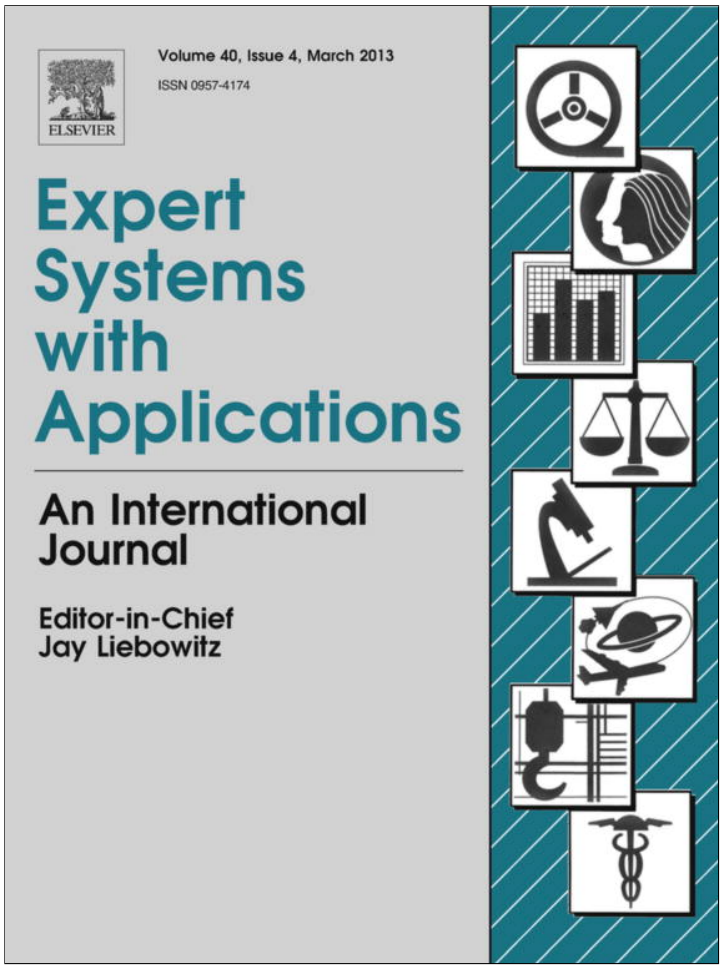

Expert Systems with Applications 40 (2013) 1274–1282

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

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

# Representative reviewers for Internet social media

Sang-Min Choi, Yo-Sub Han *

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

ARTICLE INFO

*Keywords:*
Social-network
Social-media
Influential users
Representative reviewers

ABSTRACT

Our many various relationships with persons from home, work and school give rise to our social networks. In a social network, people receive, provide, and pass a great deal of information. In this process, we often observe that certain individuals have especially strong influences on others. We call these highly influential people opinion leaders. Since the late 20th century, the number of Internet users has increased rapidly, and a huge number of people now interact with each other in online social networks. In this way, the Web community has become similar to real-world society. Internet users receive information not only from the mass media, but also from opinion leaders. For example, online articles posted by influential bloggers are often used as marketing tools or political advertisements, due to their huge influence on other users. Therefore, it is important and useful to identify the influential users in an online society. We thus propose a simple yet reliable algorithm that identifies opinion leaders in a cyber social network. In this paper, we first describe our algorithm for identifying influential users in an online society. We then demonstrate the validity of the selection of representative reviewers using the Yahoo! music and GroupLens movie databases and performing 10-fold cross-validation and z-tests.



## 1. Introduction

The mass media supplies a great deal of information, which affects us in various ways. However, people often receive information not from the mass media directly, but through the opinions of individuals called opinion leaders. In fact, it seems that opinion leaders have more influence on people than the mass media does (Katz & Lazarsfeld, 1955). While the public does not generally accept information from the mass media uncritically, they do tend to easily accept information from opinion leaders. This suggests that opinion leaders are representatives of their social networks. Since the late 20th century, the number of Internet users has noticeably increased. Especially in the Web 2.0 era, the number of users and the amount of information increased substantially due to the various types of user participation. For instance, many use Yahoo! Answers[1] to post questions and seek answers from other users, and many use Web services such as Google[2] or Wikipedia[3] to search for information available on the Web. These Web-based interactions create a platform where people meet each other and share stories and information. However, not all information on the Web is reliable, and users often encounter spam and misinformation; therefore, people are more likely to trust information from known, trustworthy sites or users. This is similar to behavior in real-world society, in which most people trust information from opinion leaders more than information from the mass media. Recently, as the Web and user activities have grown, online society has come to resemble real-world society in many ways. Studies that have modeled user participation in social networks (Durugbo, 2012), reconciled users opinions on Web applications (Jung, 2012a, 2012b), and analyzed user interactions in various social activities has shown that the Web has characteristics similar to those in real society. On the Web, we can easily find opinion leaders in blogs and social network applications. Their influential opinions appear as articles, posts, or online content, and they are frequently used for marketing tools or political advertisements. Therefore, the current work to create an algorithm that identifies influential users on the Web is useful and meaningful. In Section 2 of this paper, we discuss previous approaches to identifying opinion leaders in Web applications. Then in Section 3, we describe our algorithm that identifies representative reviewers in Internet social media. In Section 4, we provide verification of our approach through test results that show the validity of the algorithms choices for representative reviewers, and we test reliability using 10-fold cross-validation and z-tests and using the Yahoo! music[4] and GroupLens movie databases.[5] Finally, we provide conclusions and indicate future directions for this research in Section 5.

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

[1] http://answers.yahoo.com/.
[2] http://www.google.com/.
[3] http://en.wikipedia.org/.
[4] http://webscope.sandbox.yahoo.com/.
[5] http://grouplens.org/.

