Expert Systems with Applications 39 (2012) 4049–4054

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

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

# Attribute selection-based recommendation framework for short-head user group: An empirical study by MovieLens and IMDB ☆

Jason J. Jung

*Department of Computer Engineering, Yeungnam University, Dae-Dong, Gyeongsan 712-749, Republic of Korea*

ARTICLE INFO

*Keywords:*
User modeling
Recommender system
Long-tail group
Rough set
Attribute reduction

ABSTRACT

Most of recommender systems have serious difficulties on providing relevant services to the "short-head" users who have shown intermixed preferential patterns. In this paper, we assume that such users (which are referred to as long-tail users) can play an important role of information sources for improving the performance of recommendation. Attribute reduction-based mining method has been proposed to efficiently select the long-tail user groups. More importantly, the long-tail user groups as domain experts are employed to provide more trustworthy information. To evaluate the proposed framework, we have integrated MovieLens dataset with IMDB, and empirically shown that the long-tail user groups are useful for the recommendation process.



## 1. Introduction

Efficient recommendation has been regarded as a key technology of various applications. In particular, all possible activities and feedbacks from online users should be collected, and efficiently analyzed to find out any meaningful relationships between users, between items, and between users and items. Thereby, most of such recommendation systems have been focusing on user modeling for comparing the users. A variety of user modeling methods (Jung, 2008a; Symeonidis, Nanopoulos, & Manolopoulos, 2007; Zhang & Koren, 2007) have been proposed for analyzing many types of user behaviors. In collaborative filtering methods for recommendation, user ratings are useful for modeling the corresponding users' preferences. For example, as shown in Table 1, suppose that 4 users have rated 6 movies. By measuring similarity measures (e.g., Pearson correlation coefficient) between two arbitrary users, we can find that $U_4$ is most similar to $U_1$. Hence, once $U_1$ has rated a good score (e.g., 5) to $M_5$, the system can automatically recommend $M_5$ to $U_4$.

Most of the recommendation systems are interested in only the group of users who have rated enough number of items (e.g., movies). We refer to the users as a *short head* group (i.e., $U_1$, $U_2$, and $U_4$). Consequently, it makes the systems possible to compare each set of ratings to the others. In contrast, recommending the rest of users like $U_3$ is a serious challenge. These users, referred to as a *long tail* group, have rated only a few items and also peculiar items. Thus, it is even more difficult for the existing recommendation systems to find useful relationships between them. Due to the lack of user ratings from the long tail group, it is difficult to identify which item they are interested in. Also, more seriously, the chance to justify whether two users has common preferences is getting lower.

In fact, there have been some studies for extracting and also using the long tail. Most similarly, in Sathe, Lee, and Giuse (2004), a power user group has been selected for solving difficult problems in a specific medical domain. The users are assumed to have unique and outstanding knowledge and information. Meanwhile, a variety of methods, e.g., semantic query (Jung, 2008b), fuzzy logic (John & Mooney, 2001) and information integration (Jung, 2009a), have been proposed. Especially, Brusilovsky, Sosnovsky, Yudelson, Kumar, and Hsiao (2008) has tried to merge several partial (and inexact) user models collected from multiple information systems.

In this paper, we focus on exploiting auxiliary information from external sources for identifying user preferences in the long tail group. For example, as shown in Table 2, if addition information about the movies is obtained, we can recognize that $U_2$ and $U_3$ have rated the only movies directed by 'James Cameron' ($M_3$ and $M_5$) and 'Christopher Nolan' ($M_2$ and $M_6$), respectively. Thus, once the system has new movie directed by the same directors, the movie will be recommended to appropriate users in the long tail group.

However, it is a problem to determine which attributes are working properly to discriminate the user preferences in the long tail (Jung, 2009b). In order to deal with the problem, we propose a novel data integration framework to aggregate all available data sources for modeling the long tail groups more efficiently. Particularly, we focus on establishing several heuristics for measuring statistical patterns of the attributes. As a simple example in the previous Table 2, given two ratings by $U_3$ (i.e., $M_2$ and $M_6$),

☆ This paper is significantly revised from an earlier version presented at the Third International Conference on Computational Collective Intelligence in September 2011.

