Multimed Tools Appl (2016) 75:12843–12858
DOI 10.1007/s11042-016-3526-8

An algorithm for movie classification and recommendation using genre correlation

Tae-Gyu Hwang[1] · Chan-Soo Park[1] ·
Jeong-Hwa Hong[1] · Sung Kwon Kim[1]

Received: 12 March 2015 / Revised: 25 September 2015 / Accepted: 10 April 2016 /
Published online: 16 April 2016

Abstract Collaborative filtering (CF), a technique used by recommendation systems, predicts and recommends items (information, products or services) that the user might like. Amazon.com's recommender system is one of the most famous examples of CF. Recommendation systems are popular in both commercial and research sectors, and they are applied in a variety of applications such as movies, music, books, social connections and venues. In particular, movie recommendation systems produce personal recommendations for movies. Existing CF algorithms employed in movie recommendation systems predict the unknown rating of a given user for a movie using only the ratings (i.e., preferences) of other like-minded users who have seen the movie. In such approaches, there exist certain limits in improving the accuracy of recommendation systems. This paper proposes an algorithm for movie recommendation that exploits the genre of the movie to enhance the accuracy of rating predictions. The proposed algorithm 1) numerically measures the correlation between movie genres using movie rating information; 2) classifies movies using the genre correlations and generates a list of recommended movies for the target user with the classified movies; and finally 3) predicts the ratings of the movies in the list using traditional CF algorithms. The experimental results show that the proposed algorithm yields higher accuracy in movie rating predictions than existing movie recommendation algorithms.

✉ Sung Kwon Kim
skkim@cau.ac.kr

Tae-Gyu Hwang
tghwang@alg.cse.cau.ac.kr

Chan-Soo Park
cspark@alg.cse.cau.ac.kr

Jeong-Hwa Hong
jhhong@alg.cse.cau.ac.kr

[1] School of Computer Science and Engineering, Chung-Ang University, Dongjak-gu, Seoul 156-756, Republic of Korea

Keywords Collaborative filtering · Genre correlation · Movie classification · Movie recommendation

1 Introduction

With the advancement of the Internet, the number of e-commerce sites and the number of online customers and products have grown dramatically. As online markets become more competitive, online stores need targeted marketing tools that can increase sales, profits, and customer satisfaction. Recommendation systems serve these business objectives of e-commerce sites by producing personalized recommendations for customers based on their past history, purchase records and interests. In recent years, the widespread use of smart devices and social networks has enabled e-commerce sites to collect a large amount of information on users' behaviors, activities or preferences. Naturally, new technologies that can efficiently analyze and use such data in recommender systems are required. In addition, recommendation technologies are increasingly linked to other areas of expertise to improve the performance, coverage, and accuracy of recommender systems [9].

Collaborative filtering (CF), a technique used by recommendation systems, predicts and recommends items (information, products or services) that the user might like. Amazon.com's recommender system is one of the most famous examples of CF. Recommendation systems are popular in both commercial and research sectors, and they are applied in a variety of applications such as movies, music, books, social connections and venues. In particular, movie recommendation systems produce personal recommendations for movies.

CF-based movie recommendations predict a likeness score or a list of top-N recommended movies for a given user based on ratings (preference scores) from many users. Since they use only available ratings that are explicitly given by users, their prediction accuracy faces certain limits [1, 3, 7]. A number of works that employ other item attributes have been conducted to achieve more precise recommendations [4, 6, 11–13]. This paper proposes an algorithm for movie recommendation systems that utilizes the genres of the movies as well as the ratings of the movies, to increase rating prediction accuracies.

Each movie has various attributes such as name, genre, starring, director, theme or topic, mood, setting, etc. The movie genres (action, comedy, romance, etc.) can be categorized in several ways, but usually the knowledge of the expert is used to assign a genre to a movie. In general, one can find certain similarities in the movies of the same genre but there are no concrete and quantified criteria for genre classification and assignments [4]. In addition, a movie can have more than one associated genre, so there is no way to computationally pinpoint a single representative genre for the movie when this information is not explicitly given.

Considering a combination of the genres associated with a movie gives more insights than considering each of them individually. For example, it makes more sense that a movie's genre is action and crime than children and crime. The fact that some genre combinations are more sensible than others indicates that the movie genres are correlated [4, 13]. Viewers often can guess the story, mood, and setting of a movie by its genre(s), so the movie's genre influences viewers' interest in the movie and eventually the decision whether or not to watch it. Most people have at least one movie genre that they prefer. It is presumed that users who prefer action would likely enjoy the recommended action movies than users with other preferred genre choices [4].

Existing CF algorithms for movie recommendation systems make recommendations using user-given ratings of the movies without taking into account their genre or genre correlation [2, 5, 8, 10]. The algorithm proposed in this paper calculates the correlation between genres using rated movie scores, and performs the classification of movies and the prediction of a list of recommended movies for a target user based on the calculated genre correlations.

2 Related work

In this paper, the prediction accuracy of our proposed movie recommendation algorithm is examined in connection with previous works: the user-based collaborative filtering in [2], and the item-based collaborative filtering in [10].