0957-4174/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2012.08.063

## 2. Related study

In this section, we briefly describe previous research on methods that find opinion leaders in Web applications. There are two main approaches: one based on social relations and one based on contents metadata. For example, Kwak, Lee, Park, and Moon (2010) examined opinion leaders on the social-relations Web application, Twitter,[6] which composes a social network by allowing users to connect and communicate with each other using such functions as follow, reply, retweet, and post. They found that the characteristics of users with high numbers of followers are different from users with a high number of retweets, with followers reacting more sensitively to tweets by users with many retweets. In another study based on social relations, Han, Kim, and Cha (2012) described a social network based on video contents and user activities such as subscription, uploading and favorite. They used a modified PageRank algorithm to calculate user reputation in this contents-based social network, finding that it was closely related to subscriptions and the number of uploads. They proposed an algorithm that composed the social-network from users of the video contents and derived user reputation based on uploading and subscription. Another approach is based on contents metadata, such as the algorithm proposed by Agarwal, Liu, Tang, and Yu (2008) that identifies an influential user on blog sites. An important function of blogs is to post various media contents and to tag relevant information from other blogs. The influence of blogs on users in the Web can be strong; for example, the preferences of opinion leader bloggers can affect the purchases made by their visitors, making it profitable to advertise on influential blogs. Various studies have examined the essential issues of identifying influential bloggers, evaluated the effects of various collectible statistics from a blog site to determine blog-post influence, developed unique experiments using Digg,[7] and conducted experiments using whole histories of blog posts. The research to identify influential bloggers has classified the characteristics of bloggers into active, inactive, influential, and non-influential bloggers based on intuition. Active bloggers are those who often create posts, and influential bloggers are those whose posts influence others as determined by social gestures such as comments, incoming links, outgoing links, and lengths of posts. This study clearly demonstrated the existence of influential bloggers, and how influential bloggers relate to each other and to other common visitors. In another study that identified influential users based on contents metadata, Cha, Lee, Han, and Kim (2009) proposed a method to evaluate user reputation on questioning sites such as Yahoo! answers. They collected the n-gram from a title or from a question and its answers, and they calculated similarity using their proposed equation. They then determined user reputation using a modified PageRank algorithm, with scores given to links between questions and answers.

## 3. Our approach

### 3.1. Identifying representative reviewers

We propose an algorithm that identifies the representative reviewers of an evaluative group for Internet social media. By representative reviewers, we mean the users who have high representativeness for media contents as rated by many users. The representative reviewers can express evaluations of the other reviewers. Thus, they are similar to influential users in real society. To identify representative reviewers, we use Eq. (1).

$$U_s = \frac{\sum_{i \in A} |(R_S(i) - R_\mu(i)|}{|A|}, \quad (1)$$

[6] http://www.twitter.com/.
[7] http://digg.com/.

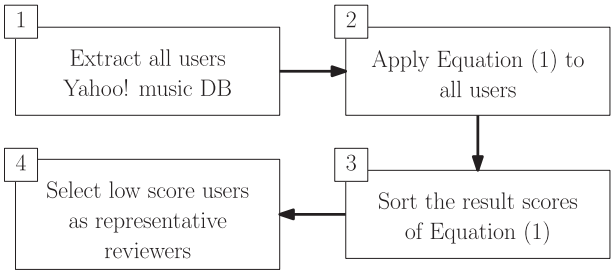


**Fig. 1.** The procedure of identifying representative users from Yahoo! music dataset.

**Table 1**
Yahoo! music database.

| Attribute | Explanation |
|---|---|
| UserID | There exist 15,400 users in total, and given by integer number from 1 to 15,400 |
| ItemID | There exist 1,000 songs in total, and given by integer number from 1 to 1,000 |
| Rating | As integer number from 1 to 5 there exist approximately 300,000 ratings |

**Table 2**
GroupLens movie database.

| Dataset | Attribute | Explanation |
|---|---|---|
| Movie dataset | MovieID, title, genre | There are total of 10,681 movies |
| User dataset | UserID, gender, age, occupation, zip-code | There are total of 69,898 users |
| Rating dataset | UserID, movieID, rating, timestamp | There are total of 10,000,054 ratings |


**Fig. 2.** The distribution of scores ($U_s$) from Eq. (1) in ascending order for the Yahoo! music database.

where $A$ is a set of contents rated by user $S$, and $|A|$ is the cardinality of A. $R_S(i)$ is the rating of content $i$ by user $S$, and $R_\mu(i)$ is the average rating of content $i$ by user $S$. Note that the result of Eq. (1) shows how close each users rating is to the average rating. We select those who have low scores from Eq. (1) as representative reviewers.

Fig. 1 shows the procedure for identifying representative reviewers in the contents raters group. First, we extract all users who evaluate media contents. Second, we apply Eq. (1) to all these extracted users. Third, we sort the result scores of Eq. (1) in ascending order. Finally, we select the low-score users as the representative reviewers.

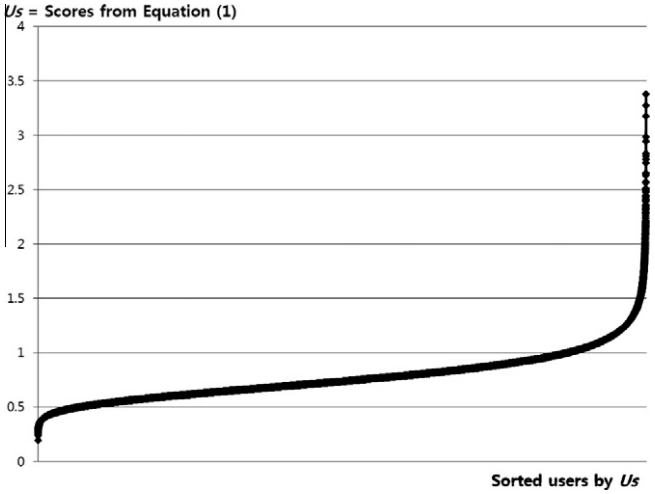


**Fig. 3.** The distribution of scores ($U_s$) from Eq. (1) in ascending order for the GroupLens movie database.

**Table 3**
A example of database.

| | item 1 | item 2 | item 3 | ... | item *M* | No. of ratings |
|---|---|---|---|---|---|---|
| user1 | $R_{11}$ | $R_{12}$ | $R_{13}$ | ... | $R_{1M}$ | 10 |
| user2 | $R_{21}$ | $R_{22}$ | $R_{23}$ | ... | $R_{2M}$ | 20 |
| user3 | $R_{31}$ | $R_{32}$ | $R_{33}$ | ... | $R_{3M}$ | 50 |
| user4 | $R_{41}$ | $R_{42}$ | $R_{43}$ | ... | $R_{4M}$ | 10 |
| ... | ... | ... | ... | ... | ... | ... |
| user*N* | $R_{N1}$ | $R_{N2}$ | $R_{N3}$ | ... | $R_{NM}$ | 30 |

### 3.2. Applying algorithm to Yahoo! and GroupLens databases

#### 3.2.1. Database

We use two open databases: Yahoo! music and GroupLens movie. Table 1 shows the Yahoo! music database.

