Journal of Universal Computer Science, vol. 19, no. 4 (2013), 581-599
submitted: 3/8/12, accepted: 20/2/13, appeared: 28/2/13

Integrating Multiple Experts for Correction Process in Interactive Recommendation Systems

Xuan Hau Pham
(Department of Computer Engineering
Yeungnam University, Gyeongsan, Korea
pxhauqbu@gmail.com)

Jason J. Jung[1]
(Department of Computer Engineering
Yeungnam University, Gyeongsan, Korea
j2jung@gmail.com)

Ngoc Thanh Nguyen
(University of Information Technology
Vietnam National University
Ho Chi Minh, Vietnam
thanhnn@uit.edu.vn)

Abstract: User rating is obviously considered to be an important type of feedback information for Interactive Recommendation System (RecSys). The quality and credibility of user ratings will eventually influence the quality of recommendation. However, in the real world, there are usually many inconsistent (e.g., mistakes and missing values) or incorrect user ratings. Therefore, expert-based recommendation framework has been studied to select the most relevant experts regarding a certain item's attribute (or value). This kind of RecSys can i) discover user preference and ii) determine a set of experts based on attributes and values of items. In this paper, we propose a consensual recommendation framework, by integrating multiple experts' ratings, to conduct a correction process which aims at modifying the ratings of other users in order to make the system more effective. Since our work assumes that ratings from experts are assumed to be reliable and correct, we first analyze user profile so as to determine preferences and find out a set of experts. Next, we measure a minimal inconsistency interval (MinIncInt) that might contain incorrect ratings. Finally, we propose solutions to correct incorrect ratings based on ratings from multiple experts. The results show that our solutions can improve both the ratings and the quality of RecSys on the whole.

Key Words: Interactive recommendation systems; RecSys; user preference; experts; incorrect rating; consensus.

Category: H.1.1, H.3.5, I.2.11

1 Introduction

Nowadays, in e-Commerce, there is a massive overload in the number of consumer products. The quantity of products is getting more and more abundant over time and

[1] Corresponding author

there are several varieties of the same product which lure the consumer. Online retail systems are being developed in order to benefit the consumers in several ways. The statistical data related to online retail systems can be referenced in detail in *The Top 500*[2]: Amazon and Netflix are two such online retail systems, which are in the Top 10 list of worldwide online retailers based on online sales in 2011. Consumers face an evident difficulty in choosing products that they may be interested in. Thus we argue that personalization technique can be a suitable choice to deal with the above issue. Also, recommendation systems are the best choice for personalization techniques [Adomavicius et al., 2011; Kazienko and Adamski, 2004; Pham and Jung, 2013; Ricci et al., 2011].

The key to a RecSys lies in a fast and effective customization mechanism for getting the relevant content (e.g., web pages, documents, movies, books, music and so on) from a large information repository in a particular domain application. This content is then retrieved and displayed to the consumer on the RecSys. There are many kinds of recommendation approaches such as collaborative filtering, content-based, knowledge-based, preference-based and expert-based. All of them, except collaborative filtering, try to extract user's preferences. These systems can generate personalized recommendations which are a set of items "potentially" related to user's preferences. These items are shown to help users finalize their decisions (e.g., products to buy, music to listen to, movies to watch, and news to read) [Pham and Jung, 2013].

Generally, a RecSys consists of three parts, namely, users, items, and ratings. Also, the recommendation process includes three steps as mentioned below:

1. Show the list of potential items to be recommended.
2. Select related items that a user is interested in and
3. Collect user rating.

User rating expresses user's opinion toward an item and reflects the relationship between a user and an item. It is obvious that the more highly an item is rated, the stronger is the relationship between that particular user and the item. It means that if a user likes an item (i.e., particular item attribute value) then he will rate that particular item highly or otherwise give it low rating [Pham and Jung, 2013]. User ratings can be used to improve the overall recommendation process. There are two approaches to collect user ratings, namely, implicit and explicit. Implicit rating is inferred automatically from collecting user's interactions with an item. Explicit ratings are obtained from user assigned values on a particular item (e.g., rating scale). Explicit user ratings are discussed in this paper.

Depending on the objectives, developers classify a RecSys into two types of classes, namely, interactive and non-interactive [Hernandez and Gaudioso, 2008]. Interactive systems focus on collecting and analyzing user interactions. In the recommendation

[2] http://www.internetretailer.com

process, users and systems can interact together (e.g., when a user selects an item, the system may display question asking the user for the reasons which made him/her select that particular item; User can also interact with the system by editing their ratings and so on). The results of interaction will be analyzed and updated. Non-interactive systems also focus on user's interactions but they collect data before the user interacts with the system and therefore implicitly conduct the collection process. Benefits of Interactive RecSys are as mentioned below:

- Systems and users may interact through questions, answers or corrections,
- Systems extract user's preferences based on user transactions
- Systems show items that have the most relevant user's preferences
- User provides ratings and can change their values
- Systems provide the user with comment functions and analyze comments of various users

In a RecSys, there is a user community in which users can share each other's opinions [Hill et al., 1995]. We can classify them into groups according to their preferences. In each such group, we can determine who the experts are and who the normal users are. Experts bring reliability to each other. An expert in a particular field is generally a person who has high knowledge regarding that field and has also contributed to that same field [Yeung et al., 2011]. In commerce, an expert is a consumer who is well-informed and provides reliable ratings about a certain item. In a RecSys, an expert is a user who has a dominant preference, a reliable rating, and the capability to share his viewpoints and advices. Expert rating is considered as a kind of expert opinion. Thus, they will show their opinions to other users for advice. Expert-based recommendation framework has been studied to select the most relevant expert in relation to a certain item's attribute (and value). By using item attributes, attribute values and user ratings, a RecSys can construct a user profile and determine user's preference. This is determined by their behavior, attitude, interactions and the overall opinion of the member community. We can extract a set of experts based on preferences. A RecSys will consider the relationship between experts and other users based on preferences. It can extract all of them to find out items' relevance in order to recommend it to the user. For example, if you like to watch *Drama* genre movies, a set of experts related this preference will be extracted. They will suggest the best movies that they watched and recommend the same to you. This will reduce the number of recommended items and ensure their quality.

