Expert Systems with Applications 39 (2012) 8079–8085

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

A movie recommendation algorithm based on genre correlations ☆

Sang-Min Choi, Sang-Ki Ko, Yo-Sub Han *

Department of Computer Science, Yonsei University, Seoul 120-749, Republic of Korea

ARTICLE INFO

Keywords:
Recommendation algorithm
Genre correlation
Cold-start problem
Sparsity problem

ABSTRACT

Since the late 20th century, the number of Internet users has increased dramatically, as has the number of Web searches performed on a daily basis and the amount of information available. A huge amount of new information is transferred to the Web on a daily basis. However, not all data are reliable and valuable, which implies that it may become more and more difficult to obtain satisfactory results from Web searches. We often iterate searches several times to find what we are looking for. To solve this problem, researchers have suggested the use of recommendation systems. Instead of searching for the same information several times, a recommendation system proposes relevant information. In the Web 2.0 era, recommendation systems often rely on collaborative filtering by users. In general, a collaborative filtering approach based on user information such as gender, location, or preference is effective. However, the traditional approach can fail due to the cold-start problem or the sparsity problem, because initial user information is required for this approach to be effective. Recently, several attempts have been made to tackle these collaborative filtering problems. One such attempt used category correlations of contents. For instance, a movie has genre information provided by movie experts and directors. This category information is more reliable than user ratings. Moreover, newly created content always has category information, allowing avoidance of the cold-start problem. In this study, we consider a movie recommendation system and improve the previous algorithms based on genre correlations to correct its shortcomings. We also test the modified algorithm and analyze the results with respect to two characteristics of genre correlations.

1. Introduction

Since its introduction in the late 20th century, the Internet has become a powerful tool in everyday life. Most Internet users search for information on the Web every day. In the Web 2.0 era, the number of Internet users and the amount of information uploaded have increased tremendously due to sites such as YouTube[1] where users upload user generated content (UGC) to share among friends. In addition, users make use of Web services such as Google[2] and Yahoo[3] to search for images, songs, or video content available on the Web. The Web has thus become a platform to upload and share content, and a lot of meaningful data are available on the Web. However, the huge amount of data is often an obstacle to finding relevant information, because there is so much spam data and erroneously information that is simultaneously present. Because of this problem, search results often have to be thoroughly scrutinized to find relevant results. This continuously increasing amount of information can decrease the accuracy and reliability of search results.

Several researchers have suggested recommendation systems to resolve this problem (Huang, Chen, & Zeng, 2004; Popescul, Ungar, Pennock, & Lawrence, 2001; Wilson, Smyth, & O'Sullivan, 2003). In a recommendation system, users do not need to scan through all search results. The recommendation system filters the results from the search and presents users with the relevant results only. In Web 2.0, recommendation systems often rely on the collaborative filtering approach (Bell & Koren, 2007; Billsus & Pazzani, 1998; Sarwar, Karypis, Konstan, & Riedl, 2000), which is a collective intelligence technique. In general, a collaborative filtering approach uses user information such as ratings, locations, or preferences to filter results. For example, consider the case where person *A* wants a certain recommendation. The traditional way of collaborative filtering is to first select neighbors. The neighbors are a group of users with similar preferences to user *A* (Herlocker, Konstan, Borchers, & Riedl, 1999; Sarwar, Karypis, Konstan, & Riedl, 2001). The next step is to select items based on the preferences of neighbors and suggest selected items to *A*. Because the traditional collaborative filtering approach is based on user information, recommendation systems based on collaborative filtering may not perform well if there is not enough user information available (Honda, Notsu, & Ichihashi, 2009).

☆ A preliminary version of this paper appeared in Proceedings of the 5th International Conference on Ubiquitous Information Management and Communication, ICUIMC'11, 2011, pp. 1:1–1:7 (Scholz, Choi, Ko, Eom, & Han, 2011).

* Corresponding author. Tel.: +82 2 2123 5725; fax: +82 2 365 2579.
E-mail addresses: jerassi@cs.yonsei.ac.kr (S.-M. Choi), narame7@cs.yonsei.ac.kr (S.-K. Ko), emmous@cs.yonsei.ac.kr (Y.-S. Han).