The GroupLens movie database has three sub-datasets: movies, users, and ratings. Table 2 shows the datasets in the GroupLens database, which has 10,681 movies, 69,898 users, and 10,000,054 ratings.

#### 3.2.2. Applying algorithm to each database

We apply the algorithm identifying representative reviewers to the Yahoo! music and GroupLens movie databases. When we apply Eq. (1) to each database, we obtain two results, which we sort in ascending order. Figs. 2 and 3 show the results of our algorithm for the two databases. In each graph, we consider the low scorers who are close to the *y*-axis as representative reviewers in each rater group.

## 4. Test and analysis

### 4.1. Tests for small size of ratings in each database

In our study, we applied Eq. (1) to a large number of users and items. As explained above, Eq. (1) uses each rating from each user. For example, user A may have 30 ratings for some items, while other users may have different numbers of ratings. Statistically, small sample sizes provide less reliable results. In the previous tests results, shown in Figs. 2 and 3, we did not consider the number of ratings from the users. Namely, we applied Eq. (1) to the database for all users, including those who had a small number of ratings, in the process for identifying representative reviewers.

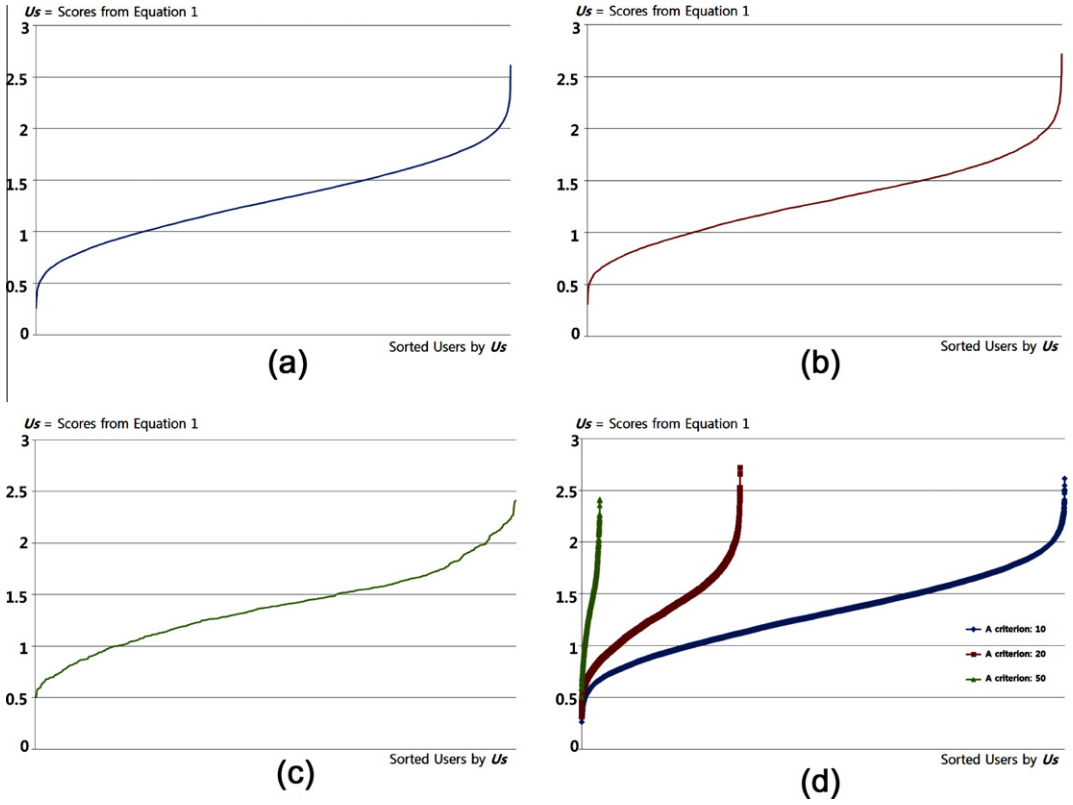


**Fig. 4.** All graphs for each criterion for the Yahoo! music database.

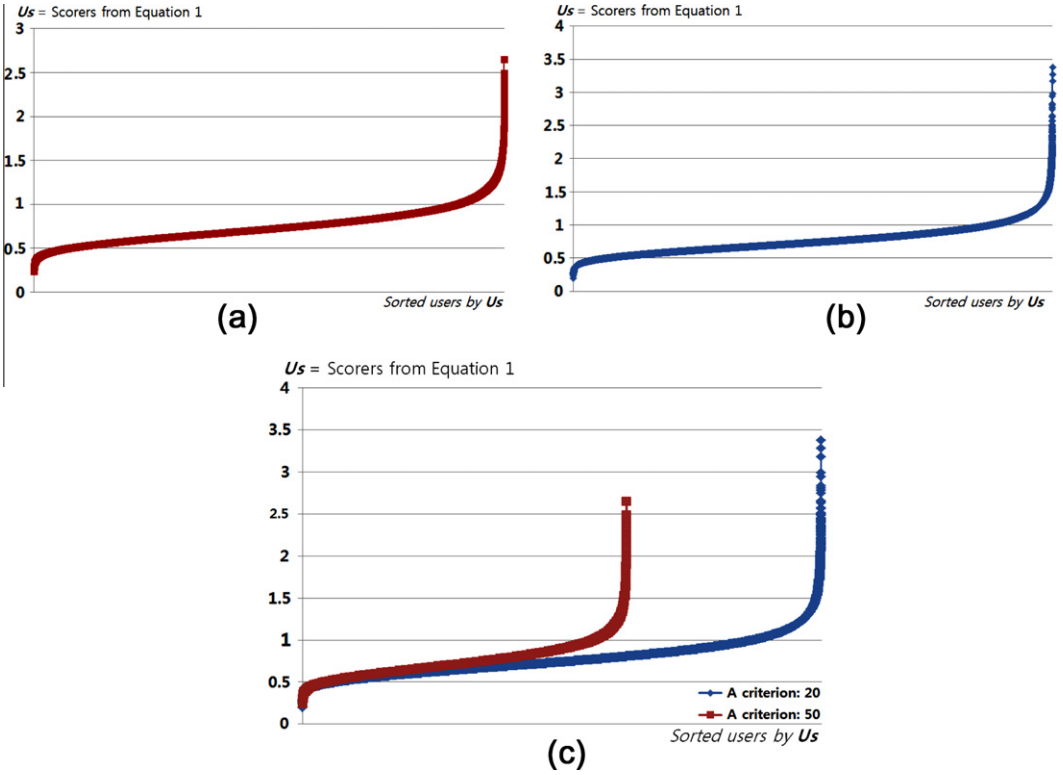


**Fig. 5.** All graphs for each criterion for the GroupLens movie database.

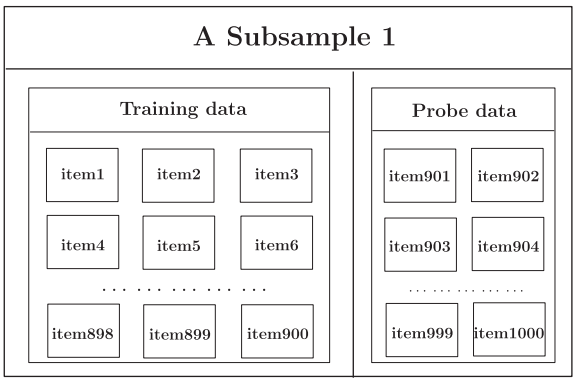


**Fig. 6.** Example of a subsample.

Therefore, if a set of representative reviewers includes many users with only one or two ratings, the reliability of this set is low, since there is a strong possibility of including these users in the set of representative reviewers by chance. For example, if user *A* has only one rating, and the number of items is a thousand, then the number of users is also a thousand. In this situation, if a set of representative reviewers contains user *A*, then the result of Eq. (1) for user *A* is lower than for the users with more ratings. Therefore, in our algorithm, we check for the number of ratings. We first provide test results for different numbers of user ratings using the Yahoo! music (with 15,400 users) and GroupLens movie (with 69,898 users) databases. The users in the Yahoo! music database have at least 10 ratings for items, and the users in the GroupLens database have at least 20 ratings. Using Eq. (1) on these databases without any manipulations with respect to the number of ratings, the resulting graphs are the same as those in Figs. 2 and 3. Then we manipulated the number of ratings from each user: for the Yahoo! music database, we used 10, 20, and 50 ratings, and for the GroupLens movie database, we used 20 and 50 ratings. We provide more details in Table 3.