These systems use a set of experts to make suggestions and support systems for user analysis processes. Also, expert opinions are objective and reliable. For each related movie belonging to a certain preference, there is a set of experts. After watching the movie, experts will give their ratings. However, opinions of experts regarding a particular movie may differ. Thus, we are faced with the problem of how integrating all such ratings in order to get the consensus value.

Consensus method is described in [Nguyen, 2008a; Nguyen, 2008b]. It has been applied in case of data conflicts where there are several different values or multiple choices [Nguyen, 2007; Nguyen, 2009]. Consensus can be considered for a set of values. Consensus finds the consent without user agreement and solves conflicts. For example, if the user has selected a rating among several values (e.g., from 1 to 5), the system will find a final congregate value for evaluating the item. Integrating method is best explained as getting the best value by combining multiple values. These values belong to the same range [Keszler and Sziranyi, 2012].

To improve the performance of the recommendation process, most of the recommendation systems should collect better ratings from users. Particularly, rating process is an important task in interactive RecSys because it can ask users to correct their own ratings. It consists of selecting ratings from users, analyzing and correcting ratings from systems. Ratings are an important type of feedback and therefore have been used as keywords in several systems. We have two resources for ratings: one is from the expert and is called as expert rating; other is from the user and is called as user rating. Expert ratings are reliable and correct. On the other hand, there are still many user ratings that are inconsistent and unreliable [Amatriain et al., 2009a; Amatriain et al., 2009b; Embarak and Corne, 2011; Pham and Jung, 2013].

In this paper, we assume that the set of experts considered give reliable and correct ratings. Based on their ratings, we hereby propose some approaches to correct the incorrect ratings. Fig. 1 shows the overview our problem. We can exploit user's preferences to find out a set of experts. We use it along with minimal inconsistency interval (MinIncInt) to check all items which have matching user's preferences in order to determine incorrect ratings. User's preference is determined using dominant attributes and values without considering user rating. It means that the quality of the ratings does not influence measuring preferences.

The result of our proposal is about correcting wrong ratings. An obvious question could be regarding what significant corrected ratings are. As we know, the quality of rating influences the quality of a RecSys. Also, ratings are a part of user interaction. Thus, when a system obtains corrected ratings, they will be used to improve the recommendation process for future usage of the system. The corrected ratings are also used to update the user profile.

The outline of paper is organized as follows. In Sect. 2, we present previous approaches used in related work. In Sect. 3, we explain our framework and define our problem. In Sect. 4, we propose approaches to get better corrected ratings. In Sect. 5, the experimental results are presented and discussed. Finally, in Sect. 6, we gather the major conclusions and future work.

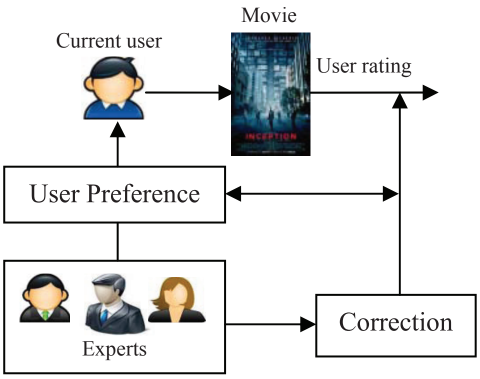

Figure 1: Rating correction process

2 Related work

Using reliable and consistent ratings will improve overall RecSys quality. The significance and role of ratings have been discussed in many previous papers related to RecSys [Ricci et al., 2011].

Besides, the discussion about noisy or inconsistent ratings was also debated. In [Embarak and Corne, 2011], the authors have considered noisy ratings and their affects on estimations and predictions in a RecSys. They use positive feedback to measure user's level of confidence and classify a user into either being honest or dishonest based on region of rejection and non-rejection for each user. They try to discover noisy ratings in order to isolate its impact. In [Amatriain et al., 2009a; Amatriain et al., 2009b], the authors discuss about noisy ratings and propose a method to solve this problem. They ask the system user to rate previously rated items again. However, it is a fact that sometimes users do not care about items they have already rated and hence do not want to spend more time to correct their ratings. This could be one of the reasons for generating noisy and incorrect ratings

In our previous work [Pham and Jung, 2013], we have discussed about incorrect ratings and proposed a method to correct given ratings based on user's preference. In our scheme, system helped users by correcting their ratings without their consent. We extracted user's preferences and recognized incorrect ratings and also ratings which do not need correction. We used a threshold as barrier to find out incorrect ratings. Corrected rating was measured by averaging all other ratings. In our previous work [Pham and Jung, 2013], we have discussed about incorrect ratings and proposed a method to correct given ratings based on user's preference. In our scheme, system helped users by correcting their ratings without their consent. We extracted user's preferences and recognized incorrect ratings and also ratings which do not need correction. We used a threshold as barrier to find out incorrect ratings. Corrected rating was measured by averaging all other ratings.

As discussed in [Adomavicius et al., 2011], ratings can be considered as a key user-recommendation interactions. Here a user is asked to rate consumed items and their ratings are then estimated and presented as predicted ratings. The authors compare predicted ratings and actual ratings and arrive at a conclusion that bias between ratings is still existent. They exploited the impact of RecSys in order to understand the influence of ratings on users' preferences. In RecSys, factors influencing the quality of user ratings have to be considered. In [Riedl et al., 2010], authors have discussed about ratings and a rating scale. They discovered the influence of the rating scale on user ratings. In [Cosley et al., 2003; Hill et al., 1995], community and interface factors have been presented without the use of a rating scale factor [Jung, 2010].

