A Recommender System for Political Information Filtering

Kevin Lim[1], Chunghwan Kim[1], Gangsan Kim[1], and Hyebong Choi[2(✉)]

[1] School of Computer Science and Electronic Engineering, Handong Global University, Pohang, South Korea
klim6263@gmail.com, obparkl@naver.com, 21000048@handong.edu

[2] School of Creative Convergence Education, Handong Global University, Pohang, South Korea
hbchoi@handong.edu

Abstract. Recommender system has been widely used and showcased its successful stories in e-business area for the last decade. It assists in making profits within a lot of companies by recommending their products that the customers would be interested in. Compared to many successful stories in e-business and industries, however, a recommender system has not been fully exploited in non-profit activities where people need information that is unbiased, accurate, up-to-date and mostly relevant to their interest, especially in politics. Even though choosing a right candidate with appropriate and accurate information is required to voters, it is not easy for them to keep up with the political issues due to the massive amounts of online media and its speed. To address these issues, we suggest a politician recommender system by using two widely used filtering: collaborative filtering and content-based filtering.

In order to build the recommendation system, we first collect public profile of current congress members in Korea and people's preference ratings to these politicians. These data are preprocessed and used in filtering methods to recommend politicians that a user would be favorable for. We compare the experimental results, and combine the two filtering whether the hybrid approach shows better performance than two individual methods. We anticipate this saves people's time and effort to obtain information to support their decision and makes people actively participate in political issues.

Keywords: Recommendation system · Data mining · Content-based filtering · Collaborative filtering

1 Introduction

Making a prudent and careful decision in political activities such as election would be substantially important to the people and the society for democratic countries to be ones of the people, for the people, and by the people. Compared to the tangible benefit, it takes substantial time and effort as well for them to obtain unbiased, up-to-date and accurate information to support their decision. It is due to the massive volume, speed,

A. Marcus and W. Wang (Eds.): DUXU 2017, Part III, LNCS 10290, pp. 129–145, 2017.
DOI: 10.1007/978-3-319-58640-3_11

variety of information that on/off-line mass media produce in every single minute. It may lead people to find easier way to make their decision with partial impression and gossip rather than focusing on intrinsic values before making decision that might change their life seriously.

To help this issue, it would be a sagacious choice to use a recommender system that filters the "Big Data" to find them out essential, unbiased, and accurate information. In this paper, we suggest an elaborate recommender system that reveals political information unbiased and closely-related to users in accordance with personal interest and inclination has been proposed. Recommender system is a system that recommends information in which a certain user has his or her own interests in the presence of too much information. In this study, it is expected that people are able to save their time and effort to obtain information to support their decision and to actively participate in political issues via the system that combines both collaborative filtering and content-based filtering algorithm.

Collaborative Filtering uses other people's evaluations to filter information. The basic idea is that people are interested in an item which other like-minded users. In contrast, the Content-based Filtering method provides recommendations by matching customer's profiles with content features. Content-based Filtering shows recommendations in the order of results measured by the similarity between users' area of interest and contents of items. Users indicate their opinions of how they think of the behavior of politicians via ratings. Hence, for this study, a survey made by 202 users' ratings about 300 politicians is used as the collected data, and the collaborative filtering correlates these numerical preferences with those of the other users to determine how to make future predictions. Also, collaborative filtering shares the ratings with others so that they can use them in making their own predictions. With this data, collaborative filtering applied by Cosine similarity is used, and to determine optimized the number of politicians recommended, the K-Nearest Neighbor algorithm has been applied. Cosine similarity is a measure to calculate the similarity among users with cosine values between two vectors of the groups, and it is applied only to the group for those who have sufficient number of record for politician preference. KNN algorithm gives K number of other highly similar neighbors, combining with users' profiles such as political orientation so that it can provide more accurate recommendation [1]. However, in the case that there is not sufficient number of users' preference data, we call it "Cold Start Problem." It degrades the recommendation quality. To mitigate the problem, content-based filtering is exploited, and it requires Jaccard similarity method due to the group for those who lack of preference records. It represents similarity among sets of binary data in terms of groups.

With comprehensive experiments in the study, two existing filtering methods with the one proposed above are compared. Since collaborative filtering with rating-based data via abundant data and content-based filtering with users' property-based data to overcome data scarcity, also called cold start problem, have been tested, Hybrid Filtering method that is mixture of collaborative and content-based filtering is suitable to politician recommender system.