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

0957-4174/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2012.01.132

Researchers have therefore developed other approaches to avoid the problems associated with traditional collaborative filtering. One such approach involves the use of category information. Note that Web pages do not have much category information. However, media content such as movies or songs do have associated category information. This category information is usually highly reliable because it is provided by experts. For instance, the genre/category of a movie is decided on by the movie's director(s) and/or producer(s). Moreover, this information is provided when content is created. The traditional collaborative filtering approach requires enough information from users to recommend content. In contrast, for category information, reliable information is already available for content recommendation. Based on this observation, we revisit the previous movie recommendation system based on genre correlations (Choi & Han, 2010), improve the existing genre correlation algorithm, and compare the results obtained using the previous algorithm and the new, modified algorithm. We also analyze characteristics 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 studies

In this section, we provide a brief description of the general collaborative filtering approach and known problems of the collaborative filtering approach. The two major problems associated with this approach are sparsity and cold-start. Popescul et al. (2001), Wilson et al. (2003) attempted to resolve the sparsity problem using various methods such as a probabilistic model and a case-based approach. Ishikawa, Géczy, Izumi, Morita, and Yamaguchi (2007) attempted to resolve the cold-start problem using an information diffusion approach. We describe the movie recommendation system based on the genre correlation method suggested by Choi and Han (2010).

2.1. Collaborative filtering based on user preference

A collaborative filtering approach based on user preferences involves the following three steps: (1) Calculate the correlation coefficient using user preferences, (2) choose neighbors of user A who wants recommendations (neighbors are a group of users who have similar preferences to user A) and (3) estimate the preference for a specific item based on the neighbors ratings. Details of these steps are provided below.

2.1.1. Calculating the correlation coefficient

To calculate the correlation coefficient of the user who wants recommendations with other users, Eq. (1), the so-called *Pearson correlation coefficient* (Billsus & Pazzani, 1998; Sarwar et al., 2000, 2001), is used:

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

where X is the user who needs recommendations, $\overline{X}$ is the average rating of X, X_i denotes the rating for the ith item by user X, $R_\mu(i)$ is the average rating of content i by user S, and Y represents the other users.

Fig. 1 shows an example of the first step. In this figure, the Pearson correlation coefficient between U_x and U_1 is -1, and that between U_x and U_2 is 0.94. This indicates that U_2 has similar preferences to U_x.

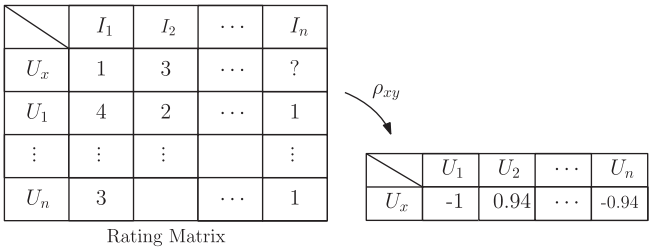

Fig. 1. The Pearson correlation coefficient calculation.

2.1.2. Selecting neighbors

In the next step, neighbors are chosen using the results of Eq. (1). In this step, a certain correlation coefficient value (close to 1) is first fixed as a threshold and users with a correlation coefficient to U_x greater than this threshold are selected as neighbors.

2.1.3. Predicting preferences

The final step is to predict preferences based on the ratings of neighbors. This step uses Eq. (2):

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

where $\overline{X}$ is the average rating of user X and Y_n is the rating by the other users for the nth item. Y is the average rating by the neighbors of X of 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 ratings for the item of interest. The result P in Eq. (2) is the predicted value of an item for user X.

2.2. Known issues associated with user preference-based collaborative filtering

The collaborative filtering approach is based on user preferences, which may cause some problems. First, there is the sparsity problem; this occurs where there are not enough data available about user preferences. If we recommend an item using neighbors computed from a small amount of ratings, then the accuracy of the recommendation will be lower than that obtained using neighbors computed from a large number of ratings (Billsus & Pazzani, 1998; Sarwar et al., 2000). The sparsity problem affects the accuracy of recommendation using collaborative filtering (Huang et al., 2004; Popescul et al., 2001; Wilson et al., 2003). The second problem is the cold-start problem. This occurs when new users or items without enough information are added (Ishikawa et al., 2007). In the traditional collaborative filtering approach, neighbors are only selected from among the users with ratings. This implies that a user without ratings cannot have a neighbor. In other words, the system cannot recommend an item to a new user who has no ratings before he/she inputs more than a certain number of ratings. Thus, the system has to wait until all users rate enough items before recommending items (Ishikawa et al., 2007; Schein, Popescul, Ungar, & Pennock, 2002; Tang & McCalla, 2004).