Experts and expertise terms have been discussed in a RecSys. In [Cremonesi et al., 2012; Yeung et al., 2011], the authors have used Spear algorithm to rank users into user communities. They have also extracted the relationship among users, items and tags to determine the expert. This approach can discover malicious users and incorrect tags based on experts. In [Kim and Kim, 2001], the authors have defined an expert group which contains experts who have high authority and much expertise. They have assumed that expert knowledge is better than other forms of knowledge. A list of web documents based on rating of experts was made and the top ranked document was recommended. In the next section we will present our framework and discuss about extracting user's preferences and method for finding incorrect ratings.

3 Expert-based recommendation framework

In above section, we have defined an expert and provided some surveys about user rating. In this section, we will discuss a set of main properties of the expert and their contributions to a RecSys. We will also consider the following properties; i) expertise (interesting values), ii) rating quality and iii) frequency. As an example, one of the methods to recognize expertise is to compute the number of movies that experts have watched in relevance to a certain preference. For example, if you are an expert about *Steven Spielberg* you should have watched a lot of movies directed by him. Other factors such as rating quality may also be considered. An expert's rating is said to have a good quality if there are many users who concur with his rating (i.e., their rating has the same value as expert's one). The frequency is defined using the total number of times a user visits a system and the number of ratings and the duration of each session.

A general RecSys expresses the relationship between a user, an item and its rating. It shows relevant items to the user, after which the user selects and rates items; finally the given ratings are used in future [Jung, 2010; Jung, 2011a]. A user is considered either as a consumer, a partner or a member of a community. Items consist of movies, books, web pages, documents, photos and so on. The rating types depend on systems

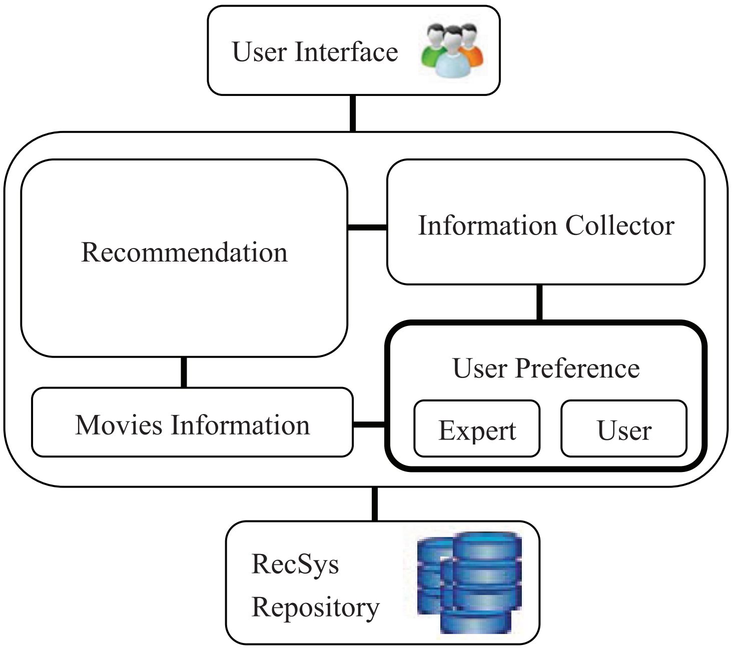

Figure 2: Expert-Based RecSys framework

[Pham and Jung, 2013; Riedl et al., 2010]. They may be ordinal, binary and numerical etc. In this paper, we will use movie recommendations to illustrate our examples.

Definition 1 (Recommendation framework). A generic framework of RecSys is represented as

$$\mathcal{S} = \langle \mathcal{U}, \mathcal{I}, \mathcal{R}, \Omega \rangle \tag{1}$$

where $\mathcal{U}$, $\mathcal{I}$, and $\mathcal{R}$ ate sets of users, items, and user ratings, respectively. Also, $\Omega \subseteq \mathcal{U} \times \mathcal{I} \times \mathcal{R}$ can indicate a matrix of

The main goal of these systems is to find out user's preferences. There are many kinds of methods to discover user's preferences. In this paper, we define an expert-based recommendation framework. The extracting user's preferences are based on attributes and attribute values of item which the user watched/purchased.

Definition 2 (Expert-based recommendation framework). Extended from the previous recommendation framework, an expert-based RecSys can be represented as

$$\mathcal{S}_{\mathcal{E}} = \langle \mathcal{U}, \mathcal{I}, \mathcal{R}, \mathcal{A}, \mathcal{V} \rangle \tag{2}$$

where $\mathcal{A}$ is a set of attributes of the item $\mathcal{I}$, and $\mathcal{V}$ is a set of values of the attribute $\mathcal{A}$.

Table 1: User-movie profile

Movie ID (title)	Genre	Actor	Director	Rating
i_1 (Dark Angel)	Action, Drama, Mystery	J. Alba, M. Weatherly	J. Cameron, C. H. Eglee	5
i_2 (True Lies)	Action, Thriller	A. Schwarzenegger, J. L. Curtis	J. Cameron	5
i_3 (Titanic)	Drama, Adventure, History	L. DiCaprio, K. Winslet	J. Cameron	2
i_4 (Avatar)	Sci-Fi, Action, Adventure, Fantasy	S. Worthington, Z. Saldana	J. Cameron	5
i_5 (Black Swan)	Drama, Mystery, Thriller	N. Portman, M. Kunis	D. Aronofsky	4

In the movie recommendation system, each user will have a user-movie profile (i.e., a user profile) which contains a list of movies, information regarding the movies and user ratings on each item. User's preference can be understood as a set of values that the user has been interested in and is extracted from user interactions. In this paper, in order to determine user's preference, we have to find out the dominant value and attribute on each profile. We have discussed methods to extract dominant values and attributes in [Jung, 2011b; Jung, 2012a; Jung, 2013]. The structure of user's preference has also been presented in [Pham and Jung, 2013]. For example, if a user likes to watch *Action* movies then the dominant attribute is *Genre*, dominant value is *Action* and user's preference will contain *Genre, Action*. Table 1 shows user-movie information. For example if we consider user u_1, he has watched five movies and accordingly we have movies information and ratings on each movie. We can show that *Director* and *J. Cameron* are the dominant attribute and dominant value on attribute, respectively. Thus, we can conclude that the preference of u_1 is $Pref(u_1) = \langle Director, J.Cameron \rangle$.