*E-mail addresses:* j2jung@intelligent.pe.kr, j2jung@ynu.ac.kr, j2jung@gmail.com

0957-4174/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2011.09.096

**Table 1**
Movie ratings by users.

| User ID | Movie ID (Title) | Rating |
|---|---|---|
| $U_1$ | $M_1$ (Robin Hood) | 5 |
| | $M_2$ (Inception) | 2 |
| | $M_3$ (Avatar) | 4 |
| | $M_4$ (Devil) | 3 |
| | $M_5$ (Titanic) | 5 |
| $U_2$ | $M_3$ (Avatar) | 1 |
| | $M_5$ (Titanic) | 2 |
| $U_3$ | $M_2$ (Inception) | 5 |
| | $M_6$ (Memento) | 3 |
| $U_4$ | $M_1$ (Robin Hood) | 5 |
| | $M_2$ (Inception) | 2 |
| | $M_4$ (Devil) | 3 |

an attribute 'Directors' has shown the most dominant coverage $\left(\text{i.e.}, \frac{|\text{Directors(Christopher Nolan)}|}{2} = 1\right)$, compared to any other attributes $\left(\text{e.g.}, \frac{|\text{Genre(Action)}|}{2} = 0.5\right)$. The user can be recommended if there is an additional movies directed by "Christopher Nolan" as well as classified in "Action".

In sum, we note that two main goals of this framework are

- to find which attribute is significant to identify long tail groups, and
- to exploit the experts of long tail groups for better recommendations to short head groups.

The outline of this paper is as follows. In the following Section 2, we explain the background knowledge about long tails, and show several existing work based on the long tail groups. Section 3 introduces several definitions for modeling users in long tail group, and present several heuristics for selecting significant attributes during data integration. Section 4 shows an experimental results for evaluating the proposed attribute selection methods and recommendation performance. Especially, Section 4.1 describes a case study on integrating MovieLens with IMDB datasets for identifying long tail groups. Finally, Section 5 draws our conclusions of this work.

## 2. Backgrounds and related work

In fact, the concept of long tail has found many different areas including online business, mass media, micro-finance, user-driven innovation, and social network mechanisms (e.g., crowdsourcing, crowdcasting, peer-to-peer), economic models, and marketing (viral marketing) (Jung, 2010c). The main assumption of this paper is that a group of users in long tail should be regarded as the professional experts in corresponding domains, and employed their opinions to provide recommendations to users in short head. In this section, we want to explain the background knowledge and previous studies about (i) how to represent user interests and (ii) how to recommend users.

### 2.1. Attribute-based user modeling

There have been a number of user modeling approaches, e.g., Bayesian network, neural networks and fuzzy logic (Kok, 1991). Most recently, with emergence of semantic web communities, ontologies (or concept hierarchies, e.g., web directories) have been exploited to derive relevant ontological elements (e.g., concepts, properties, and instances) (Jung, 2007, 2008a).

One of the simplest ways is to predefine a set of attributes, which are basis to measure the relevance to the user preferences. Also, the weight of each attribute can be computed to indicate the degree of the relevance to the user. If we assume that the attribute set $A$ is a finite set, then this user model can be regarded as a vector form.

**Definition 1** (*User*). Given a user rating $\mathcal{R}_i = \{\langle A, r\rangle | r \in [1,5]\}$, a user $u_i$ is simply represented as a set of attributes.

$$u_i = \left\{\langle a, w_a\rangle | a \in A_i, w_a = \frac{\sum r_a}{occur(a)}\right\} \quad (1)$$

用户可以表示成属性，
权重模型。

where $A_i$ is a finite set of attributes and $w_a$ is a weight of an attribute $a$.

For example, in Table 1, the attributes can be a set of genres of movies, e.g., Action, Drama, and so on. The users can be represented as a set of genres into which the movies are classified. As a matter of fact, the more important issue is adaptability of user models by updating the weight of each attribute over time. Thus, this adaptive user modeling can capture the temporal changes of user interests, because the users have shown dynamic preference depending on many situations.

However, it is difficult to represent a number of attributes which are related with each other. As shown in Table 2, by nature, we can easily understand more meaningful relationships between the attributed.

### 2.2. Recommendation by user classification

Once the users have been modeled, the recommendation system has to compare the user models for measuring similarities between two users. This is an essential assumption that most of systems are considering for providing recommendations. Thus, many social filtering methodologies and applications have been presented (Fisk, 1997; Sarwar, Karypis, Konstan, & Riedl, 2001; Su & Khoshgoftaar, 2009). Especially, such approaches have been employed to distributed environments, e.g., adaptive learning community (del Olmo, Gaudioso, & Boticario, 2003; Jung, 2011a).