2.3. A content recommendation system based on category correlations

In general, the collaborative filtering approach is based on user preferences. This implies that the system should wait until it has enough input data from users. Researchers have proposed several methods to avoid this problem (Choi & Han, 2010; Huang et al., 2004; Ishikawa et al., 2007; Popescul et al., 2001; Schein et al., 2002; Wilson et al., 2003). One such approach is to use information that is reliable and available initially. Note that this type of

information may not always be available. We chose to use a movie recommendation system domain because a movie has category information (genre) provided by experts. Recently, Choi and Han (2010) proposed a movie recommendation system based on genre correlations. Their system does not require a large input 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 computed recommendation points. The details of this system are outlined below.

2.3.1. Calculating genre correlations

Genre correlations are initially calculated based on genre combinations of each movie in a database. All movies in the movie database have at least one genre. In other words, each movie has a genre combination composed of at least one genre. Genre information is provided by movie experts, whereas user information such as preferences or ratings are decided by the user him/himself. The recommendation system relies on the reliable genre information provided by experts. The system selects a genre and counts the number of the other genres for each movie. For example, if movie A has the 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 for all movies in the database and calculate the genre correlations by percentages. For example, in Table 1, the frequency of G_1 and G_2 divided by the sum of the total frequency of G_1 between other genres is 0.2529. Thus, the genre correlation between G_1 and G_2 is 25.29%. Table 1 shows the genre correlations of all genres.

Table 1
Genre correlation matrix.

	G_1	G_2	G_3	G_4	G_5	G_6	G_7	G_8	G_9
G_1	–	25.29	2.13	3.23	7.15	18.45	0	10.19	10.2
G_2	17.04	–	7.48	20.14	4.84	3.02	0	3.36	23.81
G_3	0.53	2.76	–	20.89	2.75	0	0	0.1	4.081
G_4	1.73	16.01	44.91	–	10.23	0	0	2.75	25.85
G_5	8.65	8.69	13.36	23.13	–	12.08	36.36	23.03	12.92
G_6	7.32	1.77	0	0	3.96	–	0	9.17	0.68
G_7	0	0	0	0	0.44	0	–	0.41	0
G_8	13.31	6.52	0.53	6.71	24.86	30.2	36.36	–	6.12
G_9	1.99	6.91	3.21	9.45	2.09	0.33	0	0.91	–
G_{10}	0	0.19	0.53	0	0.11	5.03	0	0.61	0
G_{11}	3.32	1.58	0.53	0.24	4.51	2.01	0	1.22	0
G_{12}	0.399	1.77	17.11	9.2	4.51	0	18.18	1.52	1.36
G_{13}	1.59	0.59	0.53	0.49	1.43	4.36	0	3.26	0
G_{14}	4.66	5.33	2.13	1.74	22.44	3.02	0	20.79	4.76
G_{15}	14.24	13.24	4.27	3.48	3.41	2.01	0	2.34	8.84
G_{16}	17.7	6.12	2.13	0.24	3.41	19.46	0	11.21	0.68
G_{17}	6.12	2.37	1.06	0.49	1.98	0	9.09	7.74	0.68
G_{18}	1.33	0.79	0	0.49	1.87	0	0	1.32	0

	G_{10}	G_{11}	G_{12}	G_{13}	G_{14}	G_{15}	G_{16}	G_{17}	G_{18}
G_1	0	11.21	1.81	7.64	5.89	25	21.83	22.88	18.51
G_2	1.78	3.58	5.42	1.91	4.54	15.65	5.09	5.97	7.4
G_3	1.78	0.44	19.27	0.63	0.67	1.86	0.65	0.99	0
G_4	0	0.44	22.28	1.27	1.17	3.27	0.16	0.99	3.7
G_5	1.78	18.38	24.69	8.28	34.34	7.24	5.09	8.95	31.48
G_6	26.78	2.69	0	8.28	1.51	1.4	9.52	0	0
G_7	0	0	1.2	0	0	0	0	0.49	0
G_8	10.71	5.38	9.03	20.38	34.34	5.37	18.06	37.81	24.07
G_9	0	0	1.2	0	1.17	3.037	0.16	0.49	0
G_{10}	–	0.44	0	5.09	0.16	0.46	3.28	0	0
G_{11}	1.78	–	1.2	3.82	0.5	13.55	9.68	0	0
G_{12}	0	0.89	–	0	3.03	0.46	0	1.49	0
G_{13}	14.28	2.69	0	–	2.02	1.4	8.04	0	0
G_{14}	1.78	1.345	10.84	7.64	–	1.63	5.41	9.95	5.55
G_{15}	3.57	26	1.2	3.82	1.17	–	11.49	5.47	5.55
G_{16}	35.71	26.45	0	31.21	5.55	16.35	–	3.98	1.85
G_{17}	0	0	1.81	0	3.36	2.57	1.31	–	1.85
G_{18}	0	0	0	0	0.5	0.7	0.16	0.49	–