We also focus on determining which expert is based on dominant preference. For example, if a user only watches movies that have *action* genre and his ratings are reliable, then we can consider that he is an expert about *action* movie. It is easy to find that his preference is a tuple defined as $\langle Genre, Action \rangle$.

We can discover a set of experts based on user's preferences. It means that there is the correlative preference between user and each expert in a set.

Definition 3 (Experts). Given a set of user $\mathcal{U}$, the experts are users who have their own dominant preferences and reliable ratings. The set of experts E_u are represented as

$$E_u = \{e | e \in \mathcal{U}, \psi(e, u) > 0\} \tag{3}$$

where function $\psi(e, u)$ can measure the correlative preference between user u and expert e.

For each user u, we will find out a set of experts and define R_E which represent a set of ratings of E_u. Therefore we have:

$$R_E = \{r_e | e \in E_u, r_e \in R\} \tag{4}$$

Let $I_u^p = \{i_1, ..., i_k\}$ be a set of items and $R_u^p = \{r_1, ..., r_k\}$ be a set of ratings of user u where the ratings on an item belong to user's preference, respectively. In Fig. 1, correction process is presented as follows: current user selects and rates a movie, assuming that we have extracted a preference and this movie belongs to his preference. According to our hypothesis, if he likes this movie, he will give it a good rating. But in this situation, his rating is 1 and therefore we predict that his given rating is incorrect. We have to correct it based on ratings from three other experts. In order to discover incorrect ratings we have to find out MinIncInt. In our scheme, we will extend consistency functions that are presented in [Nguyen, 2008a] to measure it.

Definition 4 (Preference profile). The preference profile arises as a result of projection of user's preference onto the user profile as shown below:

$$T(u) = \{(i, r) | i \in I_u, r \in R_u; (a, v)_i \in Pref(u)\} \tag{5}$$

where I_u is a set of items of user u, $a \in \mathcal{A}, v \in \mathcal{V}$ and function $Pref(u)$ is to determine the corresponding preference of the user u. Function $Pref(u)$ [Pham and Jung, 2013] is given by

$$pref(u) = \{(a, v_a) | a \in \tau(A), v_a \in V\} \tag{6}$$

which consists of a set of pairs between item attributes and their values.

Preference $T(u)$ can contain conflicts. It means that there are a few ratings that are incorrect or inconsistent. In [Nguyen, 2008a], consistency functions are used to measure the conflict profile based on the distance among elements of the profile. Based on $T(u)$, we can get a set of user ratings that have items belonging to a preference as follows:

$$R_u^p = \{r | r \in T(u)\}$$

and based on R_u^p, we calculate the matrix of distances among ratings as shown below:

$$M(u) = [d(r_i, r_j)] \tag{7}$$

where $k = card(R_u^p), r_i, r_j \in R_u^p$ and $i, j = \overline{1, k}$

$$d(r_i, r_j) = \begin{cases} 0 & \text{if } i = j \\ |r_i - r_j| & \text{if } i \neq j \end{cases} \tag{8}$$

We determine the vector of mean distance between the ratings to the rest:

$$W^u = (w_1, ..., w_k) \tag{9}$$

where

$$w_i = \frac{1}{k-1}\sum_{j=1}^{k} d(r_i, r_j) \tag{10}$$

We will determine the value of MinIncInt for the ratings that are incorrect by refined classification using the below mentioned steps:

1. MinIncInt is assigned to [1..5] interval.
2. Sort R_u^p.
3. Determine $d_{mean} = \frac{1}{k}\sum_{i=1}^{k} w_i$ to classify R_u^p into one of two classes:

Class 1:

$$RP_u^1 = \{r_i | w_i \leq d_{mean}\} \tag{11}$$

Class 2:

$$RP_u^2 = \{r_i | w_i > d_{mean}\} \tag{12}$$

4. Loop step 2 to refine class 1 and class 2 until RP_u^1 and RP_u^2 cannot be classified any more or the interval does not change.
5. We will then get MinIncInt that contain incorrect ratings.

For example, from the Table 1 we can get:

$T(u) = \{(i_1, 5), (i_2, 5), (i_3, 2), (i_4, 5)\}$,

$MinIncInt = [1..5], R_u^p = (5, 5, 5, 2)$

The vector of mean distance is given by $W^u = (1, 1, 3, 1)$, and $d_{mean} = 1.5$

We have $RP_u^1 = \{5, 5, 5\}$ and $RP_u^2 = \{2\}$ and $MinIncInt = [1, 2]$, corresponding with movie i_3. It means that the rating belonging to this interval may be incorrect. In the next section we will present approaches to get correct ratings.

4 Correction process

In previous sections, we have presented ways to determine user's preferences, experts and a set of experts. We have also explained method to find incorrect ratings. Therefore, after we have determined which given rating is incorrect, we can correct it. The set of ratings contain discrete ratings and is distributed in interval [1..5]. In this paper, the correction process will rely on expert ratings on an item. As mentioned above, expert ratings are correct and reliable. We propose four solutions to solve this problem, namely, best matching, majority rating, weighting and maximal consensus.

4.1 Best Matching

We know that experts and user have similar preference (i.e., preference will overlap among a sets of items). Matching method is an approach used to find the proximity between current user and experts based on the same items. It means that we need to find the closest expert amongst a set of experts based on the same item. In this method, we will find an expert e who has the highest amount of the same items with user u.