As shown in previous section, there are a number of user modeling methods. Of course, this comparison process is based on how the user model consists of. If we assume that all user models are represented as a set of attributes, then we can measure the similarity by using several heuristics, as follows.

**Definition 2** (*Similarity*). Given two users models $u_i$ and $u_j$, we can compute a similarity between them.

**Table 2**
Information sources about movie with three additional attributes.

| Movie ID (Title) | Genre | Actors | Directors |
|---|---|---|---|
| $M_1$ (Robin Hood) | Action, Adventure, Drama | Russell Crowe, Cate Blanchett | Ridley Scott |
| $M_2$ (Inception) | Action, Mystery, Sci-Fi, Thriller | Leonardo DiCaprio, Ellen Page | Christopher Nolan |
| $M_3$ (Avatar) | Action, Adventure, Fantasy, Sci-Fi | Sam Worthington, Zoe Saldana | James Cameron |
| $M_4$ (Devil) | Horror, Mystery, Thriller | Chris Messina, Logan Marshall-Green | John Erick Dowdle |
| $M_5$ (Titanic) | Drama, History Romance | Leonardo DiCaprio, Kate Winslet | James Cameron |
| $M_6$ (Memento) | Crime, Drama, Mystery, Thriller | Guy Pearce, Carrie-Anne Moss | Christopher Nolan |

$$S(n_i, n_j) = \frac{|N|}{\max(|u_i|, |u_j|)} \quad (2)$$

$$= \frac{\max_{k=1}^{|N|} n_k}{|N|} \quad (3)$$

$$= \frac{\sum_{k=1}^{|N|} n_k}{|N|} \quad (4)$$

where $N = \left\{ \langle p, w_p \rangle | p \in u_i, p \in u_j, w_p = \frac{p_{u_i} + p_{u_j}}{2} \right\}$.

Finally, these similarities between two users are applied to conduct user classification processes (e.g., k-means). A large number of users can be efficiently managed for propagating new recommendations.

## 3. Long-tail user group selection

The attribute-based user modeling is efficient and easy to implicitly represent user preferences from user activities and feedbacks, only if scopes of the user preferences are identically limited. The attributes of a user should be selected from a predefined attribute set, so as to be matched with that of the other user (Jung, 2011b).

However, if the users' feedbacks are not covered with the predefined attribute set, it is not possible for the user models to be compared with each other. To solve the problem, additional information can be exploited to capture the scopes of as many users as possible. Thus, there is even more chance of extract additional attributes from user behaviors.

For example, as shown in Table 2, the movie has many different attributes to be evaluated. When users are asked to rate the movies they watched, each of them might have different subjective criteria to determine whether the movies are good or not. 主观判断

Thus, in this paper, we focus on Long Tail user Groups (LTuG) whose activities are more concentrated into a smaller set of attributes. Opposite to LTuG, Short Head user Groups (SHuG) have shown so diverse activities that the number of attributes are higher. In order to discriminate these two user groups and select the LTuG, we need to integrate external information (e.g., Table 2).

### 3.1. Dominant attributes

主导因素

In this paper, we want to figure out what is main motivation to take a certain action. In case of watching movies, ones have different opinions to choose movies. Two users $U_1$ and $U_3$ have watched the same movie $M_2$ in Table 1. We can find out that $U_3$ has chosen $M_2$ because the director is "Christopher Nolan." If there is a new movie by the same director, the system has to recommend the movie to him. In this context, "Christopher Nolan" is a dominant attribute to represent his user model.

Hence, two methods can be presented in this paper to select the dominant attributes. First method is to measure a "dominant coverage" score of each attribute from the user ratings for efficiently justifying which attribute is strongly related to the user preference.

**Definition 3** (*Dominant coverage*). Given a user $u_i$, a dominant coverage $\tau$ of each attribute can be measured by

$$\tau_i(A) = \mu_{a \in A}\left(\frac{occur(a, \mathcal{R}_i)}{|\mathcal{R}_i|}\right) \quad (5)$$

这个观众关注什么样的元素？
元素出现的次数/评价的总指标。

where $\mathcal{R}_i$ is a set of ratings by the user, and $\mu$ is a function to compute a mean value.

More importantly, we do not want to consider how the users rate the items. Even though some of users have decided to rate movies with low scores, they may be interesting on some attributes related to the movies.