2.3.2. Applying genre correlations

In the next step, the genre correlations are applied using Eq. (3). We assume that there are user preferred genres and average ratings of movies. If the user wants a movie recommendation, then s/he will provide his/her own preferred movie genres.

$$R_p = \frac{\sum_{i \in up}\left(\sum_{j \in mg} r_{ij} M_\mu\right)}{|up|}. \tag{3}$$

In Eq. (3), up is a set of preferred genres provided by the user, and mg is the set of genre combinations for each movie. r_{ij} indicates the genre correlation between genres i and j while M_μ is the average rating of a movie. The result of Eq. (3), R_p, is the recommendation points computed by applying the genre combination of the movie and the preferred genres of the user to the average rating of movie M. If the genre i is equal to the genre j, then r_{ij} becomes one.

An example of the application of genre correlations is shown in Fig. 2. In Fig. 2, the user's 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 from among the user's preferred genres and each criterion genre applies as many genre correlations to the average preference of movie A as the number of genres of movie A. In summary, the system first applies the user's preferred genres using genre correlations to the average rating and calculates recommendation points for each movie using the average ratings and genre correlations. Then, the system ranks movies according to the newly computed recommendation points and recommends highly ranked movies.

2.3.3. Improvement of the genre correlation approach

A recommendation system based on genre correlations avoids the problems associated with the general collaborative filtering approach. However, if the number of genre combinations of a movie is large, the results obtained using Eq. (3) are not accurate reflections of genre correlations. In addition, if we can determine characteristics of genre correlations, then we can compose advanced genre correlations that can be used for various devices. For these reasons, we attempted to improve the existing genre correlation algorithm.

3. Our approach

3.1. Revision of the previous algorithm

The step in the previous recommendation-based system when genre correlations are applied to average ratings is problematic.

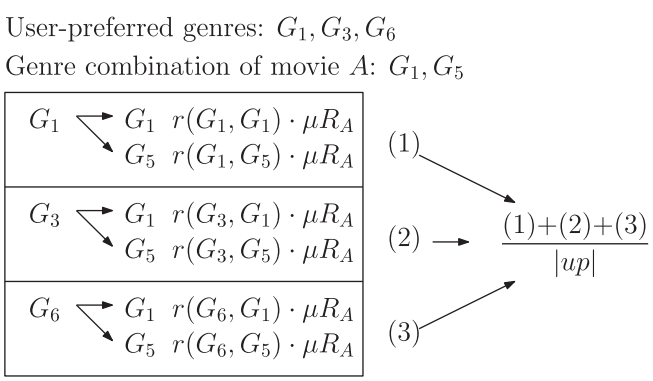

Fig. 2. An example of applying genre correlations to the average rating.

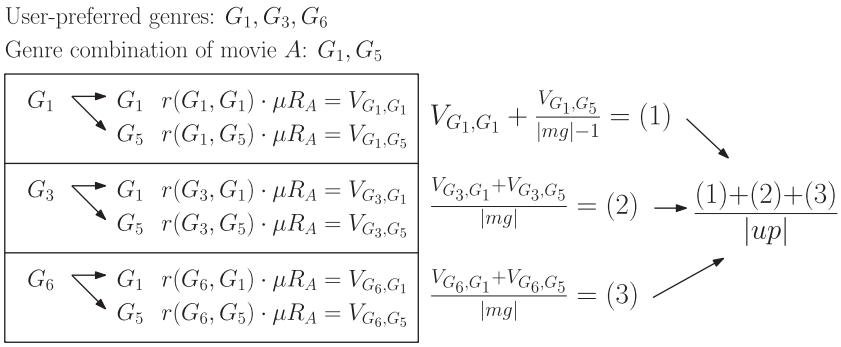

Fig. 3. An example of the calculation of recommendation points using our revised equations.