2.1 User-based collaborative filtering

The user-based CF calculates the similarity of users based on user ratings in order to find a set of users, known as neighbors, whose opinions are historically similar to the target user. It then combines the ratings of the neighbors to predict a rating or top-N recommendation for the target user. User similarities are measured using Pearson correlation coefficient, and rating predictions are performed using the preference prediction formula shown below. I_u and I_v, respectively, are the set of items which are rated by users u and v, and thus $I_u \cap I_v$ is the set of co-rated by user u and v. $r_{u,i}$ denotes the rating of item i rated by user u, $\overline{r}_u$ is the average of the ratings of the items rated by user u. The set N_u contains k nearest neighbors of user u.

2.1.1 User similarity computation

$$sim(u,v) = \frac{\sum_{i \in I_u \bigcap I_v} \left(r_{u,i} - \overline{r}_u\right)\left(r_{v,i} - \overline{r}_v\right)}{\sqrt{\sum_{i \in I_u \bigcap I_v} \left(r_{u,i} - \overline{r}_u\right)^2} \sqrt{\sum_{i \in I_u \bigcap I_v} \left(r_{v,i} - \overline{r}_v\right)^2}} \tag{1}$$

2.1.2 Preference prediction

$$P_{u,i} = \overline{r}_u + \frac{\sum_{v \in N_u}^{k} sim(u,v) \times \left(r_{v,i} - \overline{r}_v\right)}{\sum_{v \in N_u}^{k} |sim(u,v)|} \tag{2}$$

2.2 Item-based collaborative filtering

The item-based CF calculates the similarity between items using user-given ratings. Recommendations for a user are computed by finding items that are similar to other items the user has liked. Once the most similar items are found, the rating prediction is computed by taking a weighted average of the target user's ratings on these similar items. Item similarities are calculated using Pearson's correlation coefficient, and rating predictions are performed using the weighted sum equation below [9, 10].

U_i and U_j, respectively, are the sets of uses who rated items i and j, and thus $U_i \cap U_j$ is the set of users who rated both items i and j. I_u is the set of items rated by user u, and $\overline{r}_i$ is the average of the ratings given by users to item i.

2.2.1 Item similarity computation

$$sim(i,j) = \frac{\sum_{u \in U_i \bigcap U_j} \left(r_{u,i} - \overline{r}_i\right)\left(r_{u,j} - \overline{r}_j\right)}{\sqrt{\sum_{u \in U_i \bigcap U_j} \left(r_{u,i} - \overline{r}_i\right)^2} \sqrt{\sum_{u \in U_i \bigcap U_j} \left(r_{u,j} - \overline{r}_j\right)^2}} \tag{3}$$

2.2.2 Preference prediction

$$P_{u,i} = \frac{\sum_{j \in I_u} sim(i,j) \times r_{u,j}}{\sum_{j \in I_u} |sim(i,j)|} \tag{4}$$

3 Proposed method

The proposed algorithm has the pre-processing process that measures the correlation between movie genres using rated scores and uses the measured correlations to categorize a movie into a single genre cluster. When a recommendation event occurs (i.e., a user requests for a movie recommendation), the proposed algorithm calculates the genre preferred by the target user, identifies movies that belong to the target user's preferred genre and its similar genres (i.e., genres highly correlated to the target user's preferred genre), and creates a recommendation list consisting of the identified movies. Finally, the proposed algorithm predicts the ratings of the movies in the list and recommends them to the target user.

3.1 Genre correlation measurement

The genre of a movie is generally assigned by expert's subjective judgment and it is hard to quantify the criteria for genre assignments. The proposed algorithm uses movie's rated scores to calculate the correlation between movie genres. The correlation between genres a and b, denoted as *genre_corr*(a, b), is calculated using the formula below.

$$genre\_corr(a,b) = \omega \times genre\_prob(a,b) + (1 - \omega) \times genre\_weight(a,b) \tag{5}$$

Note the in (5), the genre probability denoted by *genre_prob*(a, b) and the genre weight denoted by *genre_weight*(a, b) equally contribute to *genre_corr*(a, b). The *genre_corr* is the correlation between movie genres. Since *genre_corr*(a, b) and *genre_corr*(b, a) may be different, the correlation matrix is asymmetric. It is calculated by using *genre_weight* and *genre_prob*. The *genre_weight* is calculated by using the Pearson correlation coefficient, so the weight matrix is symmetric. The *genre_prob* is the co-occurrence probability of movie genres. The probability matrix is asymmetric. In Eq. (5), is calculated to reflect each characteristic of *genre_weight* and *genre_prob* in the same ratio ($\omega = 0.5$). As a consequence, the correlation matrix is asymmetric (Figs. 1 and 2).