As shown in Table 3, the dominant coverage of each item, which is composed of an attribute, can be measured without taking into account how the users rate the items. Then, finally, we can compute the dominant coverage of the attribute by aggregating the information about the items. For example, attribute 'Director' of $U_1$ can be assigned with

$$\tau_i(Director) = \frac{0.2 + 0.2 + 0.4 + 0.2}{4} = 0.25 \quad (6)$$

where the 'Director' consists of four items. Roughly, we can understand that $U_2$ and $U_3$ have shown strong interests on 'Director' attribute (which are J. Cameron and C. Nolan, respectively), while $U_4$ does not have any dominant attribute explicitly.

Once we obtained the dominant coverage of attributes, we formulate two possible heuristics for extracting dominant attributes.

**Definition 4** (*Dominant attribute*). A dominant attribute of a user $U_i$ is selected when its dominant coverage is significantly larger than the others. Hence, a set of dominant attributes can be represented as the following two methods

$$A_i^{\tau} = \{A_j | \max_{A_k \in \mathcal{A}} \tau(A_k)\} \quad (7)$$

$$= \{A_j | \tau(A_k) \geqslant \mu_{A \in \mathcal{A}}(\tau(A_i))\} \quad (8)$$

where $\mathcal{A}$ is a set of all attributes.

As shown in Table 4, the first heuristic (i.e., Eq. (7)) allows to choose the only attribute whose dominant coverage is maximum. Unfortunately, this heuristic cannot discover any dominant attributes for $U_4$, because the user's ratings are evenly distributed on all items (i.e., no significant patterns to be discovered). On the other hand, the second heuristic (i.e., Eq. (8)) can employ a mean value as a threshold for filtering out. This heuristic can more sophisticatedly extract the dominant attributes that the first one.

Next, the second method of extracting dominant attributes is to investigate a temporal change by measuring variance during collecting the user ratings. We expect that a variance of dominant coverage for a certain duration will be playing an important role for detecting the dominant attributes. As a user keeps rating more items over time, his interests will be expressed more clearly. Thus, it is important to realize how the preferences of a user are stable over time. Especially, in an initial stage, it is difficult to precisely measure the dominant coverage (i.e., the first heuristic), due to the lack of user ratings at the moment. It is also called the cold start problem.

Thus, the coverage variance of each attribute $\rho$ can be formalized.

**Definition 5** (*Coverage variance*). Given a set of ratings from a user $U_i$ during $[t_0, t_T]$, the coverage variance can be given by

$$\rho_i(A)_{[t_0, t_T]} = var_{t=t_0}^{t=t_T}\left(\tau_i^{(t)}(A)\right) \quad (9)$$

where $A \in A_i^{\tau}$ and *var* is a function to computer a variance value.

The coverage variance can be exploited to justify whether a certain dominant attribute is being converged or not. The given time duration is segmented into several intervals, and the coverage variance should be measured in each time interval. For example, suppose that the number of time intervals is 2. We only consider that the attribute $A$ is a dominant attribute of user $U_i$, if $\rho_i(A)_{[t_0, t_1]} \leqslant \rho_i(A)_{[t_1, t_2]}$. Finally, the set of dominant attributes is denoted as $A^{\tau+\rho}$.

Once we have computes these two measurements (i.e., $\tau$ and $\rho$) of each attribute from the users, we can justify who the long tail user groups are.

**Definition 6** (*Long tail user group*). Given a number of users and their ratings, a long tail group *LTuG* is represented as