If a movie has multiple genres, then the results may be imprecise. The purpose of Eq. (3) is to choose movies that show high correlations between the user's preferred genres and the genre combination of each movie. However, take for example the case where the number of preference genres is three and the number of movie genres is two. In this case, the genre correlation between G_1 and G_1 is first applied to the average rating of the movie, and the genre correlation between G_1 and G_5 is then applied to the average rating of the movie. Next, these ratings are added. However, Eq. (3) does not divide this sum. If this sum is not divided using the number of movie genres, the sum value is much higher than the number of movie genres. For this reason, the recommendation points will increase as the number of genres of a movie increases. In other words, a movie that is not strongly correlated with the user's preferred genres may be highly recommended. We therefore propose the following equations:

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

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

Eqs. (4) and (5) solve the potential problem that can occur if Eq. (3) is used. If the selected criterion genre of the movie is one of the user's preferred genres, Eq. (4) is used. Otherwise, Eq. (5) is used. The difference between Eqs. (4) and (5) is the presence of the same genre in the criterion genre and the genre combination set. In Eqs. (4) and (5), *up* refers to the set of the user's preferred genres while *mg* indicates the genre combinations of a 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 correlation when genre i is not equal to genre j.

If Eq. (3) is used to calculate recommendations points, movies with a large number of genres could score lower than movies with a small number of genres because the recommendation points for a movie are divided by the number of genres of each movie. For example, suppose that movie A has the genre combination of G_1, G_2 and movie B has the genre combination of G_1, G_2, G_3, and G_4 and the average ratings of the two movies are the same. Then, if a user inputs G_1 and G_2 as preferred genres, movie A will receive a higher recommendation than movie B. We developed Eqs. (4) and (5) to address this problem. If a movie has genres that coincide with the user's preferred genres, then we preserve the value. We only divide the correlation values between two different genres. If we apply the revised equation to the previous example, movie B will receive more recommendation points than movie A, because the genres of movie B are among the user's preferred genres.

Fig. 3 shows how recommendation points are calculated using Eqs. (4) and (5). In Fig. 3, the user's preferred genres are G_1, G_3, and G_6, and the genre combination of movie A is G_1 and G_5. When the criterion genre is selected as G_1, Eq. (4) is used because movie A is of the genre G_1. For the other two cases (2 and 3), Eq. (5) is used because the selected criterion genres and the genres of movie A are different.

3.2. Advanced genre correlations

Advanced genre correlations can also be constructed using revised equations. To improve genre correlations, we analyzed the characteristics of genres.

There are two ways to consider genre correlations:

- Way1: According to the number of genres.
- Way2: According to the decade when the movie was made.

Way1 can reveal changes in genre correlations when movie data is limited, while way2 can indicate changes in genre correlations according to periods. Thus, we can construct accurate genre correlations based on a limited amount of data, and provide accurate ratings for users who have preferences for movies from particular decades.

4. Experiments and analysis

4.1. Database

We used an open movie database called *GroupLens database*[4]. The GroupLens database has three sub-databases: a movie database, a user database, and a rating database. Table 2 shows the characteristics of the movie database

This database contains IDs, titles, and genre combinations for all movies in the database. The total number of movies in the database at the time of this study was 10,681. Table 3 is the list of genres and Table 4 is the user database. This database contains IDs, genders, ages, occupations, and zip-codes of all users. Table 5 is the rating database. This database provides user IDs, movie IDs, and timestamps of all ratings.

4.2. Comparison of the previous method and our revised method

Table 6 shows the top 10 movies recommended according to the previous method and the revised method. We input 'Drama, Romance' as the genre combination. Using the previous method, if one movie has more genres than other movies, that movie gets more points than the other movies if the average rating for these movies is the same. Using the revised method, the genre correlation values are averaged to avoid spurious results. The results are

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

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

Table 3
Genre numbers.

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

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

Table 6
The top 10 recommended movies by the previous method and our revised method.

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

presented in Table 6. Stunt Man, The (1980) received the highest score among all movies when the previous method was used (the genre combination of this movie is 'Action, Adventure, Comedy, Romance, Thriller'). The right side of Table 6 shows the results obtained using the revised method. Shadows of Forgotten Ancestors (1964) was the movie most highly recommended by the revised method. The genre combination of this movie is exactly the same as the input genre combination 'Drama, Romance'. City Lights (1931) is the fourth movie according to the revised method with a genre combination of 'Comedy, Drama, Romance'. This result indicates that the 'Comedy' genre is highly correlated with the genres of 'Drama' and 'Romance' (see Table 1). Therefore, City Lights (1931) is a valid recommendation.