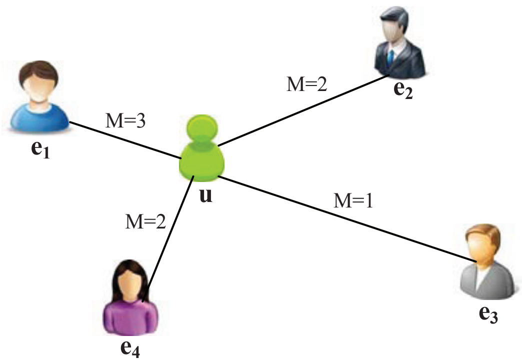

Figure 3: Best Matching

We will get matching items between e and u. We predict that if there is an expert who has watched movies that have high proximity with the user, then this expert is most influential to that user. It means that if user u and expert e have the same amount of items is higher compared with other experts, u will select e rating. The best matching method does not consider user's rating and experts' ratings. It just finds the proximity amongst them based on their set of items. The matching function is defined as follows:

Definition 5 (Matching). Matching between user u and expert e is expressed as follows:

$$\mathcal{M}(u, e) = Matching(I_u, I_e)$$

where the function $Matching(I_u, I_e)$ gets the number of matching items between e and u; I_u, I_e are set of items of user u and expert e, respectively.

If we can find out an expert who has the best matching with user u, it is easy to get correct rating to replace incorrect ratings in order to replace the incorrect ratings by using the expert ratings on that item.

$$r_{u,i} = \{r_{e,i} | \mathcal{M}(u, e) > \mathcal{M}(u, e'), \forall e' \in E - \{e\}\} \tag{13}$$

Fig. 3 illustrates our approach. For example, we consider the following sets of ratings of user u and experts e_1, e_2, e_3, and e_4, respectively. We assume that the user rating on item i_2 is incorrect.

$I_u = \{i_1, i_2, i_5, i_6\}$
$I_{e_1} = \{i_1, i_2, i_3, i_5, i_9\}$
$I_{e_2} = \{i_2, i_6, i_7, i_8, i_9\}$
$I_{e_3} = \{i_1, i_3, i_4, i_7, i_8\}$
$I_{e_4} = \{i_1, i_3, i_5, i_7, i_9\}$

Table 2: Example for majority rating

expert	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8	e_9	e_{10}	e_{11}
rating	5	4	4	3	5	5	4	3	4	5	4

We obtained the following results:
$\mathcal{M}(u, e_1) = |\{i_1, i_2, i_5\}| = 3, \mathcal{M}(u, e_2) = |\{i_2, i_6\}| = 2,$
$\mathcal{M}(u, e_3) = |\{i_1\}| = 1, \mathcal{M}(u, e_4) = |\{i_1, i_3\}| = 2.$
Therefore, the corrected rating is: $r_{u,i_2} = r_{e_1}$

4.2 Majority Rating

As we know, a set of experts are people whose preference is similar. Hence, when experts rate an item, expert ratings are convergent. It means that there are several experts who have been rating the item with the same score. A majority of experts are consensus and this rating would be selected. Majority rating has been considered for determining user credibility [Zaki and Bouguettaya, 2009].

In RecSys, majority rating has been used to determine the consensus based on a combination of expert ratings. As defined in Def. 3, ratings have five classes (i.e., rating on a scale from 1 to 5). Each expert has only one unique rating on each item. It is easy to identify an expert belonging to a certain class. In order to select dominant rating from experts, we have to find the occurrence of each rating class as follows:

$$\Omega(R_E, r) = \{occur(r) | r \in [1..5]\} \tag{14}$$

where r is a rating scale and $occur(r)$ is a function that gives the number of ratings that have been rated on this scale. Also R_E is a set of expert ratings. The result of Ω occurrence function is a vector number defining the occurrence of each rating scale.

The dominant rating is then computed as follows:

$$r_{u,i} = \{r | max(\Omega(R_E))\} \tag{15}$$

where $max(\Omega(R_E)$ returns maximal value of the vector. This value expresses the convergence of experts' rating on this item. It can be assigned to the incorrect rating from current user.

For example, assume that we have determined that a given rating of user u on an item i is incorrect. We have a set of experts consisting of eleven experts as shown in Table 2.

We measure according to the 5 classes as follows:
$\Omega(R_E, 1) = 0, \Omega(R_E, 2) = 0, \Omega(R_E, 3) = 2, \Omega(R_E, 4) = 5, \Omega(R_E, 5) = 4$
Therefore, we get: $r_{u,i} = 4$

4.3 Weighting

We see that the best matching and majority rating methods are solutions for rating correction that only consider dense distribution of the same items or rating. Best matching method finds the nearest expert in the set of experts and uses his rating. Majority rating finds a scaled class to which the largest experts belong to. It is easy to get the results when deviations among measurements are quite different.

Thus, we propose another approach which we name as weighting by measurement of the similarity between expert e and user u. *Similarity* is a general term used to express equivalence on the set of users or set of items in a RecSys. Similarity function measures the weight of relationships among two elements in the same set or in two different sets. If the weight is high then the relationship is close or otherwise. This way we can find the influence of experts with respect to user u and reconcile among experts. We will construct the similarity vector between user and experts. In this approach, we do not consider the nearest expert, because we want to measure the harmony amongst experts. It means that every expert has the same role when their ratings are considered for making judgments about user rating. The similarity vector is combined with the set of expert ratings and the overall satisfaction of experts to obtain the result.

The similarity is defined as follows:

$$sim(e,u) = \frac{\sum_{i \in I_u, I_e} (r_{e,i} - \overline{r}_e)(r_{u,i} - \overline{r}_u)}{\sqrt{\sum_{i \in I_u, I_e} (r_{e,i} - \overline{r}_e)^2} \sqrt{\sum_{i \in I_u, I_e} (r_{u,i} - \overline{r}_u)^2}} \tag{16}$$

where

I_u and I_e are a set of items of user u and a set of items of expert e, respectively.