2 Preliminaries

2.1 Dataset Description

User's Preference Data for Current Congress Members in Korea
We collect users' preference data of 300 politicians via surveys proceeded in mainly two terms, which were November 2016 and January 2017. Every user can do survey only once, and in every survey, there are totally 300 politicians who belong to different parties in South Korea. Users rate politicians on the scale of 1(very dissatisfaction) to 5 (very satisfaction), and they just rate politicians whom they know well due to more accuracy in a recommender system. In November 2016, which is the first term, we conducted online surveys of the preference about 300 congress members with 40 users in all age group. We gave a survey to every user so that he or she rated politicians whom they know well among 300 politicians. After that, we combined all surveys from 40 users, and Table 1 illustrates a part of the rating table combined with all surveys about 300 politicians as items by 40 users.

Table 1. Sample of a rating table with 40 users for 300 politicians

User	Item				
	Politician A	Politician B	Politician C	Politician D	Politician E
User 1	NA	NA	NA	NA	NA
User 2	NA	4	NA	NA	3
User 3	NA	3	3	NA	2
User 4	NA	3	3	NA	NA
User 5	NA	NA	NA	NA	NA
User 6	NA	3	NA	NA	NA
User 7	NA	NA	NA	NA	NA
User 8	NA	NA	NA	NA	NA
User 9	NA	1	3	NA	3
User 10	NA	3	NA	NA	NA

According to Table 1, NA represents a case when a user, who does not know a politician, ignores rating the politician. After the first data collection from 40 users, we decided to use 114 politicians for the second term of surveys because 114 politicians are only congress members who are rated by two or more users, and we need only this type of group to improve accuracy during evaluation of the recommender system. Through online surveys during the second term conducted in January 2017, we collected the data from 163 users of all age group on the preference of these 114 politicians. After the surveys, Table 2 is made by 195 users for 114 politicians. The reason for totally 195 users instead of 163 users is that we combined surveys of 40 users during the first term with surveys of 163 users during the second term, and we removed 8 users who did not rate at all and who rated only one politician during the surveys. This first data cleaning makes the recommender system more efficient in accurate recommendations with less error in evaluations.

Table 2. Rating table with 195 users

	Politician A	Politician B	Politician C	Politician D
User 183	0	2	0	0
User 184	0	2	0	0
User 185	0	1	0	0
User 186	2	4	5	3
User 187	0	1	0	0
User 188	0	1	0	0
User 189	0	4	0	0
User 190	3	2	0	4
User 191	5	4	0	3
User 192	0	2	0	0
User 193	0	1	0	2
User 194	0	3	2	3
User 195	0	2	5	5

Public Profile of Current Congress Members in Korea

In order to build a content-based recommendation system, we need items' attributes, so we collect the public profile of current congress members in Korea by using web-scraping from the website that monitors legislative activities. We obtain the item (politician) attributes matrix as described in Table 3. We decide to recommend a politician based on politician's attributes with five categories: Political Career, Standing Committee, Political Orientation, Specialization, and Military Service. In fact, there are sixteen standing committee in National Assembly of Korea, but we classify them into eight areas by considering their relevance as described in Table 4.

Party 1 is a ruling party and conservative. Party 2 is the first opposition party with the most seats and progressive. Party 3 is a moderate political party holding a casting vote. Political career is a standard that indicates whether a politician is a first-time member of National Assembly. In South Korea, every Korean man has to fulfill his

Table 3. Description of attributes matrix of congress members

Name	Party	Political career	Standing committee	Political orientation	Specialization	Military service
Politician A	Party 1	Two or more	Jurisdiction, culture and science	Conservative	Lawyer	Finished
Politician B	Party 3	Two or more	State affairs	Moderate	Police	Excepted
Politician C	Party 2	New	Welfare, jurisdiction, state affairs, society and economy	Progressive	Lawyer	Excluded (female)
Politician D	Party 3	New	Culture and science, state affairs	Moderate	Lawyer	Finished
Politician E	Party 3	New	Welfare, state affairs	Moderate	Politics	Finished
Politician F	Party 3	Two or more	Welfare, society and economy	Conservative	Public officer	Finished

Table 4. Description of standing committees in South Korea and their decision-making coverage.