In Table 3, user 1,...,user *n* represent the userIDs, and item 1,...,item *m* represent the itemID in a database. Each *R* is the

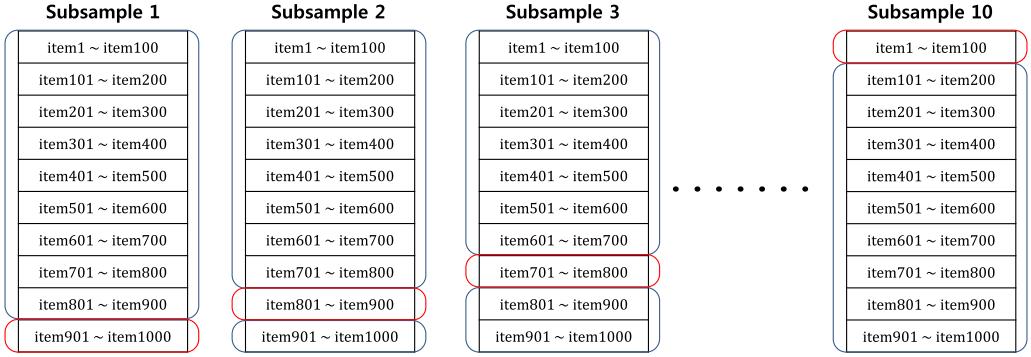


**Fig. 7.** Structures of each subsample.

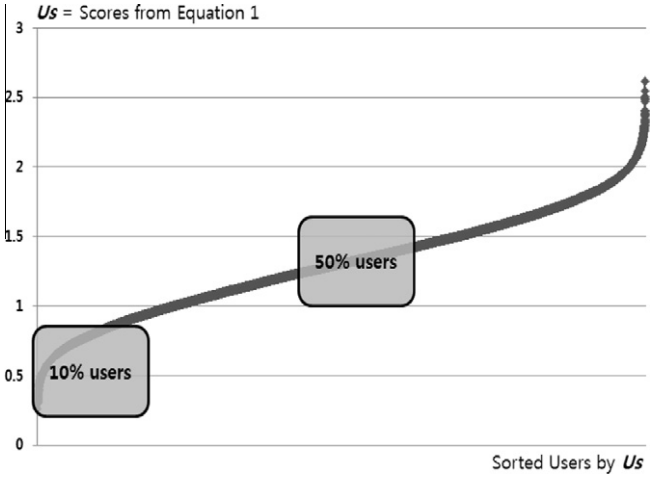


**Fig. 8.** The positions of the 10% and 50% lowest scorers in the results graph.

rating for a specific item provided by a particular user. For example, $R_{12}$ is the rating for item 2 from user 1. Note that although we filled all cells in Table 3 using $R_{nm}$ marks, some can have no ratings. Table 3 also shows the total number of ratings for each user. For example, users 1 and 3 have a total of 10 ratings and 50 ratings, respectively. This means that user 1 provided ratings for 10 items among a total of $m$ items, and user 3 provided ratings for 50 items among a total of $m$ items. We used 10, 20, and 50 as our criteria for the number of ratings. For example, for the criterion of 20, we applied Eq. (1) to users with at least 20 ratings; in Table 3 case, this applied to users 2, 3, and $n$.

Figs. 4 and 5 show the results of Eq. (1) in ascending order for each criterion using the Yahoo! music and GroupLens movie databases.

The graphs (a)–(c) in Fig. 4 show the results for the Yahoo! music database with criteria of 10, 20, and 50, respectively, and the graphs (a) and (b) in Fig. 5 show the results for the GroupLens movie database with criteria of 20 and 50, respectively. These five graphs in Figs. 4 and 5 have shapes similar to those in Figs. 2 and 3 even though the different criteria result in different numbers of users. In Fig. 4, with a criterion of 10 there are 15,400 users, while with criteria of 20 and 50 there are 5,050 and 577 users, respectively. In Fig. 5, with a criterion of 20 there are 69,878 users, and with a criterion of 50 there are 43,608 users. As the criterion number increases, the number of users decreases. As shown in Table 3, a criterion of 10 includes all users in the table, and a criterion of 20 includes only users 2, 3, and $N$. Figs. 4(d) and 5(c) present the graphs for all criteria together for the two respective databases. In these graphs, the $x$-axis represents the sorted users and the