$\overline{r}_u$ and $\overline{r}_e$ are average user ratings and average expert ratings, respectively.

The similarity vector between users and experts in constructed by:

$$F_E = \{sim(e,u) | \forall e \in E\}$$

when $\forall f_e \in F_E$, the corrected rating is determined as follows:

$$r_{u,i} = \frac{1}{|E|} (\sum_{e \in E} f_e . r_e) + \beta_{E,i} \tag{17}$$

where $\beta_{E,i}$ is the function which measures the satisfaction of experts with respect to item i and $\beta_{E,i} \in [0..1]$. The satisfaction has been discussed in order to survey user opinion about specific applications [Bailey and Pearson, 1983; Zhang and von Dran, 2000]. In this approach, we consider expert satisfaction; this factor expresses opinion of experts about certain items that they watch and rate. Hence, satisfaction of expert will be expressed by using the expert rating. We assume that if expert rates on a scale from 3 to 5, they are satisfied with this movie. Otherwise, if an expert rates a movie on a scale of 1 to 2, then they are unsatisfied with the movie. For example, the movie *Titanic* has been watched and rated by 10 experts in which 7 of them rate the movie on a scale from 3 to 5 and 3 of them rate the same movie on a scale of 1 to 2. Thus, we calculate the satisfaction of expert as 0.7

Suppose the following example

$$E = \{e_1, e_2, e_3\}, R_E = \{4, 3, 5\}$$

we have

$$sim(e_1, u) = 0.8, sim(e_2, u) = 0.9, sim(e_3, u) = 0.7$$

And

$$F_E = \{0.8, 0.9, 0.7\}, \beta_{E,i} = 1$$

Therefore the corrected rating is: $r_{u,i} = \frac{1}{3}(\sum_{e \in E} f_e.r_e) + 1 = 4$

4.4 Maximal consensus

In this method, we only use a set of expert ratings to find consensus based on determining the dominant and consistency interval [Nguyen et al., 2012]. Maximum consensus presents a maximal agreement. We know that ratings in an experts' set are different. We will apply the same method that we used to find MinIncInt in Sect. 3. However, we will now find the maximum interval instead of finding minimum interval and denote it by $MaxConInt$. Values in this interval are concentrated distribution values and majority values.

Using the steps used for determining minimal inconsistency interval in Sec. 3, we can find out maximal interval in similar way as follows:

$$MaxConInt = \{r | r \in R_u \wedge r \notin MinIncInt\} \tag{18}$$

The corrected rating will be measured as follows:

$$r_{u,i} = \frac{\sum_{r \in MaxConInt} r}{K} \tag{19}$$

where $K = \{card(R_E) | r \in R_E \wedge r \in R_{mci}\}$

For example, assume that for an item i_5, the rating of u is incorrect. We can then extract a set of expert ratings as follows: $R_E = \{5, 2, 4, 1, 5, 5, 2\}$. Using minimal inconsistency interval explained in Sec. 3 and Equa. 18, we will get the maximal consistency interval $MaxConInt = [4, 5]$. The corrected rating $r_{u,i5}$ is therefore 4.5.

Table 3: Statistics on Datasets

Dataset	#user	#rating	#movie
DRS1	50	4196	1800
DRS2	200	14396	1912

Table 4: Statistics on ratings

	Rating Scale				
	5	4	3	2	1
DRS1	1262	1428	1033	306	167
DRS1	3366	5208	4065	1206	551

5 Experimental results

In previous section, we have presented the correction process. We have shown four approaches to correct wrong ratings. In this section, we will show experimental results that we have obtained.

To implement our proposal, we have used a dataset from MovieLens[3]. Also, in order to determine user's preferences, we have combined information about movies from IMDB[4]. In our implementation, we have used two datasets which are composed of DRS1 and DRS2. In Table 3, we can see that DRS1 has an average of 84 ratings per each user. With DRS2, we have 72 ratings per each user. We have statistical listing of the ratings in Table 4.

We have obtained the results as follows: in DRS1 dataset, we obtained 39 users who have incorrect ratings and number of incorrect ratings was 121. In DRS2 dataset, we have determined 136 users who have incorrect ratings and number of incorrect rating was 378.

We have used Root Mean Squared Error (RMSE) to measure the accuracy of corrected ratings. RMSE is measured using the deviation of incorrect ratings and corrected ratings. Fig 4 and Fig. 5 show RMSE of correction on our datasets. In these figures, we can see that in some cases, the system can find incorrect ratings, but cannot find experts who watched the same movie and therefore RMSE is equal to zero. In some cases, number of experts in a set is only one and therefore corrected rating will be assigned using the rating of this expert. If RMSE is high, it means that there is a set of experts for this user on an item that the system has to correct. If RMSE is low, it means that the set of experts is empty or this user has many incorrect ratings but some of them cannot be corrected because experts did not watch these movies.