3.1.1 Genre probability

The interest of an action movie lover toward adventure movies is not necessarily equivalent to the interest of an adventure movie lover toward action movies. Thus, the correlation between genres needs to be calculated asymmetrically [4, 5]. In the proposed algorithm, the conditional probability is used to compute the genre probability.

$$genre\_prob(a,b) = P\left(b\middle|a\right) = \frac{P\left(a\bigcap b\right)}{P(a)} = \frac{\left|I_{a\bigcap b}\right|}{|I_a|}, \tag{6}$$

where I_a is the set of movies belonging to genre a, and $I_{a\cap b}$ is the set of movies belonging to both genres a and b.

3.1.2 Genre weight

The genre weight equation, a variant of Pearson's correlation coefficient, calculates the rating correlation of the movies belonging to genre a and b. In the equation below, $pnt_i(a,b)$ denotes the penalty of movie i, $s_{*,i}$ denotes the set of ratings given by the users who rated movie i, and $\bar{s}_a$ is the average of the ratings of the movies belonging to genre a.

$$genre\_weight(a,b) = \frac{\sum_{i\in I_{a\bigcap b}} pnt_i(a,b)\left(s_{*,i} - \bar{s}_a\right) \times pnt_i(a,b)\left(s_{*,i} - \bar{s}_b\right)}{\sqrt{\sum_{i\in I_{a\bigcap b}} \left(pnt_i(a,b)\left(s_{*,i} - \bar{s}_a\right)\right)^2}\sqrt{\sum_{i\in I_{a\bigcap b}} \left(pnt_i(a,b)\left(s_{*,i} - \bar{s}_b\right)\right)^2}} \tag{7}$$

As mentioned earlier, there can be more than one genre associated with a single movie. The smaller the number of genres associated with a movie is, the higher the correlation between associated genres is. Hence, pnt_i is given differentially according to the number of genres to which a movie belongs. The genre weight equation has two target genres (genres a and b), so the numerator of the pnt_i formula is 2 and the denominator is the number of genres to which movie i belongs. G_i is the set of genres to which movie i belongs.

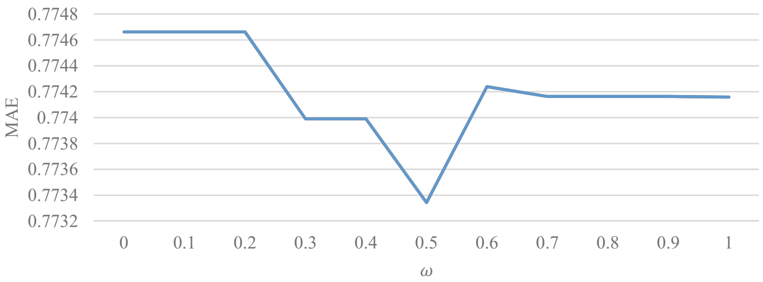

Fig. 1 UBGC prediction accuracy of MAE for each ω values

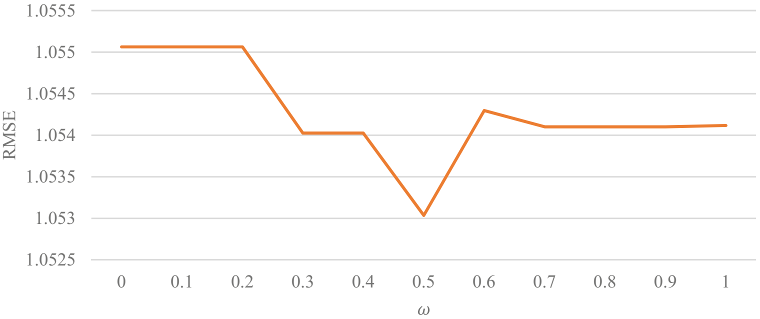

Fig. 2 UBGC prediction accuracy of RMSE for each ω values

$$pnt_i(a,b) = \frac{2}{|G_i|} \tag{8}$$

$s_{u,i}$ denotes the bias-removed rating of movie i. It is calculated by subtracting the user bias, the movie bias, and the average of all ratings from the rating of movie i given by user u [1, 8]. $s_{u,i}$ is an element in the formula that computes the correlation between movie genres.

$$s_{u,i} = r_{u,i} - \mu - b_u - b_i \tag{9}$$

The average rating of genre a denoted by $\overline{s}_{\mathrm{g_a}}$ is calculated by subtracting the average bias of the users who rated the movies in genre a $\left(\overline{b}_*\right)$, the average bias of the movies in genre a $\left(\overline{b}_{\mathrm{a}}\right)$, and the average of all ratings (μ) from the average rating of the movies in genre a ($\overline{r}_a$). $\overline{s}_a$ corresponds to the element of average rating in Pearson's correlation coefficient, i.e., the average rating of a user in the user-based CF and the average rating of an item in the item-based CF [5, 10]. The correlation between genres is computed using $\overline{s}_a$ and $s_{u,i}$. $\overline{b}_*$ is the average bias of the users who rated the movies in genre a.

$$\overline{s}_a = \overline{r}_a - \mu - \overline{b}_* - \overline{b}_a \tag{10}$$

The average bias of genre a is denoted by $\overline{b}_{\mathrm{a}}$. This is calculated by subtracting the average bias of the users who rated the movies in genre a ($\overline{b}(U_a)$), and the average bias of the movies in genre a ($\overline{b}(I_a)$) from the average rating of the movies in genre a ($\overline{r}(I_a)$).

$$\overline{b}_a = \overline{r}(I_a) - \mu - \overline{b}(U_a) - \overline{b}(I_a) \tag{11}$$

3.2 Movie classification

The classification of movies is performed using associated movie genres and genre correlations [4]. In the MovieLens 100 k dataset, 18 genres were identified and thus a 18 × 18 matrix was created. Note that the movies with 'unknown' genres in the dataset were excluded. The genres of each movie are correlated to classify the movie into a single corresponding class in the matrix that has the highest correlation score.