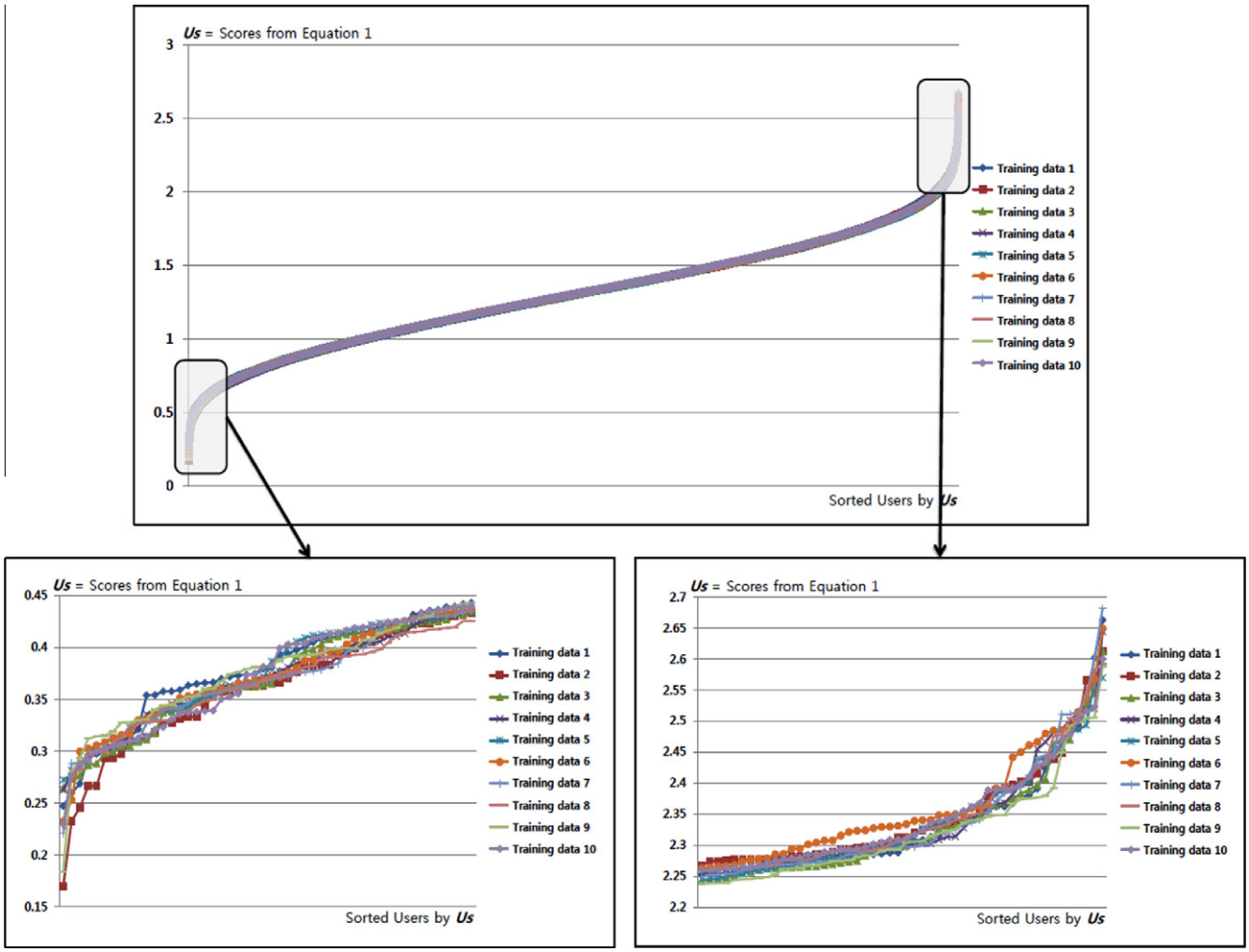


**Fig. 9.** The results of 10 training datasets using the Yahoo! database.

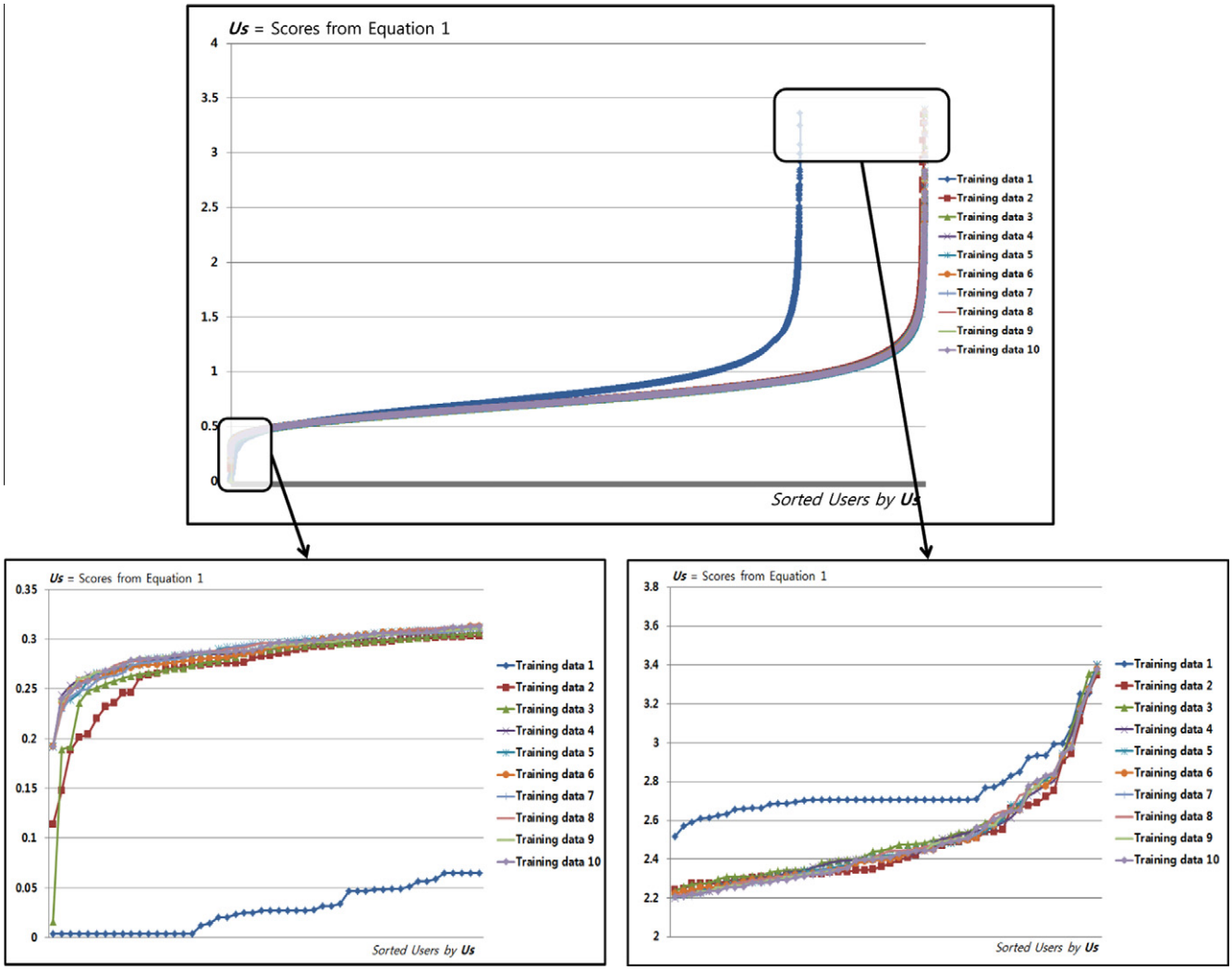


**Fig. 10.** The results of 10 training datasets using the GroupLens database.

| UserID | Rated Items List |
|---|---|
| user1 | item1, item2, item3, item4, item5, item6, item7 |
| user2 | item2, item3, item4, item5, item8, item9, item10 |
| user3 | item1, item3, item4, item5, item7, item8, item10 |
| ... | ..... |
| user$N$ | item3, item4, item5, item6, item8, item9, item10 |

| Training Data |
|---|
| item2, item3, item4, item5 |

| UserID | Rated Items List |
|---|---|
| user1 | item1, item6, item7 |
| user2 | item8, item9, item10 |
| user3 | item1, item7, item8, item10 |
| ... | ..... |
| user$N$ | item6, item8, item9, item10 |

**Fig. 11.** Process for organizing the probe set.

$y$-axis shows the values obtained using Eq. (1). Although there are different numbers of users in these cases, the shapes allow us to extract the users who have high representativeness in each graph: these are the users positioned near the $y$-axis.

### 4.2. 10-fold cross-validation for each database

To show the validity of results obtained using Eq. (1), we provide the results of 10-fold cross-validation for the Yahoo! music

**Table 4**
The result of Eq. (1) applying 10% and 50% scorers.

| | 10% Scorers | 50% Scorers |
|---|---|---|
| Validation data 1 | 0.5100 | 0.7654 |
| Validation data 2 | 0.4878 | 0.7746 |
| Validation data 3 | 0.5036 | 0.7634 |
| Validation data 4 | 0.4902 | 0.7476 |
| Validation data 5 | 0.4882 | 0.7685 |
| Validation data 6 | 0.4906 | 0.7550 |
| Validation data 7 | 0.5071 | 0.7279 |
| Validation data 8 | 0.4808 | 0.7216 |
| Validation data 9 | 0.4686 | 0.7204 |
| Validation data 10 | 0.4462 | 0.7445 |
| Average value | 0.4873 | 0.7489 |

and GroupLens movie databases. To test 10-fold cross-validation, we first divided the database into 10 subsamples. The criterion is the number of items when we divide into subsamples. In our test, a subsample has two datasets: training data with 900 items and probe data with 100 items. Fig. 6 shows the organization for one such subsample, with training data composed of items 1 to 900, and probe data composed of items 901 to 1000.

We called this dataset subsample 1. All ten 1000-item subsamples, subsamples 1 through 10, have different item IDs for their training data and probe data, structured as shown in Fig. 7.