[3] http://www.movielens.org
[4] http://www.imdb.com

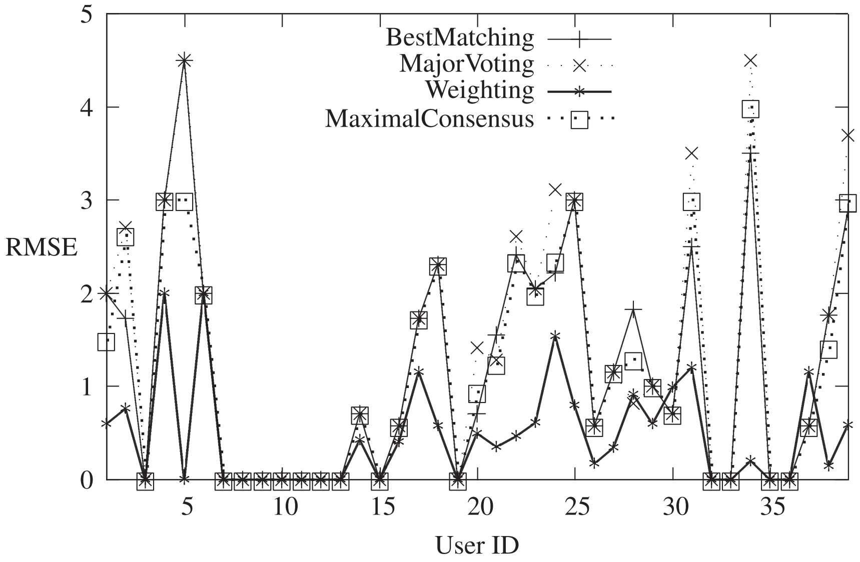

Figure 4: RMSE of correction on DRS1

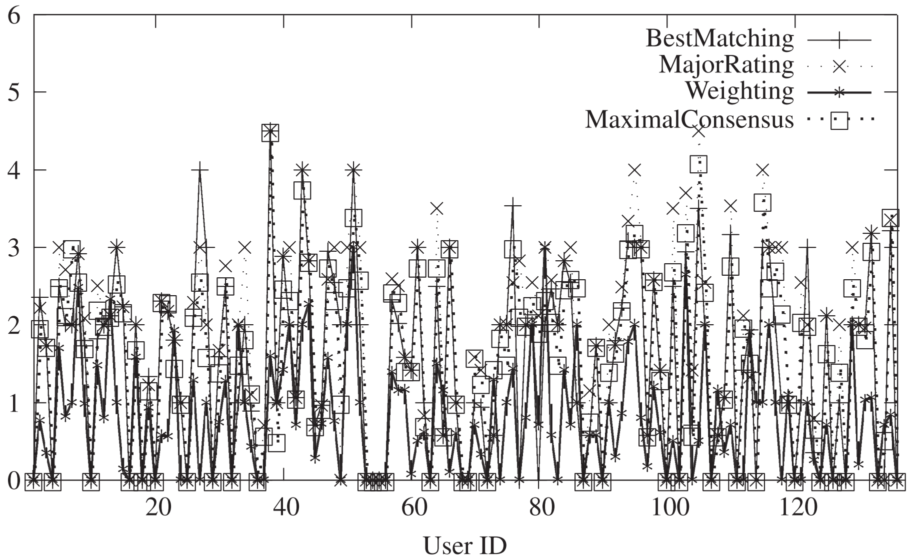

Figure 5: RMSE of correction on DRS2

6 Conclusions and future work

There have been many kinds of recommendation systems. They have tried to find the most unbiased and reliable recommendations for as many users as possible. Different from them, the main assumption of this work is that some users can have biased ratings and they can be regarded as a set of experts. Also, we have assumed that expert opinions (i.e., expert ratings) are reliable and correct, while the ordinary users may give incorrect ratings. Hence, in this paper, we have proposed several approaches to solve this problem. We have made the expert-based recommendation framework to determine the incorrect ratings and to find a set of experts based on user's preferences. We have then used ratings from experts to correct wrong ratings. We have considered all expert ratings and integrated them to get better ratings using the following approaches: 1) The best matching approach considers the relationship between user and expert based on the matching of set of items; 2) The majority rating approach considers the dense distribution on the set of expert ratings; 3) The weight approach finds the harmony amongst experts; 4) The maximal consensus determines the consensus of experts based on the convergent rating interval.

In this paper, we have implemented the proposed system on two datasets. However, the number of users in datasets is still small. There were some mistakes when we combined data from MovieLens and IMDB. Hence, it was difficult to determine user's preference and find experts in some cases. We have only focused on correcting that has not yet been considered in order to put experts into recommendations where list of recommended items are presented based on experts' opinions. Moreover, in the real world case, we have to consider that even experts can make some mistakes on rating items, and to find more reliable solution for the problems. Related to the later, we will concentrate on recommendation based on experts' ratings and user's preferences. The expert-based methods can be considered along with criticism of experts. It will also be applied to other products such as music, books, photos and so on. The quality of recommendation may be better if we can find out a set of experts and a list of ranked experts. The list of ranked experts will be made by measuring factors such as endorsement, contribution and satisfaction. Moreover, in application aspect, for dealing with a large amount heterogeneous data [Jung, 2012c], we have to carefully consider mashup-based tools [Jung, 2012b]. Finally, we have to consider more standard evaluation schemes, e.g., Interclass Correlation Coefficients (ICC) method and Kendall's coefficient and other benchmark methods to prove the improvement of the proposed system.

Acknowledgement

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government (MEST) (No. 2011-0017156). This research was partially funded by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant no. 102.01 - 2011.10.

References

[Adomavicius et al., 2011] Adomavicius G., Bockstedt J., Curley S., Zhang J.: Recommender Systems, Consumer Preferences, and Anchoring Effects. In Proceedings of the 3rd ACM conference on Recommender systems (RecSys'11), pp.35-42, 2011.

[Amatriain et al., 2009a] Amatriain X., Pujol J.M., Oliver N.: I like it... i like it not: Evaluating user ratings noise in recommender systems. In Houben G.J., McCalla G.I., Pianesi F., Zancanaro M. (eds). Proceedings of the 17th International Conference on User Modeling, Adaptation, and Personalization (UMAP'09), volume 5535 of Lecture Notes in Computer Science, pp. 247-258. Springer, 2009.

[Amatriain et al., 2009b] Amatriain X., Pujol J.M., Tintarev N., Oliver, N.: Rate it again: increasing recommendation accuracy by user re-rating. In Bergman L.D., Tuzhilin A., Burke R.D., Felfernig A., Schmidt-Thieme, L. (eds). Proceedings of the 3rd ACM conference on Recommender systems (RecSys'09), pp. 173-180, ACM, New York, NY, USA, 2009.

