincidence between two recommendation results using each subset of movies and the total set of movies

5. CONCLUSIONS

The traditional recommendation system based on collaborative filtering requires 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 may cause the cold-start problem in recommendation systems. Various approaches appear to tackle this problem [3, 5, 7]. One of which is a movie recommendation system based on the genre correlations. Note that movie genres are described by experts such as directors or producers. Since the previous algorithm uses genre information, we do not need to consider the cold-start problem [2, 9]. We have reexamined this approach and improved the algorithm. We have proposed an algorithm that constructs genre correlations and applied the algorithm to the GroupLens movie database. The experimental results have shown that the new algorithm can provide better recommendations. Next, we have considered a subset of movies and demonstrated that a reasonable size of a subset can still provide good recommendation. In future, we aim to apply the proposed method to more large size of open database such as Yahoo Music or YouTube, and conduct more experiments.

6. ACKNOWLEDGMENTS

The Yonsei university research group is supported by the Basic Science Research Program through NRF funded by MEST (2010-0009168).

7. REFERENCES

[1] R. M. Bell and Y. Koren. Lessons from the netflix prize challenge. *SIGKDD Explorations*, 9(2):75–79, 2007.

[2] D. Billsus and M. J. Pazzani. Learning collaborative information filters. In *The 15th International Conference on Machine Learning*, pages 46–54, 1998.

[3] S.-M. Choi and Y.-S. Han. A content recommendation system based on category correlations. In *The Fifth International Multi-Conference on Computing in the Global Information Technology*, pages 1257–1260, 2010.

[4] J. L. Herlocker, J. Konstan, A. Borchers, and J. Riedl. An algorithm framework for peforming collaborative filtering. In *Proceedings of the 1999 Conference on Research and Development in Information Retrieval*, pages 230–237, 1999.

[5] K. Honda, A. Notsu, and H. Ichihashi. Collaborative filtering by sequential extraction of user-item clusters based on structural balancing approach. In *Fuzzy Systems, 2009.*, pages 1540–1545, 2009.

[6] Z. Huang, H. Chen, and D. D. Zeng. Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering. *ACM Trans. Inf. Syst.*, 22(1):116–142, 2004.

[7] M. Ishikawa, P. Géczy, N. Izumi, T. Morita, and T. Yamaguchi. Information diffusion approach to cold-start problem. In *Web Intelligence/IAT Workshops*, pages 129–132, 2007.

[8] A. Popescul, L. H. Ungar, D. M. Pennock, and S. Lawrence. Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments. In *Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence*, pages 437–444, 2001.

[9] B. M. Sarwar, G. Karypis, J. A. Konstan, and J. Riedl. Analysis of recommendation algorithms for e-commerce. In *ACM Conference on Electronic Commerce*, pages 158–167, 2000.

[10] B. M. Sarwar, G. Karypis, J. A. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International World Wide Web Conference*, pages 285–295, 2001.

[11] A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock. Methods and metrics for cold-start recommendations. In *The 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 253–260, 2002.

[12] T. Y. Tang and G. I. McCalla. Utilizing artificial learners to help overcome the cold-start problem in a pedagogically-oriented paper recommendation system. In *Adaptive Hypermedia and Adaptive Web-Based Systems, Third International Conference*, pages 245–254, 2004.

[13] D. C. Wilson, B. Smyth, and D. O'Sullivan. Sparsity reduction in collaborative recommendation: A case-based approach. *IJPRAI*, 17(5):863–884, 2003.