To perform 10-fold cross-validation, we applied our method for identifying representative reviewers to each of the 10 sets of training data. From this, we obtained 10 results graphs similar in shape to those in Figs. 2 and 3. From these, we identified the representative reviewers based on the Eq. (1) scores. We identified the lowest 10% of scorers and the lowest 50% of scorers, noting that the 10% lowest scorers have higher representativeness than the 50% lowest scorers. Fig. 8 shows the positions of the 10% and 50% lowest scorers in our results graph.

Next, we checked the average scores of Eq. (1) for the probe data for the 10% and 50% lowest scoring users. This process is explained in more detail below.

#### 4.2.1. Applying Eq. (1) to training data in each subsample

We performed 10-fold cross-validation of the Yahoo! and GroupLens databases by applying Eq. (1) to the 10 subsamples training data, then sorting the 10 results in ascending order. Figs. 9 and 10 show the 10 resulting graphs. Here, the *y*-axis corresponds to scores from Eq. (1) and the *x*-axis represents sorted users. All the graphs have similar shapes and Eq. (1) scores, making it difficult to judge the differences among them. However, by examining the bottom two graphs in Figs. 9 and 10 showing the top 50 and bottom 50 scorers, we can judge the differences. The bottom 50 users in Fig. 10 have difference positions for each training data. The reason of these differences is the number of users for results of each training data. It means that the result of training data 1 has small amount of users than other training data. The cause of small amount of users in graph is that the users who gave ratings for training data are lower than others. The graphs in right side of bottom two graphs in Fig. 10 have no difference for users since we can consider bottom 50 users in each training data. Examination of these curves shows that the scores in the two bottom graphs are almost never different. Thus, we can anticipate that representative reviewers exist in each subsamples set of training data.

#### 4.2.2. Applying Eq. (1) to probe data in each subsample

Next, we validated the results of the training data using the probe data. We first extracted the top 10% users and 50% users in each sorted result. An example of the sorted results is shown as a graph in Fig. 8. We applied Eq. (1) to the top 10% users and 50% users using only the probe data.