In the MovieLens 10 M dataset, the proposed algorithm computed the correlations of seven genres, as shown in the matrix below. Suppose that three movies i_1, i_2,and i_3 are classified using these genre correlations. The matrix rows represent the genres of the movies and the columns represent the user's preferred movie genres (Table 1).

Table 1 Example of the calculated several genre correlations

	g_1	g_2	g_3	g_4	g_5	g_6	g_7
g_1	1	0.6533	0.0122	0.5154	0.6063	0.6073	0.0003
g_2	0.7204	1	0.5551	0.6065	0.6375	0.0253	0.5025
g_3	0.0629	0.6975	1	0.8258	0.7132	0.0087	0.5032
g_4	0.5440	0.7075	0.6760	1	0.7411	0.0075	0.4985
g_5	0.5421	0.5380	0.5164	0.5333	1	0.5366	0.5042
g_6	0.6415	0.0232	0.0022	0.0035	0.6250	1	0.4989
g_7	0.0010	0.5058	0.5018	0.4987	0.5330	0.5012	1

To classify movie i_1, the number of cases that it belongs to action, adventure and crime are identified, and the correlation values of the identified genre pairs are compared. Movie i_1 is classified into the pair of genres (the class in the matrix) that has the highest correlation value. Note that the correlation of two identical genres is not considered when the movie belongs to more than one genre (Table 2).

The seven movie genres are denoted by g_1 (Action), g_2 (Adventure), g_3 (Animation), g_4 (Children), g_5 (Comedy), g_6 (Crime), and g_7 (Documentary), respectively.

As shown in the table above, movie i_1 belongs to three different genres. Once the pairs of identical genres are excluded, there are 6 possible cases. Among the 6 cases, g_1, g_2 has the highest correlation value. Hence, movie i_1 is categorized into the g_1, g_2 class. Similarly, movie i_2 is classified into the g_4, g_3 class that has the highest correlation score. Movie i_3 belongs to a single genre so it is categorized into the g_7, g_7 class.

3.3 Movie recommendation

After the pre-processing process described in Sections 3.1 and 3.2 is carried out, the proposed algorithm performs the movie recommendation process that produces a list of recommended movies and predicts the ratings of the movies in the list.

If the target user prefers genre g_1, the genres to be recommended are chosen in the order of g_1, g_2, g_6, g_5, g_4, and g_3, and the movies belonging to the chosen genre(s) are included in the recommendation list. For example, if the target user's one favorite genre (g_1) and two similar genres (g_1, g_2) are chosen, the moves classified in $c_{g_1,\, g_1}$ and c_{g_1, g_2} are recommended to the target user.

3.3.1 Creation of a list of recommended movies

Using the ratings given by the target user, the frequency of ratings of the identified 18 genres are calculated, and the top-*N* frequently rated genres are chosen as the target user's preferred

Table 2 Example of the movie classification using genre correlation

$i_1 \rightarrow g_1, g_2, g_6$	$i_2 \rightarrow g_3, g_4$	$i_3 \rightarrow g_7$
1) $g_1, g_2 = 0.720412$ 2) $g_1, g_6 = 0.641596$ 3) $g_2, g_1 = 0.653353$ 4) $g_2, g_6 = 0.023255$ 5) $g_6, g_1 = 0.607321$ 6) $g_6, g_2 = 0.025365$	1) $g_3, g_4 = 0.676008$ 2) $g_4, g_3 = 0.825847$	$g_7, g_7 = 1.0$

genres. Here, N is equivalent to $UPGC$ in the equation below. The movies in the target user's preferred genres and genres similar to the target user's preferred genres are included in a movie recommendation list for the target user. The number of similar genres is denoted by SGC. The ratings of the movies in the recommendation list are predicted using item-based CF algorithm and user-based CF algorithm [5, 10].

$$RecommendedList_u = \bigcup_{upg \in UPG_u}^{UPGC} \bigcup_{sg \in SG_{upg}}^{SGC} c_{upg,sg} \tag{12}$$

3.3.2 Prediction of movie ratings

The ratings of the movies in the recommendation list are predicted using classical user-based and item-based CF algorithms. In user-based CF, the preference prediction equation is used to predict the ratings (preference scores) that the target user would give to the recommended movies. In item-based CF, the weighted sum equation is used to perform movie rating predictions.