**Table 3**
An example of computing the dominant coverages of attributes.

| Users | Genre | $\tau(Genre)$ | Actors | $\tau(Actors)$ | Director | $\tau(Director)$ |
|---|---|---|---|---|---|---|
| $U_1$ | Action | 0.176 | R. Crowe | 0.1 | R. Scott | 0.2 |
| | Adventure | 0.118 | C. Blanchett | 0.1 | C. Nolan | 0.2 |
| | Drama | 0.059 | L. DiCaprio | 0.2 | J. Cameron | 0.4 |
| | Mystery | 0.118 | E. Page | 0.1 | J.E. Dowdle | 0.2 |
| | Sci-Fi | 0.118 | S. Worthington | 0.1 | | |
| | Thriller | 0.118 | Z. Saldana | 0.1 | | |
| | Fantasy | 0.059 | C. Messina | 0.1 | | |
| | Horror | 0.059 | L. Marshall-Green | 0.1 | | |
| | History | 0.059 | K. Winslet | 0.1 | | |
| | Romance | 0.059 | | | | |
| $U_2$ | Action | 0.143 | L. DiCaprio | 0.25 | J. Cameron | 1 |
| | Adventure | 0.143 | S. Worthington | 0.25 | | |
| | Drama | 0.143 | Z. Saldana | 0.25 | | |
| | Sci-Fi | 0.143 | K. Winslet | 0.25 | | |
| | Fantasy | 0.143 | | | | |
| | History | 0.143 | | | | |
| | Romance | 0.143 | | | | |
| $U_3$ | Action | 0.125 | L. DiCaprio | 0.25 | C. Nolan | 1 |
| | Drama | 0.125 | E. Page | 0.25 | | |
| | Mystery | 0.25 | G. Pearse | 0.25 | | |
| | Sci-Fi | 0.125 | C.-A. Moss | 0.25 | | |
| | Thriller | 0.25 | | | | |
| | Crime | 0.125 | | | | |
| $U_4$ | Action | 0.2 | R. Crowe | 0.167 | R. Scott | 0.33 |
| | Adventure | 0.1 | C. Blanchett | 0.167 | C. Nolan | 0.33 |
| | Drama | 0.1 | L. DiCaprio | 0.167 | J.E. Dowdle | 0.33 |
| | Mystery | 0.2 | E. Page | 0.167 | | |
| | Sci-Fi | 0.1 | C. Messina | 0.167 | | |
| | Thriller | 0.2 | L. Marshall-Green | 0.167 | | |
| | Horror | 0.1 | | | | |

**Table 4**
An example of selecting the dominant attributes from Table 3.

| Heuristics | Eq. (7) | Eq. (8) |
|---|---|---|
| $U_1$ | {Director = "J. Cameron"} | {Director = "J. Cameron"} |
| $U_2$ | {Director = "J. Cameron"} | {Actors = ★, Director = "J. Cameron"} |
| $U_3$ | {Director = "C. Nolan"} | {Director = "C. Nolan"} |
| $U_4$ | {} | {Director = ★} |

$$LTuG = \left\{U_i | A_i^{\tau+\rho} \neq \phi\right\} \tag{10}$$

where $\phi$ indicates an empty set.

### 3.2. Recommending short head users

We are regarding the LTuG as an expert group who is strongly interested in particular attributes (e.g., a director and an actor). In this paper, we want to exploit them to help to recommend the short head user group (SHuG).

The recommendation system needs to provide relevant information about a certain attribute to the short head user group. Thus, the basic idea of recommending the SHuG is to reuse the user rating given by the LTuG. To provide $U_j \in$ SHuG with the information about an attribute $A_q$, we conduct the following steps;

1. We can retrieve a set of users from LTuG, whose dominant attributes include $A_q$, by justifying the following condition

$$A_q \in A_j^{\tau+\rho} \tag{11}$$

where it returns boolean value.

2. The ratings of the SHuG can be integrated with the ratings from a set of users from LTuG, as follows;

$$\mathcal{R}_j \leftarrow \mathcal{R}_j + (\cup_i \mathcal{R}_i) \tag{12}$$

where it returns a whole rating set.

The integrated user ratings are simply assumed to be applied to normal recommendation schemes (e.g., collaborative filtering). We want to show that the performance of the normal recommendation scheme can be improved by the integration process to the SHuG.

## 4. A case study and experimental results

In this section, we want to describe how we have evaluated the proposed LTuG-based recommendation system. As a case study, movie recommendation has been chosen for applying the proposed method. To do so, we have collected MovieLens dataset[1] as user ratings, and investigated a set of attributes from Internet Movie Database.[2]

### 4.1. A case study on MovieLens

We have selected 8 attributes from IMDb, as follows.

- Genre = {Action, Adventure, Animation, Biography, Comedy, Crime, Documentary, Drama, Family, Fantasy, Film-Noir, Game-Show, History, Horror, Music, Musical, Mystery, News, Reality-TV, Romance, Sci-Fi, Sport, Talk-Show, Thriller, War, Western}.
- Director (i.e., multiple directors are allowed).
- Actors (i.e., maximum 5 top actors have been selected from the list).

[1] http://movielens.umn.edu/.
[2] IMDb, http://www.imdb.com/.