Fig. 11 shows an example of organizing a probe set in which there are *N* users and each user has rated 7 items. The training data are items 2, 3, 4, and 5. We remove these training items from the users rated items list to obtain a new list of users rated items that excludes training items, as shown in the bottom table in Fig. 11. To demonstrate the validity of the results of Eq. (1) for the training data, we used this new rated items list composed of probe data. In the process of organizing probe sets, we first excluded training items rated by 10% users and 50% users so that there were only probe items in the rated items list of users. We applied Eq. (1) to these remaining items and calculated the average scores for the

**Table 5**
The results for each probe dataset in the Yahoo! database for each criterion for the users number of rated items $C_1$ ($C_2$, $C_3$ and $C_4$, respectively) denotes the number of users who have rated at least once (5 times, 10 times and 15 times, respectively).

| Probe Dataset | No. of users | | Average | | No. of users | | Average | |
|---|---|---|---|---|---|---|---|---|
| | 10% users | 50% users | 10% users | 50% users | 10% users | 50% users | 10% users | 50% users |
| | $C_1$ | | | | $C_2$ | | | |
| Dataset 1 | 1360 | 1374 | 0.7995 | 1.2960 | 175 | 188 | 0.7552 | 1.3191 |
| Dataset 2 | 1344 | 1312 | 0.8459 | 1.3157 | 154 | 184 | 0.7810 | 1.2912 |
| Dataset 3 | 1269 | 1248 | 0.8274 | 1.3009 | 83 | 94 | 0.7614 | 1.2959 |
| Dataset 4 | 1334 | 1325 | 0.8241 | 1.3172 | 125 | 158 | 0.7848 | 1.2642 |
| Dataset 5 | 987 | 998 | 0.7781 | 1.3515 | 50 | 53 | 0.7592 | 1.3118 |
| Dataset 6 | 1135 | 1201 | 0.8082 | 1.2856 | 84 | 93 | 0.7429 | 1.3153 |
| Dataset 7 | 1116 | 1192 | 0.8397 | 1.2739 | 60 | 78 | 0.7728 | 1.2743 |
| Dataset 8 | 1326 | 1325 | 0.8195 | 1.3029 | 116 | 122 | 0.7572 | 1.2824 |
| Dataset 9 | 1350 | 1298 | 0.8479 | 1.2955 | 109 | 96 | 0.7105 | 1.2983 |
| Dataset 10 | 1308 | 1282 | 0.7855 | 1.2803 | 108 | 137 | 0.7752 | 1.2918 |
| | $C_3$ | | | | $C_4$ | | | |
| Dataset 1 | 7 | 18 | 0.7150 | 1.2758 | 1 | 6 | 0.5099 | 1.2935 |
| Dataset 2 | 11 | 22 | 0.7825 | 1.2998 | 2 | 4 | 0.6267 | 1.3618 |
| Dataset 3 | 8 | 8 | 0.6574 | 1.3619 | – | 1 | – | 1.6852 |
| Dataset 4 | 9 | 13 | 0.6634 | 1.2980 | 1 | 3 | 0.3758 | 1.4910 |
| Dataset 5 | 3 | 8 | 0.8604 | 1.3521 | – | 1 | – | 1.2149 |
| Dataset 6 | 3 | 11 | 0.6684 | 1.4275 | 1 | 1 | 0.6299 | 1.3923 |
| Dataset 7 | 3 | 8 | 0.8967 | 1.4615 | – | 1 | – | 1.0913 |
| Dataset 8 | 13 | 16 | 0.7030 | 1.1980 | – | 2 | – | 1.3291 |
| Dataset 9 | 7 | 14 | 0.6328 | 1.2467 | 1 | 4 | 0.3825 | 1.2861 |
| Dataset 10 | 6 | 12 | 0.6208 | 1.3647 | – | 1 | – | 1.3341 |

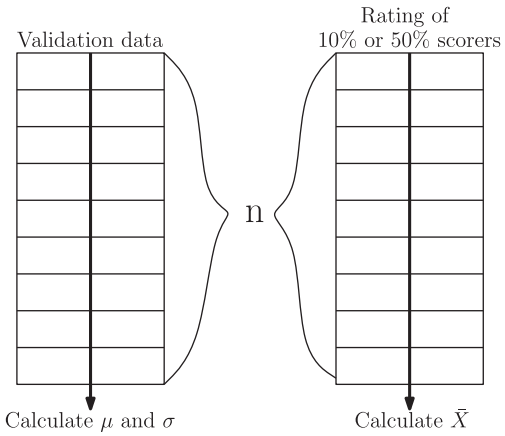


**Fig. 12.** Each element of Eq. (2).

top 10% users and 50% users. Table 4 shows these results when we applied Eq. (1) to each validation dataset for 10% and 50% scorers who were selected using the training data in the GroupLens database.

As shown in Table 4, all 10% scorers had lower average scores than all 50% scorers. This indicates that the ratings of representative reviewers who had low scores in each training dataset were well-representative of the ratings in each validation dataset. Thus, we considered the 10% low scorers selected through each training dataset to be representative reviewers in each validation dataset. This test result implies that representative reviewers exist in each raters group.

Because the Yahoo! database has fewer users and items than the GroupLens database, and since tests based on smaller data sizes have lower statistical reliability, we added some conditions when testing the Yahoo! database. After we had removed 90% of items, the training items, in calculating the average scores with probe data, we checked the number of remaining items in each users rated items list, and compared this number to selected criteria of 1, 5, 10, and 15. We used users who had more than the criterion number of items when we applied Eq. (1) to the probe data. Table 5 shows the results of Eq. (1) for each probe dataset in the Yahoo! database, for each criterion.

The Yahoo! music database had a total of 15,400 users and our validation targets were the top 10% (1,540 users) and 50% (7,700 users). In Table 5, the number of users for an attribute means the number of users who rated items more than the criterion amount of times. The averages for the 10% and 50% users are the average scores resulting from Eq. (1).

As shown in Table 5, all average results for the 10% users are lower than those for the 50% users. Also, almost no probe datasets met the criterion of users having more than 20 rated items. Only probe datasets 2 and 8 met the criterion of 20 rated items: each had one user among their 50% users. There were also some vacancies when a criterion of 15 was used. For criteria 15 and 20, the numbers of users were small in each probe dataset, providing low reliability, so we ignored these results. In comparison, the numbers of users who had rated at least the criterion number of items were markedly larger for the other examined criteria of 1, 5, and 10.

Examination of Table 5 makes it clear that there are fewer 10% than 50% users who have rated items, for both training and probe data. Additionally, the number of 10% and 50% users is similar for each criterion, although the number of rated items is low in some cases. The most important thing is that all cases of 10% users have lower average scores than all cases of 50% users. Further, the 10% users selected based on the training data have similarly low scores when using the probe data. This means that these 10% users in the Yahoo! music database have good representativeness in the probe data, based on the algorithm using Eq. (1).