Standing committee	Decision-making coverage	Classified
National defense	National defense	National security
Intelligence	National information	
Strategy and finance	Financial and economic policies	Society and economy
Trade, industry and energy	Trade, industry and energy	
Agriculture, food, rural affairs, oceans and fisheries	Agriculture, food, rural affairs, oceans and fisheries	
House steering	Matters concerning the operation of the national assembly	State affairs
Land, infrastructure and transport	Land, infrastructure and transport	
Security and public administration	Internal administration and election	
National policy	Political affairs	
Foreign affairs and unification	Foreign affairs and unification	Foreign and unification
Legislation and judiciary	Review and supervise matters concerning judicial institutions	Jurisdiction
Science, ICT, future planning, broadcasting and communications	Science, technology and broadcasting communication	Culture and science
Education, culture, sports and tourism	Education, culture, sports and tourism	
Environment and labor	Environment and labor	Environment and labor
Gender equality and family	Gender equality and family	Welfare
Health and welfare	Health and welfare	

duty of military service. However, some people do not have the duty of it for some reasons such as illness or difficulty in living. On the other hand, women are excluded from military service obligations.

2.2 Recommender System and Application

A recommender system is a system that guesses and recommends particular items that a user would prefer via information filtering among massive amount of information. The definition of filtering, one of the IT terms, is a technique to pick appropriate items out of various contents. There are mainly two algorithms used in a recommender system, which are an algorithm called collaborative filtering and an algorithm called content-based filtering.

The collaborative filtering filters information by using recommendations of other people. It is based on the idea that people who agreed with their evaluation of certain items in the past are likely to agree again in the future. In this type of recommendation, filtering items from a large set of alternatives is done collaboratively between users'

preferences. The collaborative filtering only considers user preferences and does not take account of the features or contents of the items being recommended [2]. The collaborative filtering leads to an advantage in which this approach requires a large set of user preferences for more accurate results [2]. When more abundant data in terms of users and items has been collected, this filtering exploits better showcases of recommendation. In collaborative filtering, there are two variants of algorithms to approach, which are user-based collaborative algorithm and item-based collaborative algorithm.

User-based collaborative algorithm, also called memory-based algorithm, uses the whole user database to create recommendations by analyzing rating data from many individuals [1]. This algorithm gives a target user recommendation, which is also preferred by similar users. There is an assumption that similar users do rate similarly. However, the user-based collaborative filtering has some limitations. One is that it is difficult to measure the similarities between users because the size of data for making the recommender system keeps changing, so it is hard to find the optimal similarities. The other is the scalability issue. As the number of users and items increases, the computation time of the algorithm grows exponentially, thus it makes the system slower [3].

Item-based collaborative algorithm, also called model-based algorithm, produces recommendations based on the relationship between items inferred from the rating matrix [1]. In this algorithm, there is an assumption, which represents that users prefer items similar to other items they like. As item-based collaborative algorithm calculates similar items, it is proposed to overcome the scalability, which is a limitation of user-based algorithm. However, an issue is the ratings, which include some discrete values, and these ratings cannot provide much information about relationship between users and items [3].

In terms of collaborative filtering, if the size of information increases, accuracy of the recommender system is also improved. However, there is still an issue when using collaborative filtering only. The issue is called cold start problem, and it occurs in a case of the sparsity of information available in the recommendation algorithm. Even though the collaborative filtering has the cold start problem, the content-based filtering has a solution of this issue because it does not rely on users' preferences data for items.

Content-based filtering is a filtering method that recommends items based on similarity between user's profile and the contents of the items, so it basically recommends items that are similar to those that the user has bought or liked in the past. We can define the process of content-based filtering in three steps. First, it analyzes and categorizes items' attributes. Second, it retrieves user's profile based on user's interests or purchases of items. Third, it calculates the similarity between items and the user's profile in order to recommend the items to the user [2]. For example, in a movie recommendation system, the database contains the attributes of each movie such as genre, director, stars, and studio. If a user watched a movie 'Avatar' and rated high scores, the system would build this user's profile considering the attributes of the movie 'Avatar'. Then, the system recommends movies that have similar attributes to the user's profile.

Content-based filtering does not have cold start problem, since it does not require other customers' data to recommend items to users. This method can begin the recommendation as long as there is enough information about the items and the users in

the database. On the other hand, this method has some disadvantages: limited content analysis and over-specialization. Limited content analysis refers to the situation where the recommendation's performance is not precise and poor due to a lack of attributes representing the items. Over-specialization, also known as serendipity problem, means that content-based filtering only recommends items within its expected range, so there would be no surprise recommendation that is not similar to the user's profile [4].