4.3. Comparison of correlation matrices for different numbers of movies

To compare the recommendation results for different numbers of movies, we used 14 movie subsets containing different numbers of movies: 100, 200, 500, 1000, 2000, 3000, ⋯, 9000, 10,000, and 10,681. We constructed genre correlation matrices 100 times per subset. Because the same set of movies gives the same correlation matrix, we used randomly-selected movie sets. Because there were 14 subsets, we constructed a correlation matrix 1400 times. The results are shown in Figs. 4 and 5; the graphs shown have two y-axes, with one y-axis showing the correlation coefficient and the other axis showing the standard deviation. We omitted the values for 100, 200, 500, and 10,681 movies from these figures for better presentation. To calculate the correlations between the movies, we first extracted subsets from the total set. Next, we calculated correlations between the total set and each subset using the Pearson correlation coefficient.

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

Note that R in Eq. (6) is the average correlation coefficient. We repeated this 100 times and obtained the average correlation coefficient for each subset. The histogram in Fig. 5 shows the standard deviation of 100 correlation coefficients. The standard deviation for the 1000-movie subset was slightly lower than 0.01. For the other subsets, the standard deviations were much lower than 0.01. The histogram shows a sharp decrease between 1000 and 2000 movies.

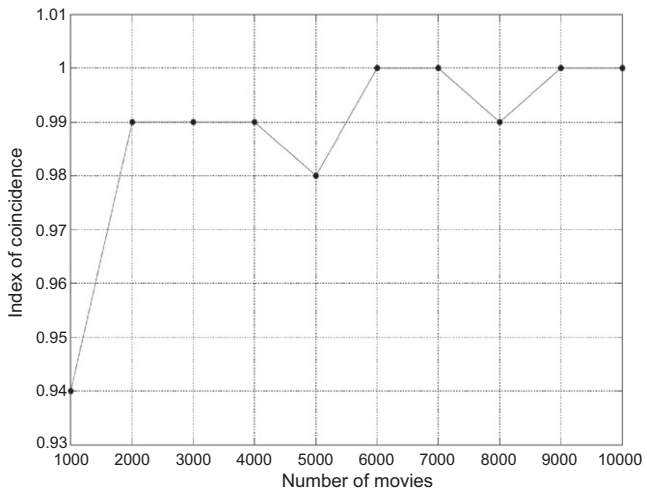

Fig. 4. Graph showing the index of coincidence between two recommendation results using 10 movies subsets and the total set of movies.

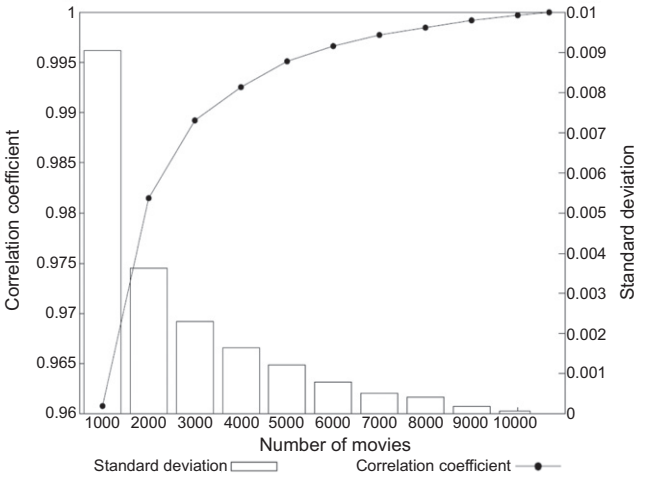

Fig. 5. Comparisons of the genre correlation matrices for different numbers of movies.

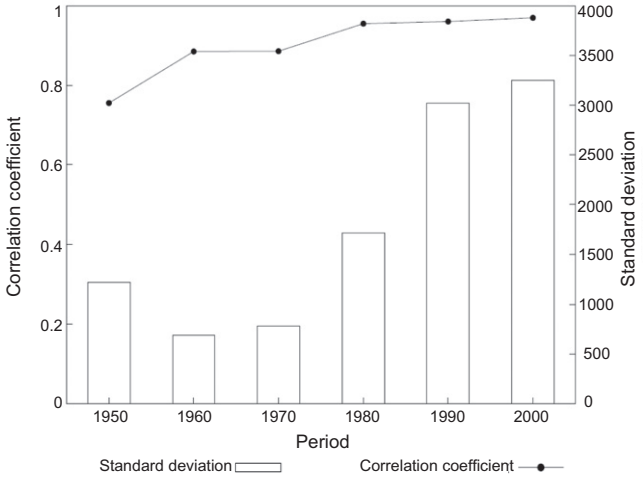

Fig. 6. Comparison of genre correlation matrices constructed with six distinct sets of movies corresponding to decades.