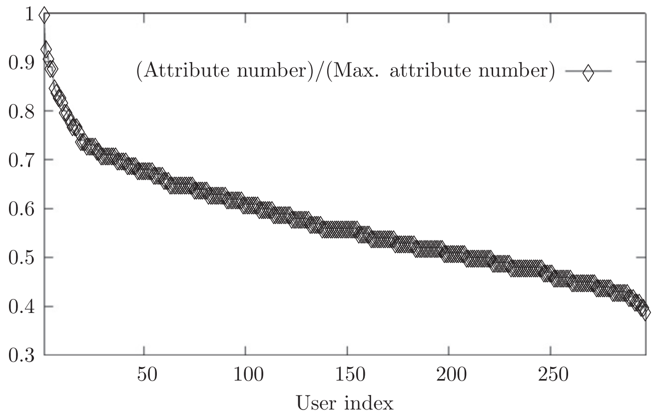


**Fig. 1.** A long tail users from MovieLens dataset.

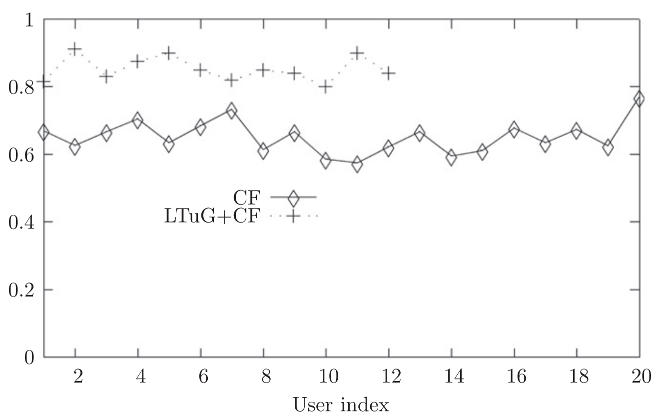


**Fig. 2.** A long tail users from MovieLens dataset.

- Storyline (i.e., maximum 5 keywords have been selected by TF-IDF analysis).
- Plot keywords.
- Production company.
- Country.
- Language.

It is important to confirm whether the user rating dataset can contain the long tail user groups in the real case. Fig. 1 shows a long tail phenomenon from user rating patterns on MovieLens items. We have found out that (i) attribute "director" is the most important dominant attribute, and (ii) about 17.8% of users have been regarded as a long tail user group.

### *4.2. Experimental results and discussion*

We have invited 20 graduated students (i.e., $U_1$ to $U_{20}$) to provide two types of recommendations and also to get their feedbacks about the recommendations. Such recommendation types are (i) collaborative filtering (denoted as CF) and (ii) the proposed long tail user group-based collaborative filtering (denoted as NTuG+CF). This experimentation had been conducted during 25 days for monitoring the temporal changes of dominant attributes.

Fig. 2 demonstrates the experimental results. Regarding the establishment of the long tail user group, 8 users (i.e., $U_{13}$ to $U_{20}$) out of 20 have been selected as a long tail user group. This result (40%) is about twice higher than the result from the case study of MovieLens. We have found that 2 students have rated only Korean movies and it has made the dominant coverage of "Country" attribute much higher than the others.

More importantly, performances of the recommendation given by both methods (i.e., CF and LTuG+CF) have been compared. We measured the precision by the following equation.

$$\text{Precision} = \frac{\text{Number of positive feedbacks from the user}}{\text{Number of recommended movies}} \tag{13}$$

As a result, LTuG+CF has outperformed CF by about 30.7% higher precision. It proves that the user ratings of the selected user group (i.e., long tail users) can obviously help to provide relevant recommendation to the short head user group.

## 5. Conclusion and future work

As a conclusion, we have proposed a novel user modeling method for long tail users. The main contribution of this work is to efficiently establish the long tail users who can be regarded as expert group on a certain attribute. Hence, the user ratings and feedback given by this long tail user groups have been exploited to provide more relevant recommendation to non-expert user group, called short head group. Moreover, as additional contribution, we have shown the data integration scheme (e.g., MovieLens and IMDb) which can extract the meaningful but hidden attributes for user modeling.

In future work, dynamic patterns of user ratings should be more studied, because user ratings are a kind of streaming data over time Jung, Lee, and Choi, 2009. Also, there are a number of different patterns to be investigated (Jung, 2010a). Again, we want to collect more case studies to consider the long tail phenomena in the context of recommendation. Another important issue is to integrating information from heterogeneous sources. We are expecting to collect real ontologies from ontology-based recommendation systems (Jung, 2010b), and integrate them for better user modeling.