To build recommendations using both collaborative and content-based filtering algorithm, two similarity methods within neighboring range has to be calculated. First, collaborative algorithm requires calculation of similarity in order to predict the missing ratings based on neighborhood of either similar users or items. The range of neighborhood is measured via similarity between users or items, and there are two ways of measuring the similarity, which are Pearson correlation coefficient and the Cosine similarity. Pearson correlation coefficient is [2] a popular correlation coefficient calculated between two variables as the covariance of the two variables or users divided by the product of their standard deviations, and this is given by ρ (rho):

$$\rho_{X,Y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y}$$

Cosine similarity is [2] a measure of similarity between two vectors or users of an inner product space that measures the cosine of the angle between them, and the equation is given by

$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|}$$

X, Y in Pearson correlation coefficient and A, B in cosine similarity denote the row vectors between two users. The range of both similarity methods is from −1 to 1. The value of negative one represents lowest similarity while the value of positive one is representing highest similarity. Higher similarity indicates closer relationship between two users. Second, for content-based filtering algorithm, Jaccard's coefficient is applied. Jaccard's coefficient is a measurement of similarity between binary sets of variables, and it is defined as the intersection of two data sets divided by their union. It becomes higher when the two data sets have more attributes in common.

$$\text{Jaccard Coefficient}(\text{A}, \text{B}) = \frac{|A \cap B|}{|A \cup B|}$$

Jaccard distance is a measurement of how dissimilar the two sets are, and the formula is [5]

$$\text{Jaccard Distance}(\text{A}, \text{B}) = 1 - \textit{Jaccard Coefficient}(A, B) = \frac{|A \cup B| - |A \cap B|}{|A \cup B|}$$

2.3 Evaluation Metrics

Classification Accuracy Metrics
A confusion matrix [6] is a matrix which contains information about actual and predicted classifications, as described in Table 5.

Table 5. Confusion matrix with actual and predicted classifications

		Predicted	
		Negative	Positive
Actual	Negative	True negative	False positive
	Positive	False negative	True positive

Several standard terms have been defined as follows:

- The **accuracy (AC)** is the proportion of the total number of correct predictions. It is determined using the equation:

$$\text{AC} = \frac{TP + TN}{TP + TN + FP + FN}$$

- The **recall** is the proportion of positive cases that were correctly identified, as calculated using the equation:

$$\text{Recall} = \frac{TP}{TP + FN}$$

- The **precision** is the proportion of the predicted positive cases that were correct, as calculated using the equation:

$$\text{Precision} = \frac{TP}{TP + FP}$$

- The **F-measure** is the harmonic mean of precision and recall, as calculated using the equation:

$$\text{F-measure} = \frac{2\,Precision\,Recall}{Precision + Recall} = \frac{2}{1/Precision + 1/Recall}$$

ROC graphs are the way besides confusion matrices to examine the performance of classifiers. A ROC graph is a plot with the false positive rate on the X axis and the true positive rate on the Y axis [6]. The area under the curve means accuracy (Fig. 1).

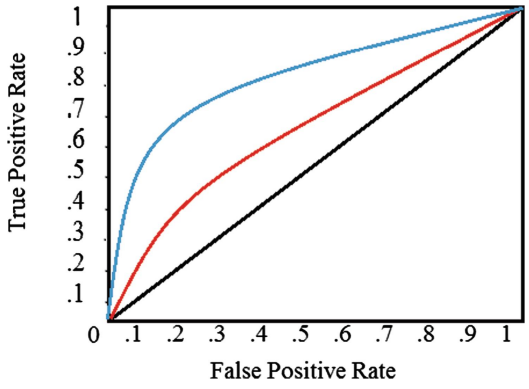

Fig. 1. ROC curve grap

Predictive Accuracy Metrics

Let y_i denote the ith observation and $\hat{y}_i$ denote a forecast of y_i. The forecast error is simply $e_i = y_i - \hat{y}_i$, and accuracy is measured based on e_i. The two most commonly used measures are based on the squared errors or absolute errors [7]:

$$\text{Root mean squared error: RMSE} = \sqrt{mean(e_i^2)},$$
$$\text{Mean absolute error: MAE} = mean(|e_i|).$$

The **root mean squared error (RMSE)** is the square root of the mean squared error. This is the statistic whose value is minimized during the parameter estimation process, and it is the statistic that settles the width of the confidence intervals for predictions.