4 Experimental evaluation

4.1 Datasets description

The movie datasets available at MovieLens were used to assess the prediction accuracy of the proposed algorithm. The MovieLens 10 M dataset was used to compute movie genre correlations and the MovieLens 100 k dataset was used to make movie recommendations (Table 3).

In the MovieLens 100 k dataset, 80 % of the data is the training set and 20 % is the test set (5-fold cross validation is used). Using the data in this dataset, the accuracy of the movie ratings predicted by the proposed and conventional algorithms was evaluated. Five pairs of the training set and the test set were experimented to measure rating prediction accuracy, and the average results of these five experiments were used for comparison.

4.2 Evaluation metrics

The accuracy of the rating predictions of the proposed and conventional algorithms was measured using the Mean Absolute Error and the Root Mean Squared Error [8, 10].

Table 3 This table is MovieLens dataset description

Dataset	MovieLens 100 k	MovieLens 10 M
Users	943	71,567
Movies	1682	10,681
Ratings	100,000	10,000,054
Genres	18 (movies with 'unknown' genres excluded)	18 (unknown, IMDB excluded)

4.2.1 MAE

$$MAE = \frac{\sum_{i \in N} |p_i - q_i|}{|N|} \tag{13}$$

4.2.2 RMSE

$$RMSE = \sqrt{\frac{\sum_{i \in N} (p_i - q_i)^2}{|N|}} \tag{14}$$

In (13) and (14), p_i is the real rating of movie i in the test-set, and q_i is the predicted rating of movie i in the test-set, and N is the set of movies in the test-set for which rating predictions are made.

4.3 Evaluation

The experiments were executed for the user-based and item-based CF methods in a separate manner. In each method, the classical CF algorithm, and the proposed algorithm using genre correlations were experimented and their prediction accuracies were compared.

4.3.1 User-based CF (UBCF)

Figures 3 and 4 shows the accuracy of the movie rating predictions produced by the UBCF algorithm. k denotes the parameter of the k-nearest neighbor algorithm. The prediction accuracy was measured in terms of MAE and RMSE by varying k, from 10 to 942. In both MAE and RMSE, the optimal result (the most accurate prediction) is obtained when $k = 450$.

4.3.2 User-based CF using genre correlations (UBGC)

The parameters used in the proposed algorithm are the number of nearest neighbors denoted by k, the number of target user's preferred genres denoted by *UPGC*, and the number of genres similar to the target user's preferred genres denoted by *SGC*. To find the optimal performance,

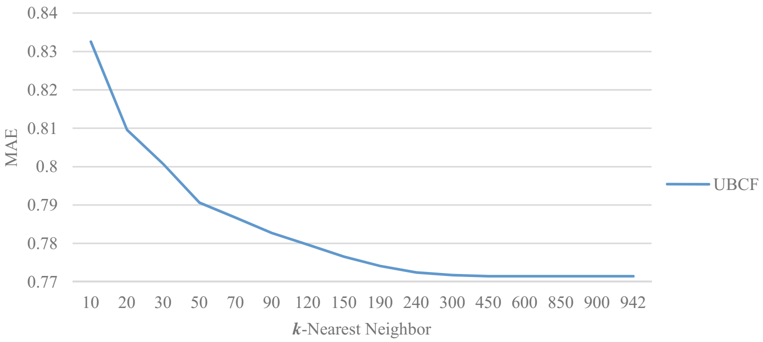

Fig. 3 UBCF prediction accuracy of MAE for each ***k*** values

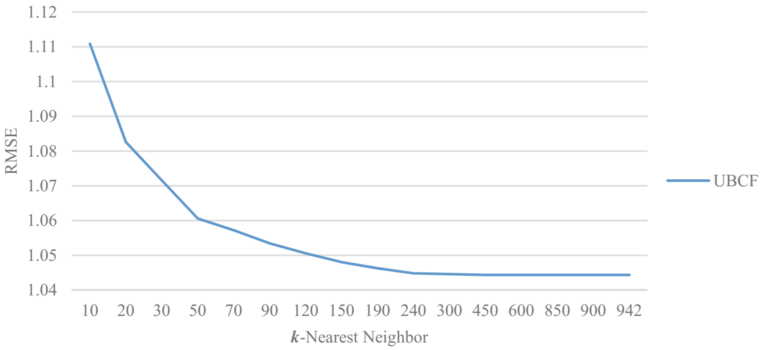

Fig. 4 UBCF prediction accuracy of RMSE for each ***k*** values

changes in the accuracy of the proposed algorithm were examined by varying the value of the parameters. For parameter *k*, the value was gradually increased from 10 to 900 (Figs. 5 and 6).

For a single *k* value, 81 results were produced by varying *UPGC* and *SGC* from 1 through 9, and the median and maximum values of the results were identified. In the conducted experiments, the number of recommended movies was small when *UPGC* and *SGC* were small, so the median values rather than the minimum values were used in MAE and RMSE measures. In addition, the maximum values were used in MAE and RMSE measures to find *k* that has a small range of deviations.