## Acknowledgment

This work was supported by the National Research Foundation of Korea (NRF) Grant funded by the Korean government (MEST) (No. 2011-0017156).

## References

Brusilovsky, P., Sosnovsky, S., Yudelson, M., Kumar, A., & Hsiao, S. (2008). User model integration in a distributed adaptive e-learning system. In S. Berkovsky, F. Carmagnola, D. Heckmann, T. Kuflik & A. Krüger (Eds.), *Proceedings of the 6th international workshop on ubiquitous user modeling* (pp. 1–10).
Fisk, D. (1997). An application of social filtering to movie recommendation. In H. S. Nwana & N. Azarmi (Eds.), *Software agents and soft computing: towards enhancing machine intelligence, concepts and applications. Lecture Notes in Computer Science* (Vol. 1198, pp. 116–131). Springer.
John, R. I., & Mooney, G. J. (2001). Fuzzy user modeling for information retrieval on the world wide web. *Knowledge and Information Systems, 3*(1), 81–95.
Jung, J. J. (2007). Ontological framework based on contextual mediation for collaborative information retrieval. *Information Retrieval, 10*(1), 85–109.
Jung, J. J. (2008a). Ontology-based context synchronization for ad-hoc social collaborations. *Knowledge-Based Systems, 21*(7), 573–580.
Jung, J. J. (2008b). Query transformation based on semantic centrality in semantic social network. *Journal of Universal Computer Science, 14*(7), 1031–1047.
Jung, J. J. (2009a). Consensus-based evaluation framework for cooperative information retrieval systems. *Knowledge and Information Systems, 18*(2), 199–211.
Jung, J. J. (2009b). Trustworthy knowledge diffusion model based on risk discovery on peer-to-peer networks. *Expert Systems with Applications, 36*(3), 7123–7128.
Jung, J. J., Lee, H., & Choi, K. S. (2009). Contextualized recommendation based on reality mining from mobile subscribers. *Cybernetics and Systems, 40*(2), 160–175.
Jung, J. J. (2010a). Ontology mapping composition for query transformation on distributed environments. *Expert Systems with Applications, 37*(12), 8401–8405.

Jung, J. J. (2010b). Reusing ontology mappings for query segmentation and routing in semantic peer-to-peer environment. *Information Sciences, 180*(17), 3248–3257.

Jung, J. J. (2011a). Service chain-based business alliance formation in service-oriented architecture. *Expert Systems with Applications, 38*(3), 2206–2211.

Jung, J. J. (2011b). Boosting social collaborations based on contextual synchronization: An empirical study. *Expert Systems with Applications, 38*(5), 4809–4815.

Jung, J. J. (2010c). Integrating social networks for context fusion in mobile service platforms. *Journal of Universal Computer Science, 16*(15), 2099–2110.

Kok, A. J. (1991). A review and synthesis of user modelling in intelligent systems. *Knowledge Engineering Review, 6*(1), 21–47.

del Olmo, F. H., Gaudioso, E., & Boticario, J. (2003). A multiagent approach to obtain open and flexible user models in adaptive learning communities. In P. Brusilovsky, A. T. Corbett, & F. de Rosis (Eds.), *Proceedings of 9th international conference on user modeling (UM 2003), Johnstown, PA, USA, June 22–26. Lecture Notes in Computer Science* (Vol. 2702, pp. 203–207). Springer.

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

Sathe, N. A., Lee, P., & Giuse, N. B. (2004). A power information user (piu) model to promote information integration in tennessee's public health community. *Journal of Medical Library Association, 92*(4), 459–464.

Su, X., & Khoshgoftaar, T. M. (2009). A survey of collaborative filtering techniques. *Advances in Artificial Intelligence, 2*(4).

Symeonidis, P., Nanopoulos, A., & Manolopoulos, Y. (2007). Feature-weighted user model for recommender systems. In C. Conati, K. F. McCoy, & G. Paliouras (Eds.), *Proceedings of the 11th international conference on User Modeling*. *Lecture Notes in Computer Science* (Vol. 4511, pp. 97–106). Springer.

Zhang, Y., & Koren, J. (2007). Efficient bayesian hierarchical user modeling for recommendation system. In W. Kraaij, A. P. de Vries, C. L. A. Clarke, N. Fuhr, & N. Kando (Eds.), *Proceedings of the 30th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 47–54). New York, NY, USA: ACM.