The **mean absolute error (MAE)** is also measured in the same units as the data, but slightly smaller than, the root mean squared error. It is less sensitive to the occasional very large error because it does not square the errors in the calculation.

3 Methodology

3.1 Preprocessing

For both collaborative and content-based filtering, Table 2, which is a rating table with 195 users for 114 items or politicians, is mainly applied. However, if we just use the same rating table for both filtering algorithms, then optimized recommendations cannot be showcased because they basically use different similarity methods. According to the definition of filtering algorithms, collaborative filtering derives optimized recommendation mostly when enough items exist. On the other hand, content-based filtering is required when a scarcity of items occurs. To reduce this cold start problem, we specify a threshold of the amounts of items defining a word 'enough' so that we combine content-based filtering with collaborative filtering. However, data with few items rated by users degrades the system effect. To prevent this degradation, we assume that users

who rated less than 10 items or politicians distract our experiment, and we thus make a new rating table by removing lists of these users in our experiment.

We convert the rating table into both real rating matrix and binary rating matrix. The real rating matrix is used in calculating similarity of collaborative filtering. In content-based filtering, we use Jaccard distance to calculate the similarity between items and users, and two binary datasets are needed to use this measure. The first binary dataset is acquired by transforming the item description matrix, Table 3, into the binary form as described in Table 6. This is a part of the binary form because there are 26 attributes in the item matrix. Each item has different number of attributes because they can belong to more than one standing committee, so we divided the each attribute by the square root of the total number of attributes of that item to give equal weights. The second binary dataset is a user profile binary matrix, and it is retrieved by combining the user's rating matrix and Table 6 in order to calculate the Jaccard distance between the items and the users. The user profile binary matrix is demonstrated in Table 7.

Table 6. Binary form of attributes matrix of congress members

Name	Society and economy	Culture and science	Welfare	State affairs	Jurisdiction	Progressive	Conservative	Moderate
Politician A	0	1	0	0	1	0	1	0
Politician B	0	0	0	1	0	0	0	1
Politician C	1	0	1	1	1	1	0	0

Table 7. Binary matrix for user-item (politician)

Item	User							
	User 1	User 2	User 3	User 4	User 5	User 6	User 7	User 8
Politician A	0	0	−1	−1	0	0	0	0
Politician B	0	0	0	−1	0	−1	0	0
Politician C	0	0	1	1	0	−1	0	0
Politician D	0	−1	−1	−1	0	−1	−1	−1
Politician E	0	0	1	1	0	−1	1	0

We normalize the user's rating matrix because different users have different criteria in rating items, and retrieve the user's profile for 26 attributes based on their preferences for politicians. In this method, we assume that a user is interested in attributes that belong to a politician whom he or she prefers. With this assumption, we obtain the user's profile from the user's preference data for the politicians and their attributes.

For collaborative filtering, it produces the recommendations based on the relationship between items or users inferred from the rating matrix. To reduce user-bias or item-bias problem, we first normalize the user-item rating matrix before computing similarity. We use a package *recommenderlab* in R language, which includes normalization in center basis.

With user-item real rating matrix, we make a tuple form reshaping the original data by *melt* function in R language in order to improve accuracy in evaluation. While the real rating matrix is composed of rows with users who rated items, the tuple consists of three columns, which are User, Politician, and Rating. If we use the user-item matrix, we use each row as a vector form when analyzing data in programming. However, if we use user-item-rating tuple as an object, we can easily proceed the experiment. Table 8 illustrates the tuple form with three columns of different contents in R language.

Table 8. Tuple of user-item-rating

User	Politician	Rating
2	Politician A	1
2	Politician B	4
2	Politician C	3
2	Politician D	2
2	Politician E	1
2	Politician F	5

3.2 Recommendation Model

A basic idea is that we build two types of recommender systems, which include collaborative filtering via *recommenderlab* and content-based filtering in R language. Before using the rating table for either real rating matrix or binary matrix, the table should be normalized to improve recommendations, and *recommenderlab* provides automatic calculation of normalization based on center in UBCF and IBCF methods. In contrast, since there is not any recommendation package in content-based filtering, we normalized each object directly. Depending on threshold of information, it is determined that which algorithm is applied to the boundary of the data. To set up the certain amount of 'enough' information, we find a boundary between two filtering algorithms to elicit optimized outcome.