The prediction accuracy was best when *k* = 450. Hence, parameter *k* is fixed to 450 and the accuracy of the rating predictions was examined by varying *UPGC* and *SGC*. The results are presented in the table below (Tables 4 and 5).

The optimum performance of the UBGC was determined by considering the fact that the number of recommended movies is small when UPGC and SGC are small. The UBGC produces the optimum prediction quality (MAE is 0.74228 and RMSE is 1.01907) when UPGC = 3 and SGC = 1. With these values, the proposed algorithm employing user-based CF was compared to the conventional user-based CF algorithms, UBCF. The table below shows this comparison (Table 6).

4.3.3 Classical item-based CF (IBCF)

The table below shows the accuracy of movie ratings predicted by the IBCF algorithm (Table 7).

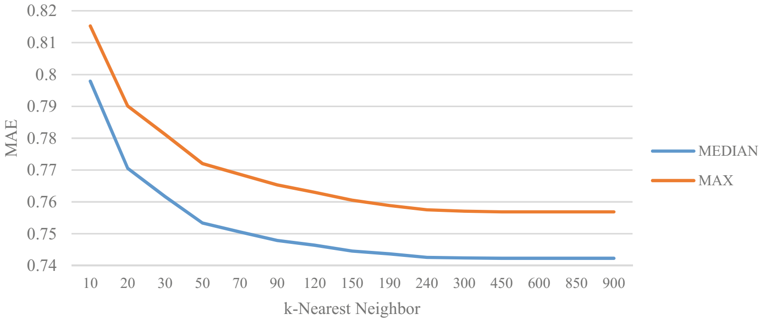

Fig. 5 Find the optimal parameter ***k*** values (Median)

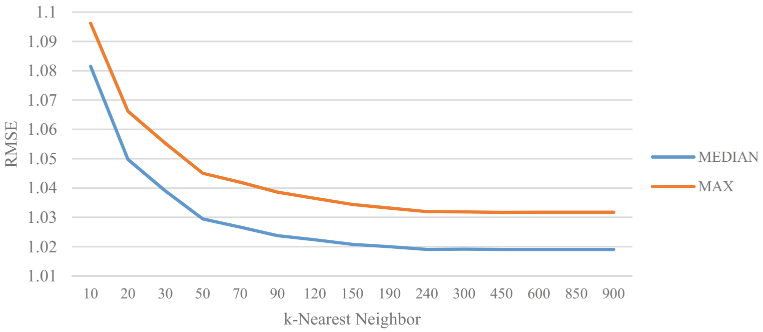

Fig. 6 Find the optimal parameter *k* values (Median)

4.3.4 Item-based CF using genre correlations (IBGC)

Like UBGC, the IBGC produces 81 results by varying *UPGC* and *SGC* from 1 through 9. A difference from the UBGC is that the IBGC achieves better prediction accuracies when the target user's preferred genre is fixed and its similar genres are increased to more than 2 (Tables 8 and 9).

In the IBGC, the lowest MAE score is 0.76620 when $UPGC=5$ and $SGC=2$ and the lowest RMSE score is 1.04893 when $UPGC=1$ and $SGC=5$. Giving first priority to the number of recommended movies in determining an optimal performance, the case of $UPGC=5$ and $SGC=2$ was chosen and compared with the conventional item-based CF algorithm (IBCF), as shown below (Table 10).

5 Conclusions

The proposed algorithm assessed using the MovieLens datasets yields better results than the conventional movie recommendation algorithms although their differences in performance (prediction accuracy) are not very large. The proposed algorithm quantifies the correlation between movie genres using user-given ratings and uses the genre correlations to classify the

Table 4 Find the optimal parameters UPGC and SGC (MAE)

MAE $k=450$		UPGC								
		1	2	3	4	5	6	7	8	9
SGC	1	0.74239	0.74269	0.74228	0.74363	0.74407	0.74454	0.74503	0.74568	0.74562
	2	0.74458	0.74427	0.74307	0.74418	0.74471	0.74526	0.74563	0.74618	0.74631
	3	0.74321	0.74549	0.74484	0.74518	0.74526	0.74525	0.74560	0.74602	0.74606
	4	0.74428	0.74559	0.74511	0.74567	0.74568	0.74568	0.74591	0.74625	0.74623
	5	0.74613	0.74743	0.74584	0.74590	0.74590	0.74572	0.74591	0.74628	0.74622
	6	0.74671	0.74776	0.74640	0.74674	0.74698	0.74700	0.74683	0.74694	0.74694
	7	0.74521	0.74625	0.74561	0.74672	0.74717	0.74732	0.74720	0.74721	0.74715
	8	0.74516	0.74605	0.74510	0.74611	0.74659	0.74706	0.74718	0.74720	0.74717
	9	0.74380	0.74526	0.74544	0.74630	0.74659	0.74703	0.74716	0.74718	0.74717

Table 5 Find the optimal parameters UPGC and SGC (RMSE)