**Table 6**
The result of Eq. (2) applying 10% and 50% scorers in GroupLens movie database.

| | 10% Scorers | 50% Scorers | $n$ |
|---|---|---|---|
| Validation data 1 | −1.9501 | −2.4556 | 373 |
| Validation data 2 | 0.1066 | 0.8882 | 383 |
| Validation data 3 | 0.9514 | 0.7799 | 298 |
| Validation data 4 | 0.6852 | 0.8086 | 240 |
| Validation data 5 | 0.4686 | 0.3640 | 157 |
| Validation data 6 | 0.7525 | −0.0818 | 95 |
| Validation data 7 | 0.7283 | 0.6930 | 76 |
| Validation data 8 | 0.9949 | −0.0344 | 63 |
| Validation data 9 | 0.7590 | 0.3535 | 26 |
| Validation data 10 | 0.7854 | −0.0787 | 30 |

**Table 7**
The result of Eq. (2) applying 10% and 50% scorers in Yahoo! music database.

| | 10% Scorers | 50% Scorers | $n$ |
|---|---|---|---|
| Validation data 1 | 2.2822 | −3.3564 | 99 |
| Validation data 2 | 0.7787 | −2.5097 | 100 |
| Validation data 3 | 1.5117 | −2.7789 | 98 |
| Validation data 4 | 1.2240 | −2.8151 | 99 |
| Validation data 5 | 1.9062 | −2.6806 | 100 |
| Validation data 6 | 1.1668 | −4.2173 | 100 |
| Validation data 7 | 0.8519 | −2.9645 | 99 |
| Validation data 8 | 0.9087 | −2.9285 | 100 |
| Validation data 9 | 0.5974 | −3.5976 | 100 |
| Validation data 10 | 1.3910 | −3.5481 | 99 |

### 4.3. Z-Test of validity of our representative reviewers

We determine the validity of our algorithms selection of representative reviewers using the following equation for the z-test of significance.

$$Z = \frac{\overline{X} - \mu}{\sigma/\sqrt{n}}, \tag{2}$$

where $\overline{X}$ is an average rating of the 10% or 50% scorers, $\mu$ is an average rating of each item in each validation dataset, $\sigma$ is the standard deviation for $\mu$, and $n$ is the number of movies that are rated by both 10% and 50% scorers.

To perform the z-test, we consider $\mu$ to be a population and $\sigma$ to be the standard deviation of $\mu$. Our null hypothesis is that $\mu$ is equal to $\overline{X}$, the average rating of each item in the validation dataset, and the hypothesis is that $\mu$ is not equal to $\overline{X}$. The critical region to reject the null hypothesis for significance level 95% is ±1.96, and for 99% it is ±2.58.

Fig. 12 shows each element of Eq. (2). Tables 6 and 7 show the result of Eq. (2) applied to each validation dataset for 10% and 50% scorers. The z-test is generally used for $n$ of at least 20, which is met by all $n$ values in Tables 6 and 7. The z-test results shown in each table have values of 10% scorers positioned between significance levels 95% and 99%. From this, we do not reject the null hypothesis, and we conclude that we can consider 10% scorers as representative reviewers. In other words, a meaningful, small sample of users can well represent all users in a database, and we can use these representative reviewers in recommendation systems.

## 5. Conclusions

We have proposed an algorithm to identify representative users in online societies, and we have validated the proposed algorithm using the Yahoo! music and GroupLens movie databases and per-

forming 10-fold cross-validation and z-tests. Our results show that the proposed algorithm selects sets of 10% or 50% of users who are highly representative, based on the probe data, and they further show that the 10% user sets are more representative than the 50% user sets. In our tests, we have considered the number of users and rated items to ensure good reliability. Our results indicate that, with our approach, well-representative users can be identified among raters in internet social media. Representative users can be useful in many ways. One application is in Web marketing. For example, a company that has developed a new Web application can assess the response of the general public by surveying representative users. Another application is to improve recommendation systems based on the collaborative filtering approach (Bell & Koren, 2007; Billsus & Pazzani, 1998; Choi & Han, 2010; Herlocker, Konstan, Borchers, & Riedl, 1999; Sarwar, Karypis, Konstan, & Riedl, 2000). There is ongoing research to identify meaningful users who affect processes of recommendation to other users (Jung, 2012a). The representative reviewers identified in our approach are likely to be such meaningful users since they well represent the ratings of other users. Some Web recommendation systems based on user ratings have the cold-start problem. This problem occurs when new items are added to a database since they do not have any ratings and are excluded from recommendations (Choi, Ko, & Han, 2012). Representative users can help to alleviate this problem by being employed to evaluate new items.

## Acknowledgements

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

## References

Agarwal, N., Liu, H., Tang, L., & Yu, P. S. (2008). Identifying the influential bloggers in a community. In *Proceedings of the 1st international conference on Web search and Web data mining, WSDM'08* (pp. 207–218).

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

Billsus, D., & Pazzani, M. J. (1998). Learning collaborative information filters. In *Proceedings of the 15th international conference on machine learning, ICML'98* (pp. 46–54).

Cha, J. W., Lee, H. W., Han, Y. S., & Kim, L. (2009). User reputation evaluation using co-occurrence feature and collective intelligence. In *Proceedings of the 3rd international conference on online communities and social computing: Held as part of HCI international 2009, OCSC'09* (pp. 305–311).

Choi, S. M., & Han, Y. S., (2010). A content recommendation system based on category correlations. In *Proceedings of the 2010 5th international multi-conference on computing in the global information, ICCGI'10* (pp. 1257–1260).

Choi, S. M., Ko, S. K., & Han, Y. S. (2012). A movie recommendation algorithm based on genre correlations. *Expert Systems with Applications, 39*, 8079–8085.

Durugbo, C. (2012). Modelling user participation in organisations as networks. *Expert Systems with Applications, 39*, 9230–9245.

Han, Y. S., Kim, L., & Cha, J. W. (2012). Computing user reputation in a social network of Web 2.0. *Computing and Informatics, 31*, 447–462.

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).

Jung, J. J. (2012a). Attribute selection-based recommendation framework for short-head user group: An empirical study by MovieLens and IMDB. *Expert Systems with Applications, 39*, 4049–4054.

Jung, J. J. (2012b). Computational reputation model based on selecting consensus choices: An empirical study on semantic wiki platform. *Expert Systems with Applications, 39*, 9002–9007.

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

Kwak, H., Lee, C., Park, H., & Moon, S. B. (2010). What is twitter, a social network or a news media? In *Proceedings of the 19th international World Wide Web conference, WWW'10* (pp. 591–600).

Sarwar, B. M., Karypis, G., Konstan, J. A., & Riedl, J. (2000). Analysis of recommendation algorithms for e-commerce. In *Proceedings of the 2nd ACM conference on electronic commerce, EC'00* (pp. 158–167).