Baseline Model
Using a *POPULAR* method in *recommenderlab*, we design a baseline model that recommends the most popular politicians that showed up in the user-rating matrix most frequently. Having a solid baseline based on the popularity makes it possible to identify why content-based and collaborative filtering are better than the baseline. Therefore, the baseline model is a tool of getting the optimized performance via both content-based filtering and collaborative filtering.

Defining the Training and Test Sets
We separate the entire data formed by tuples into two sets: training and test sets. The two sets are as follows:

Training set: This set includes users for the model to learn
Test set: This set includes users whom we recommend politicians.

To evaluate the model, we randomly split the data into training set and test set with fixed ratio of 8:2. The training set is used to train the model in collaborative filtering. To make better performance in collaborative filtering model, optimal training set is required, and we found out that randomly sampling 80% of the entire data is the best condition to make a training set due to sparsity of the information. The test set is to evaluate the performance of both collaborative and content-based filtering.

Application of Collaborative Filtering and Content-Based Filtering

In collaborative filtering, there are two ways to calculate similarity, which are cosine similarity and Pearson correlation coefficient. Comparing two methods, we observed which similarity method is the best fit for pursuing the most optimal recommender system. Figure 2 shows the comparison between Cosine similarity and Pearson correlation coefficient.

In Fig. 2, IBCF stands for item-based collaborative filtering, and UBCF stands for user-based collaborative filtering. According to the Fig. 2, regardless of which type of collaborative filtering is applied, a filtering used with cosine similarity shows better performance than a filtering used with Pearson correlation coefficient. Therefore, finding the optimal neighborhood is integrated by cosine similarity in Collaborative filtering. In content-based filtering, however, Jaccard distance similarity is calculated to measure the distance between two binary sets.

After making a decision of which similarity method is applied to, we do data cleaning work to find the threshold of 'enough' information because we can make a

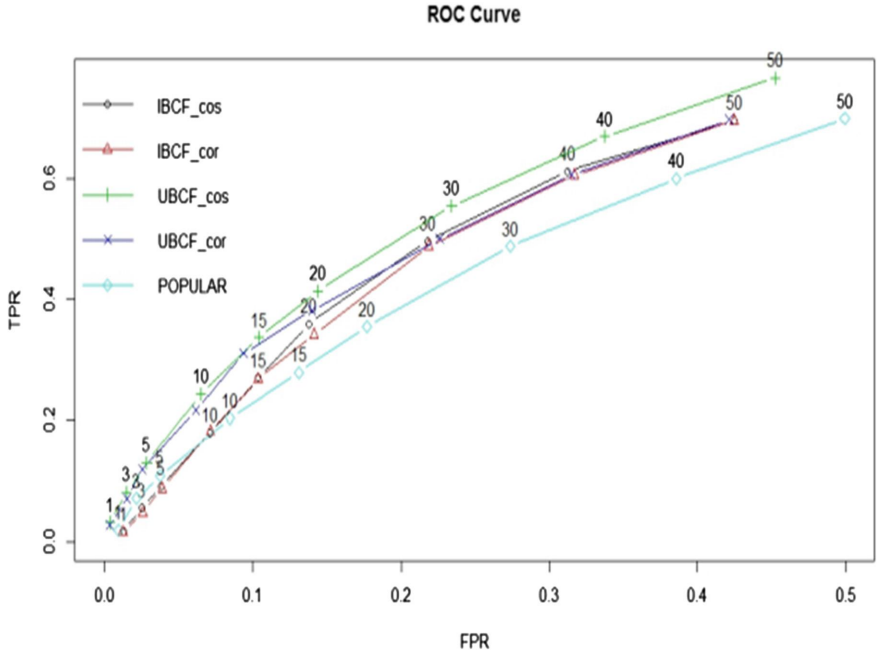

Fig. 2. IBCF and UBCF graph with cosine similarity and Pearson correlation coefficient

choice of which model has to be applied for outstanding recommendations at the boundary. Data cleaning work is done by a number of experiments of evaluation. To set up the boundary between collaborative filtering and content-based filtering, *F-measure* is applied after Top-N closest neighbors or politicians are recommended to a user. N is the number of items recommended to the user. Also, we check how precisely and correctly each model recommends items compared to baseline model.