RMSE $k = 450$		UPGC								
		1	2	3	4	5	6	7	8	9
SGC	1	1.01705	1.01886	1.01907	1.02083	1.02144	1.02186	1.02247	1.02296	1.02300
	2	1.01990	1.02057	1.01980	1.02136	1.02203	1.02261	1.02304	1.02349	1.02375
	3	1.01999	1.02237	1.02189	1.02235	1.02262	1.02259	1.02296	1.02330	1.02344
	4	1.02131	1.02262	1.02228	1.02281	1.02301	1.02292	1.02322	1.02355	1.02359
	5	1.02341	1.02473	1.02320	1.02314	1.02323	1.02298	1.02320	1.02356	1.02358
	6	1.02392	1.02520	1.02395	1.02412	1.02451	1.02446	1.02434	1.02448	1.02446
	7	1.02256	1.02345	1.02304	1.02410	1.02487	1.02491	1.02477	1.02478	1.02472
	8	1.02217	1.02345	1.02258	1.02343	1.02432	1.02465	1.02474	1.02477	1.02474
	9	1.02139	1.02288	1.02291	1.02367	1.02432	1.02460	1.02470	1.02474	1.02473

Table 6 Comparison of MAE and RMSE

Algorithm	k	MAE	RMSE	Etc.
UBCF	450	0.77143	1.04435	
UBGC	450	0.74228	1.01907	UPGC = 3, SGC = 1, $\omega = 0.5$

movies in a dataset before the movie recommendation process begins. This pre-processing enables the proposed algorithm to make more accurate rating predictions, thus producing high-quality movie recommendations. In order for genre correlation information to be reliable and useful for movie recommendations, further work and verifications are needed but the approach proposed in this paper shows a way to exploit item's genre information and it can be extended to be applicable in other domains. In the experiments, it was observed that the proposed algorithm has lower prediction accuracies than the conventional algorithms when the weights are applied to rating predictions. The reasons and solution for this problem should be studied in the future. Since the movie genre is one of the attributes of the movie, the authors expected that the proposed algorithm would achieve better results when item-based CF is used. In the proposed algorithm using item-based CF, increasing SGC while fixing UPGC made prediction accuracy gradually increase. In the proposed algorithm using user-based CF, increasing UPGC rather than SGC led to higher prediction accuracies. When UPGC and SGC were small, the number of recommended movies were small but the accuracy of predictions was better. In the proposed algorithm, the value for SGC should be determined based on movie characteristics when item-based CF is used. Similarly, the value for UPGC should be determined based on user characteristics when user-based CF is used.

Table 7 IBCF prediction accuracy of MAE and RMSE

Algorithm	MAE	RMSE
Item-based collaborative filtering	0.7939601	1.092543

Table 8 Find the optimal parameters UPGC and SGC (MAE)

MAE		UPGC								
		1	2	3	4	5	6	7	8	9
SGC	1	0.92857	0.94551	0.90286	0.95003	0.95587	0.94277	0.94936	0.92756	0.91822
	2	0.83330	0.79149	0.80814	0.77549	0.76620	0.78240	0.78478	0.78736	0.78682
	3	0.78921	0.79267	0.81725	0.79230	0.78297	0.79452	0.79414	0.79694	0.79677
	4	0.77974	0.81002	0.81257	0.79066	0.78703	0.78996	0.79489	0.79603	0.79378
	5	0.76911	0.79713	0.80860	0.78630	0.78119	0.78999	0.79319	0.79372	0.79240
	6	0.78672	0.78903	0.80509	0.78782	0.79355	0.79258	0.79345	0.79423	0.79324
	7	0.77046	0.78773	0.79996	0.78248	0.78040	0.79075	0.79249	0.79634	0.79437
	8	0.76922	0.79091	0.80784	0.78442	0.78455	0.79090	0.79191	0.79283	0.79466
	9	0.77879	0.79489	0.80591	0.78879	0.78580	0.78809	0.79154	0.79548	0.79356

Table 9 Find the optimal parameters UPGC and SGC (RMSE)

RMSE		UPGC								
		1	2	3	4	5	6	7	8	9
SGC	1	1.03452	1.24123	1.23571	1.30348	1.30562	1.29383	1.29366	1.27649	1.26525
	2	1.13090	1.07352	1.09534	1.07125	1.06507	1.07859	1.07903	1.08008	1.07819
	3	1.06372	1.07108	1.10721	1.09040	1.07978	1.08923	1.08484	1.08686	1.08600
	4	1.06000	1.09604	1.10487	1.08780	1.08347	1.08408	1.08645	1.08710	1.08391
	5	1.04893	1.08497	1.10240	1.08501	1.07884	1.08444	1.08597	1.08547	1.08304
	6	1.07662	1.07507	1.10049	1.08587	1.09193	1.08674	1.08597	1.08598	1.08395
	7	1.05386	1.07335	1.09414	1.08344	1.07709	1.08565	1.08578	1.08862	1.08489
	8	1.04951	1.07713	1.10012	1.08310	1.08178	1.08525	1.08435	1.08473	1.08572
	9	1.06061	1.07969	1.10083	1.08643	1.08275	1.08266	1.08481	1.08782	1.08476

6 Future work

In this paper, the experiments were conducted with two MovieLens datasets, the MovieLens 10 M dataset for genre correlation calculation and the MovieLens 100 k dataset for movie rating predictions.