[Bailey and Pearson, 1983] Bailey J.E., Pearson S.W.: Development of a tool for measuring and analyzing computer user satisfaction. Management Science 29(5): 530–545, 1983.

[Cosley et al., 2003] Cosley D., Lam S.K., Albert I., Konstan J.A., Riedl J.: Is seeing believing? How recommender system interfaces affect users opinions. In: Cockton G., Korhonen P., (eds) Proceedings of the 2003 SIGCHI conference on human factors in computing systems (CHI'03), pp. 585-592, 2003.

[Cremonesi et al., 2012] Cremonesi P., Garzotto F., Turrin R.: User effort vs. accuracy in rating-based elicitation. In Cunningham P., Hurley N.J., Guy I., Anand S.S. (eds). Proceedings of the 6th ACM Conference on Recommender Systems (RecSys '12), pp. 27-34, ACM, Dublin, Ireland, 2012.

[Embarak and Corne, 2011] Embarak O.H., Corne D.W.: Detecting vicious users in recommendation systems. In Proceedings of the 4th International Conference on Developments in e-Systems Engineering (DESE 2011), pp. 339-344, 2011.

[Hernandez and Gaudioso, 2008] Hernández del Olmo F., Gaudioso E.: Evaluation of recommender systems: A new approach. Expert Systems with Applications, 35(3):790–804, 2008.

[Hill et al., 1995] Hill W., Stead L., Rosenstein M., Furnas G.: Recommending and evaluating choices in a virtual community of use. In Proceedings of ACM Conference on Human Factors in Computing Systems (CHI'95), pp. 194-201, 1995.

[Jung, 2010] Jung J.J.: Reusing Ontology Mappings for Query Segmentation and Routing in Semantic Peer-to-Peer Environment. Information Sciences, 180(17):3248–3257, 2010.

[Jung, 2011a] Jung J.J.: Ubiquitous Conference Management System for Mobile Recommendation Services Based on Mobilizing Social Networks: a Case Study of u-Conference. Expert Systems with Applications, 38(10):12786-12790, 2011.

[Jung, 2011b] Jung J.J.: Service Chain-based Business Alliance Formation in Service-oriented Architecture. Expert Systems with Applications, 38(3):2206-2211, 2011.

[Jung, 2012a] Jung J.J.: Attribute selection-based recommendation framework for short-head user group: An empirical study by MovieLens and IMDB. Expert Systems with Applications, 39(4):4049–4054, 2012.

[Jung, 2012b] Jung J.J.: ContextGrid: A Contextual Mashup-based Collaborative Browsing System. Information Systems Frontiers, 14(4):953–961, 2012.

[Jung, 2012c] Jung J.J.: Evolutionary Approach for Semantic-based Query Sampling in Large-scale Information Sources. Information Sciences, 182(1):30–39, 2012.

[Jung, 2013] Jung J.J.: Cross-lingual Query Expansion in Multilingual Folksonomies: a Case Study on Flickr. Knowledge-Based Systems, 42:60–67, 2013.

[Kazienko and Adamski, 2004] Kazienko P., Adamski M.: Personalized Web Advertising Method. In Proceedings of the 2004 International Conference on Adaptive Hypermedia, pp. 146-155, 2004.

[Keszler and Sziranyi, 2012] Keszler A., Sziranyi T.: A mixed graph model for community detection. International Journal of Intelligent Information and Database Systems 6(5):479–494, 2012.

[Kim and Kim, 2001] Kim D.E., Kim S.W.: Dynamic Models of Expert Groups To Recommend Web Documents. In Proceedings of the 5th European Conference on Research and Advanced Technology for Digital Libraries (ECDL'01), pp. 275-286, 2001.

[Nguyen, 2007] Nguyen N.T.: A Method for Ontology Conflict Resolution and Integration on Relation Level. Cybernetics and Systems 38(8):781-797, 2007.

[Nguyen, 2008a] Nguyen N.T.: Advanced Methods for Inconsistent Knowledge Management (Advanced Information and Knowledge Processing), Springer-Verlag, 2008.

[Nguyen, 2008b] Nguyen N.T.: Inconsistency of Knowledge and Collective Intelligence. Cybernetics and Systems 39(6):542-562, 2008.

[Nguyen, 2009] Nguyen N.T.: Processing Inconsistency of Knowledge in Determining Knowledge of a Collective. Cybernetics and Systems 40(8):670-688, 2009

[Nguyen et al., 2012] Nguyen T.M., Kawamura T., Tahara Y., Ohsuga A.: Self-supervised capturing of users' activities from weblogs. International Journal of Intelligent Information and Database Systems 6(1):61-76, 2012.

[Pham and Jung, 2013] Pham X.H., Jung J.J.: Preference-based user rating correction process for interactive recommendation systems. Multimedia Tools and Applications, 65(1): 119-132, 2013.

[Ricci et al., 2011] Ricci F., Rokach L., Shapira B., Kantor P.B.: Recommender systems handbook. Springer US, 2011.

[Riedl et al., 2010] Riedl C., Blohm I., Leimeister J.M., Krcmar H.: Rating scales for collective intelligence in innovation communities: Why quick and easy decision making does not get it right. In Proceedings of the International Conference on Information Systems (ICIS'10), pp. 101-109, 2010.

[Yeung et al., 2011] Yeung C.M.A., Noll M.G., Meinel C., Gibbins N., Shadbolt N.: Measuring expertise in online communities. IEEE Intelligent Systems, 26(1):26–32, 2011.

[Zaki and Bouguettaya, 2009] Zaki M., Bouguettaya A.: RATEWeb: reputation assessment for trust establishment among web services. The VLDB Journal 18(4): 885-891, 2009.

[Zhang and von Dran, 2000] Zhang P., von Dran G.M.: Satisfiers and dissatisfiers: a two-factor model for Website design and evaluation. American Society for Information Science 51(14):1253-1268, 2000.