Within the area of collaborative filtering, in order to compare the performance of user-based filtering to that of item-based filtering, we use *RMSE* and *MAE* figure in evaluation. *RMSE* and *MAE* show the efficiency of each algorithm by demonstrating a fact that lower value refers to less error detected, and the fact represents the better performance in the system. In our experiment, we prove that which model within collaborative filtering area shows the best function via *RMSE* and *MAE.*

4 Result and Discussion

Evaluation Measure

We proved that content-based filtering is essential for a group that consists of users who did not rate enough items, and collaborative filtering is necessary for a group that consists of users who rated enough items. After we removed a group of users who rated less than 10 politicians for more accuracy in preprocessing, we defined how 'enough' information or data is reliable. Table 9 shows the performance evaluation to compare each model to the baseline model, which is written POPULAR in the table. N represents the number of politicians recommended to users. CBF stands for content-based filtering.

According to the Table 9, the *F-measure* is measured via a group of users who rated 10 to 40 politicians. In this experiment, user-based collaborative filtering and content-based filtering outperformed the popularity-based model because all *F-measures* are higher than those of the baseline. Another observation is that *F-measures* with content-based filtering provide the better performance than those with collaborative filtering.

Table 9. F-measure average with comparison both UBCF and CBF to POPULAR

N	UBCF	POPULAR	CBF
10	0.243012	0.214243	0.332091
11	0.248496	0.211481	0.316799
12	0.239636	0.209155	0.302887
13	0.240742	0.204241	0.348843
14	0.239034	0.213361	0.356697
15	0.235963	0.205829	0.354188
16	0.232032	0.204934	0.356026
17	0.226061	0.204575	0.350058
18	0.219688	0.1984	0.32345
19	0.215003	0.193966	0.318362
20	0.209113	0.194544	0.314459

For the most optimized scale of N, which is the number of recommended politicians, we used Table 8. According to the Table 8, positive values represent that users relatively prefer politicians, but negative values represent that users do not prefer politicians. We observed that the average number of positive values where every user relatively prefers is about 10 politicians, and we concluded that the best scale of N is between 10 and 20. Figure 3 illustrates the superior performance of content-based filtering compared to other two models when recommending between 10 and 20 politicians with users who rated between 10 and 40 politicians.

However, for data cleaning based on the users who rated more than 40 politicians, collaborative filtering showed outstanding performance than the content based filtering in terms of recommendations among users who rated more than 40 politicians, Fig. 3 describes a graph for Table 10.

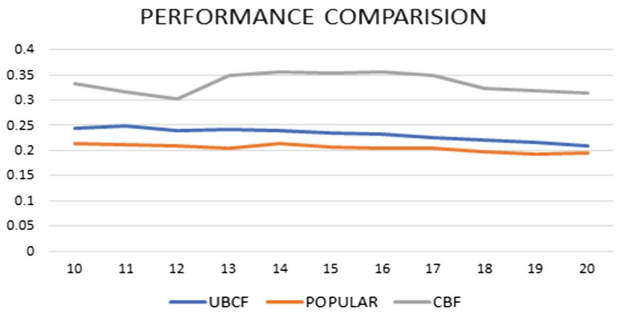

Fig. 3. Graph for comparison of the performance between two filtering model

Table 10. F-measures of 3 recommendation models

N	UBCF	POPULAR	CBF
25	0.546558	0.474066	0.370602
30	0.575327	0.511171	0.40977
35	0.598693	0.535087	0.432486
40	0.618448	0.561064	0.446404
45	0.631795	0.584981	0.457378

Since we have known that the average number of politicians who are rated positively by the users is about 37, we decided to make the scale of N from 25 to 45 as the number of recommended politicians.

According to Fig. 4, with higher performance of UBCF compared to CBF, we recognized that using UBCF is more efficient than using CBF within the threshold of more than 40 politicians rated by users. Also, we concluded that the threshold between collaborative and content-based filtering is about 40 politicians or items.