The correlation coefficients showed a reverse pattern to the standard deviations. The correlation coefficient increased dramatically between 1000 and 2000 movies.

We then applied the proposed algorithm in Section 3.1 to the subsets of movies to verify the usefulness of the algorithm. We compared the top 10 movies from each subset of movies and the top 10 movies from the total set 10 times. Fig. 4 shows the index of coincidence. This figure shows that when the number of movies was larger than 2000, the same results were obtained compared to the total set of movies. This implies that we can compute genre correlations with a certain number of movies (here, 2000) instead of the entire set of movies to provide movie recommendations.

4.4. Comparisons of correlation matrices according to decade

We compared correlation matrices for each decade in a similar way to that described above. We compared the correlation matrices constructed with the total set of movies. Fig. 6 shows the results of the comparison. For convenience, we summed all the movies before 1960 into the subset of the 1950s. We expected the 1950s subset of movies to have higher correlation coefficients than shown in Fig. 4, because this subset contained more movies than the 1960 and 1970 subsets. However, Fig. 6 shows the trend in genre combinations changed over time. The correlation coefficients showed a steady increase over time.

Fig. 7 shows the index of coincidence between the results from the total set of movies and from each subset of movies divided according to decade. This experiment was conducted in a similar fashion to that described above. The index of coincidence of the recommended movies is shown in Fig. 4. There were less coincident recommended movies for movies made before the 1960s than movies from other periods.

To confirm our expectation that each decade had a particular type of genre combination, we investigated the movie Ben-Hur (1959) in more detail. We calculated recommendation points of Ben-Hur from two genre correlation matrices constructed using two subsets of movies. The first subset consisted of movies made prior to the 1960s while the second subset consisted of movies produced in the 2000s. We input four genre combinations, namely 'Action, Adventure', 'Drama, Romance', 'Action, Drama', and 'Action, Adventure, Drama, Romance'. As we expected, higher recommendation points for Ben-Hur were obtained when we used the genre correlation matrix of movies produced before the 1960s as shown

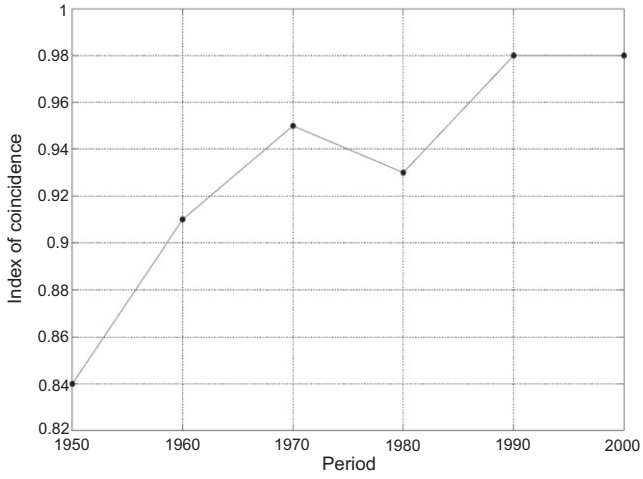

Fig. 7. Graph showing the index of coincidence between two recommendation results for each subset of movies divided according to decade and the total set of movies.

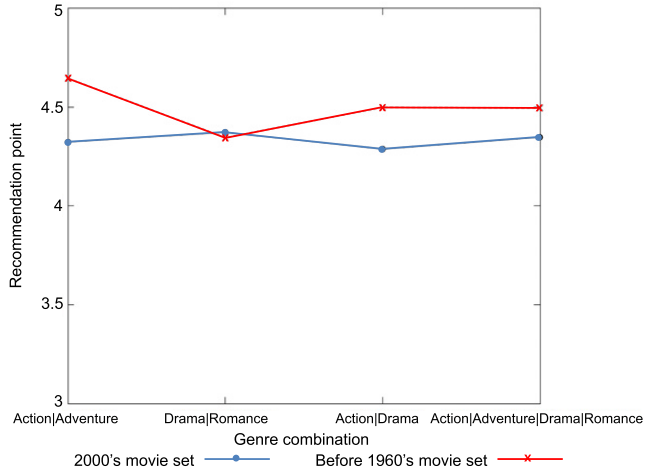

Fig. 8. A graph showing the relative recommendation points for Ben-Hur (1959) when we used movies made before the 1960s and movies made in the 2000s.