The MovieLens 10 M dataset was used for genre correlation measurement because the MovieLens 100 k dataset could not be used due to data sparsity - a sparse genre correlation

Table 10 Comparison of MAE and RMSE

Algorithm	MAE	RMSE	Etc.
IBCF	0.79340	1.09254	
IBGC	0.76620	1.06507	UPGC = 5, SGC = 2

matrix is produced. In order to get consistent and reliable results, it is important that both genre correlation calculation and movie recommendation are carried out in a same dataset, either the MovieLens 100 k dataset or the MovieLens 10 M dataset. Thus, a way to resolve this problem needs to be developed.

The proposed algorithm using used-based CF should address the problem of not being able to make rating predictions due to the unavailability of movie ratings from some nearest neighbors, and it is also less efficient than the conventional CF algorithms in term of computational cost. The proposed algorithm using item-based CF is able to prepare a recommendation list in advance via the preprocessing process, thereby producing much faster recommendations. Since user-based and item-based CF suffers from the fundamental data sparsity problem, another means of recommending movies that does not rely on rating prediction is needed. One way to address this issue is recommending the movies in user's preferred genres and/or genres similar to the user's preferred genres.

Acknowledgments This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (NRF-2015R1D1A1A01059937).

References

1. Bell RM, Koren Y, Volinsky C (2008) The Bellkor 2008 solution to the Netflix prize. Stat Res Dep AT&T Res
2. Breese JS, Heckerman D, Kadie C (1998) Empirical analysis of predictive algorithms for collaborative filtering. Proc 40th Conf Uncertain AI 1:43–52
3. Cacheda F, Carneiro V, Fernández D, Formoso V (2011) Comparison of collaborative filtering algorithms: limitations of current techniques and proposals for scalable, high-performance recommender systems. ACM Trans Web 5:1–33. doi:10.1145/1921593
4. Choi SM, Ko SK, Han YS (2012) A movie recommendation algorithm based on genre correlations. Expert Syst Appl 39:8079–8085. doi:10.1016/j.eswa.2012.01.132
5. Ding Y, Li X (2005) Time weight collaborative filtering. Proc 14th ACM Int Conf Inf Knowl Manag. doi:10.1145/1099554.1099689
6. Edson B, Santos Jr, Rudinei Goularte, Marcelo G, Manzato G (2014) Personalized collaborative filtering: a neighborhood model based on contextual constraints. Proc 29th Annu ACM Symp Appl Comput 1:919–924. doi:10.1145/2554850.2555017
7. Herlocker JL, Konstan JA, Terveen LG, Riedl JT (2004) Evaluating collaborative filtering recommender systems. ACM Trans Inf Syst 22:5–53. doi:10.1145/963770.963772
8. Koren Y, Bell R, Volinsky C (2009) Matrix factorization techniques for recommender systems. Computer 42:30–37. doi:10.1109/MC.2009.263
9. Linden G, Smith B, York J (2003) Amazon.com recommendations: item-to-item collaborative filtering. Internet Comput IEEE 7:76–80. doi:10.1109/MIC.2003.1167344
10. Sarwar B, Karypis G, Konstan J, Riedl J (2001) Item-based collaborative filtering recommendation algorithms. Proc 10th Int Conf WWW 1:285–295. doi:10.1145/371920.372071
11. Soares M, Viana P (2014) Tuning metadata for better movie content-based recommendation systems. Multimed Tools Appl. doi:10.1007/s11042-014-1950-1
12. Wu IC, Niu YF (2013) Integrating the anchoring process with preference stability for interactive movie recommendations. In: Yamamoto S (ed) Human interface and the management of information. Springer, Berlin, pp 639–648
13. Zheng Q, Horace HSIP (2012) Customizable surprising recommendation based on the tradeoff between genre difference and genre similarity. Web Intell Intell Agent Technol 1:702–709. doi:10.1109/WI-IAT.2012.70

Tae-Gyu Hwang He is currently a Ph.D. degree course student in Computer Sciences and Engineering from Chung-Ang University, Seoul, Korea. His areas of research interest are algorithms and big data analysis.

Chan-Soo Park He is currently a Ph.D. degree course student in Computer Sciences and Engineering from Chung-Ang University, Seoul, Korea. His areas of research interest are algorithms and big data analysis.

Jeong-Hwa Hong He is currently a master's degree course student in Computer Sciences and Engineering, Chung-Ang University, Seoul, Korea. His areas of research interest are algorithms and big data analysis.

Sung Kwon Kim He received his bachelor's degree from Seoul Nation University, Seoul, Korea, his master's degree from Korea Advanced Institute of Science and Technology (KAIST), Korea, and his Ph.D. degree from University of Washington, Seattle, U.S.A. He is currently a professor at Division of Computer Science and Engineering, Chung-Ang University, Seoul, Korea.