After a result that collaborative filtering makes higher performance than content-based filtering in data cleaning based on users who rated more than 40 politicians, we observed that which filtering algorithm in collaborative filtering exploits

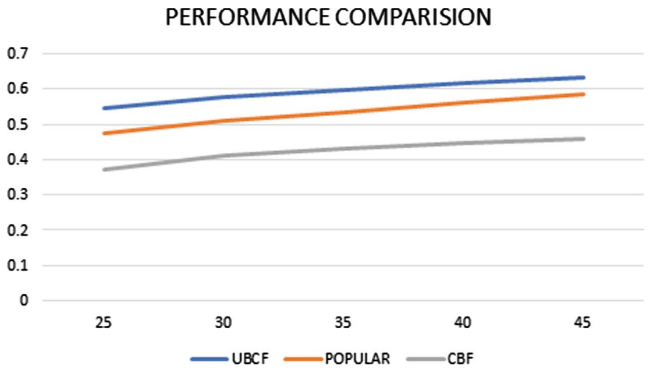

Fig. 4. Graph of performance comparison with more than 40 politicians

better outcomes. We proved that user-based filtering showcases less error than item-based filtering, which indicates that UBCF derives better function than IBCF. According to Table 11, after calculating the error via RMSE and MAE, we proved that user-based algorithm shows the better performance with less error value than item-based algorithm (Fig. 5).

Table 11. Accuracy between UBCF and IBCF

	RMSE	MAE
IBCF	1.052119	0.741415
UBCF	0.957778	0.679757
POPULAR	1.031537	0.779242

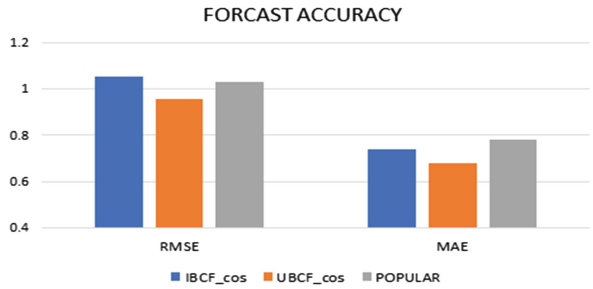

Fig. 5. Graph of RMSE and MAE for collaborative filtering methods

Result and Discussion

According to the experiments, we observed that there was not enough preference data to build a user's profile in terms of content-based filtering for users who rated less than 10 politicians. Also, in collaborative filtering, we removed the group of the users in order to prevent the degradation of the performance due to the scarcity of data.

After the elimination of the group, we realized that it is better to use both content-based filtering and collaborative filtering to improve the performance of the recommendations. We measured the appropriate threshold proceeded via the experiments.

We observed that the threshold is a group of users who rated 40 politicians. Based on the boundary, we applied content-based filtering algorithm into making recommendations for users who rated the politicians below the threshold. The optimal number of recommended politicians in content-based filtering was 10 to 20 politicians. On the other hand, for users who rated politicians above the threshold, we concluded that using collaborative filtering shows the best performance of recommendations with 25 to 45 politicians as the optimal number of recommended politicians.

5 Conclusion and Future Work

Through these experiments, we have confirmed that recommendations of items reflecting the users' profile and their preferences have improved the performance of the recommendation system. At the same time, however, there are some challenges to implementing a more accurate recommendation system. First, we expect that the system will produce a higher performance if the user's actual preference data for item attributes is reflected when building the user's profile in content-based filtering. Also, the data with about 200 users is insufficient for the recommendation system to achieve satisfactory performance. If a preference data with more users is collected, then it will also contribute to more accurate recommendations.

Our research can be used as a tool that can lead to a sagacious political decision-making and active political participation in democratic society for the future.

References

1. Hahsler, M.: recommenderlab: A Framework for Developing and Testing Recommendation Algorithms. Southern Methodist University, Texas (2011)
2. Gorakala, S.K., Usuelli, M.: Building a Recommendation System with *R*. Packt Publishing, Birmingham (2015)
3. Yao, G., Cai, L.: User-Based and Item-Based Collaborative Filtering Recommendation Algorithms Design. University of California, San Diego
4. Lops, P., et al.: Content-based recommender systems: state of the art and trends. In: Ricci, F., et al. (eds.) Recommender Systems Handbook, pp. 73–105. Springer, Heidelberg (2011)
5. Niwattanakul, S., et al.: Using of Jaccard coefficient for keywords similarity. Paper presented at International MultiConference of Engineers and Computer Scientists, Hong Kong, 13–15 March 2013
6. Hamilton, H.J.: Knowledge Discovery in Databases. University of Regina School of Computer Science (2012). http://www2.cs.uregina.ca/~dbd/cs831/index.html. Accessed 10 Feb 2017
7. Hyndman, R.J., Athanasopoulos, G.: Forecasting: principles and practice. OTexts (2013). https://www.otexts.org/fpp/2/5. Accessed 10 Feb 2017