in Fig. 8. The difference between the two recommendation points when we used 'Action, Adventure' as the input genre combination was about 0.32. Note that in Table 6, the difference between the recommendation points of the second and tenth movie was about 0.32. Therefore, in cases like this, we can expect a change in rank of 8 steps at most. However, the recommendation points were very similar when the genre combination was 'Drama, Romance'. This is because of the commonness of this particular genre combination throughout the history of film. In other words, 'Drama, Romance' was a common genre combination before the 1960s and is still fairly common. Because the 'Drama, Romance' combination is not correlated with one particular period, there was no difference in the recommendation points calculated using the two different methods (Katz & Lazarsfeld, 1955).

5. Conclusions

Traditional recommendation systems require a certain amount of user preference data to determine groups of users and recommend items based on these groups. If there are not enough data, then the system becomes very unreliable because of the cold start problem. To solve this problem, various approaches have been suggested, one of which is a movie recommendation system based on category correlations. This latter approach is based on genre information. Movie genres are described by experts such as directors or producers. Thus, these genre descriptions are more reliable than genres defined by ordinary users. By using algorithms based on genre information, the cold start problem is not an issue (Billsus & Pazzani, 1998; Sarwar et al., 2001). To improve this approach, we proposed a method that constructs genre correlation, and we applied our proposed method to the GroupLens movie database. We analyzed genre correlations using specific criteria; different numbers of movie and decadal differences in movies. Our results indicate that by using our improved algorithms, reliable genre correlations can be constructed. Our results also indicate that more precise recommendations can be obtained using decade-based genre correlations. In the future, we aim to apply our approach to larger open databases and to conduct experiments with more users. We also plan to utilize Open APIs of content-sharing sites with category information, such as Yahoo Music or YouTube, in our future studies.

Acknowledgements

This research was supported by the Basic Science Research Program through NRF funded by MEST (2010–0009168) and the IT R&D program of MKE/IITA 2008-S-024–01.

References

Bell, R. M., & Koren, Y. (2007). Lessons from the netflix prize challenge. *SIGKDD Explorat., 9*, 75–79.

Billsus, D., & Pazzani, M. J. (1998). Learning collaborative information filters. In: *The 15th international conference on machine learning* (pp. 46–54).

Choi, S. M., & Han, Y. S. (2010). A content recommendation system based on category correlations. In: *The fifth international multi-conference on computing in the global information technology* (pp. 1257–1260).

Herlocker, J. L., Konstan, J., Borchers, A., & Riedl, J. (1999). An algorithm framework for performing collaborative filtering. In: *Proceedings of the 1999 conference on research and development in information retrieval* (pp. 230–237).

Honda, K., Notsu, A., & Ichihashi, H. (2009). Collaborative filtering by sequential extraction of user-item clusters based on structural balancing approach. In: *Fuzzy systems* (pp. 1540–1545).

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

Ishikawa, M., Géczy, P., Izumi, N., Morita, T., & Yamaguchi, T. (2007). Information diffusion approach to cold-start problem. In: *Web intelligence/IAT workshops* (pp. 129–132).

Katz, E., & Lazarsfeld, P. (1955). *Personal influence: the part of played by people in the flow of mass communications*. Free Press.

Popescul, A., Ungar, L. H., Pennock, D. M., & Lawrence, S. (2001). Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments. In: *Proceedings of the 17th conference in uncertainty in artificial intelligence* (pp. 437–444).

Sarwar, B. M., Karypis, G., Konstan, J. A., & Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. In: *Proceedings of the 10th international world wide web conference* (pp. 285–295).

Sarwar, B. M., Karypis, G., Konstan, J. A., & Riedl, J. (2000). Analysis of recommendation algorithms for e-commerce. In: *ACM conference on electronic commerce* (pp. 158–167).

Schein, A. I., Popescul, A., Ungar, L. H., & Pennock, D. M. (2002). Methods and metrics for cold-start recommendations. In: *The 25th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 253–260).

Scholz, B., Choi, S. M., Ko, S. K., Eom, H. S., & Han, Y. S. (2011). Analyzing category correlations for recommendation system. In: *Proceedings of the 5th international conferenece on ubiquitous information management and communication* (pp. 1:1–1:7) New York, NY, USA; ICUIMC '11, ACM.

Tang, T. Y., & McCalla, G. I. (2004). 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* (pp. 245–254).

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