The Eighteenth International Symposium on Artificial Life and Robotics 2013 (AROB 18th '13),
Daejeon Convention Center, Daejeon, Korea, January 30-February 1, 2013

On a serendipity oriented recommender system based on folksonomy

Hisaki Yamaba[1], Michihito Tanoue[1], Kayoko Takatuka[1], Naonobu Okazaki[1] and Sigeyuki Tomita[1]

[1] University of Miyazaki, Miyazaki, Miyazaki 889-2192, Japan
(Tel: 81-985-58-7425, Fax: 81-985-58-7425)
(yamaba@cs.miyazaki-u.ac.jp)

Abstract: This paper proposes a recommendation method that focuses on not only predictive accuracy but also serendipity. On many of the conventional recommendation methods, each item is categorized according to their attributes (a genre, an authors, etc.) by the recommender in advance, and recommendation is performed using the categorization. In this study, impressions of user to items are adopted as a feature of the item, and each item is categorized according to the feature. The impressions are prepared by using folksonomy. A recommender system based on the method was developed by java language, and the effectiveness of the proposed method was verified through recommender experiments.

Keywords: recommender system, serendipity, folksonomy

1 INTRODUCTION

In these days, importance of recommender systems which can present useful information to users is increasing because numerous information comes to be broadcasted according to the expansion of the Internet.

A lot of researches have been investigated about recommentation systems; many of them were realized based on the collaborative filtering method or the contents based filtering method [1]. The collaborative filtering method recommends items such that users who have similar tastes with an active user (a user who is given recommendation) like. On the other hand, the content based filtering method recommends items that are similar with items that the active user likes.

Recommender systems based on these conventional approaches focus on the predictive accuracy. However, development of recommender systems that consider measure that go beyond the accuracy of its recommendation is attempted in recent years [1].

This study proposes a method for serendipitous recommendation such that users feel surprised by recommended books. A recommender system was developed by Java language and the validity of the proposed method was confirmed through a series of recommendation experiments.

2 PROPOSED METHOD

2.1 Serendipity oriented recommendation

The purpose of recommender systems is to recommend items that are useful for users. However, items that are suit users' tastes are not useful in case that the users are familiar with the items. This means that recommender systems are required to recommend items that not only suit tastes of users but also are novelty (they are unknown to the users) [1]. In these days, serendipity also comes to be required to recommender systems. The word serendipity is created by Horace Walpole based on the fairy tale titled "Three princes of Sarendip." In general, this word means the ability to find something good or useful while not specifically searching for it. However, in the area of research in recommender systems, this word means that recommended items are unforeseeable, unexpected, or surprising for the user [1] . When a user searches an item that suits to his tastes, the search will be performed around the area that the item is expected. Therefore, it is supposed that the user cannot find such an item that does not locates in the expected area. On the contrary, if the user finds such a book by accident, it will be a serendipitous discovery for the user. In this study, it is assumed that a user feels serendipity when a recommended item suits the taste of the user, and is unknown to him/her, and is not included in the area that the user expected such items locate.

Under many of the conventional recommendation methods, items for recommendation (e.g. books) are categorized according to their attributes (a genre, an authors, etc.) by a recommender in advance. Recommendation is performed using the categorization. For example, books that are classified into mystery novels will be recommended to users who love mystery novels. However, a book that is classified into love stories will not be recommended to such users even if the book has flavor of mystery novels (Fig.1). However, such books might be serendipitous books for the kind of users (Fig.2).

In this study, impressions of users on books are adopted as one of attributes of the books. For example, if users feel a flavor of mystery novels from a book, which is not regarded as a mystery novel but regarded as a love story, the impression will be added to the characteristics of the book. It is expected that such books will be recommended to mystery fans by using impressions of users in the process of recommendation. Serendipity in this study means such ability to

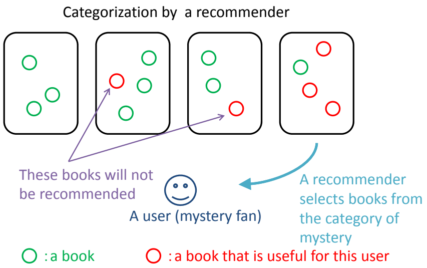
Categorization by a recommender
These books will not be recommended
A user (mystery fan)
A recommender selects books from the category of mystery
○ :a book
○ :a book that is useful for this user

Fig. 1. Conventional recommendation.

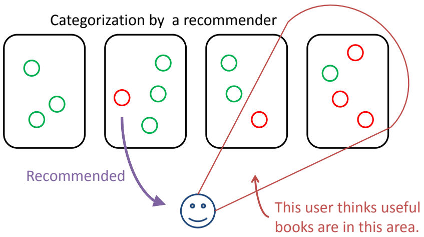
Categorization by a recommender
Recommended
This user thinks useful books are in this area.

Fig. 2. Serendipity on recommendation.

find out useful books beyond the conventional books classification. In order to realize the books classification mentioned above, folksonomy was adopted.

2.2 Folksonomy

Folksonomy is a bottom up style classification system while conventional classification systems adopt a top down style. Under such conventional systems, items presented to users are classified in advance based on categories defined by service providers. Folksonomy classifies items using tags that are given by users. Tags are keywords that are generated by users following characteristics, impressions and so on of each item. Users are allowed to select any words as tags. And also, users are allowed to give more than two tags to a book. Folksonomy has features shown below:

- The classification reflects impressions or recognition of users.
- It is easy for users to give tags to items because what users have to do is only to input keywords; some special knowledge is not required.
- Classification results are flat, not stratified.

2.3 Introduction of the idea called ”concepts”

In cases that classification is performed using tags that are attached to items, problems caused by synonym or polysemy have to be resolved. For example, suppose a tag “blog” is given to an item and a tag “weblog” is attached to another item. Though they have tags that indicate same meanings, the two items are not recognized to have same feature.

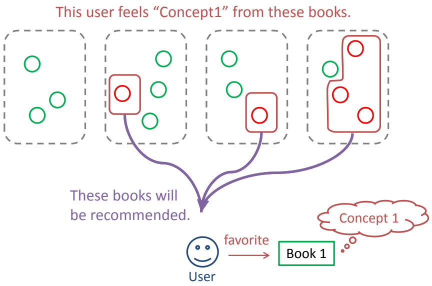
This user feels “Concept1” from these books.
These books will be recommended.
favorite
Book 1
Concept 1
User

Fig. 3. Basic idea of the proposed method.

In this study, impressions of users themselves are used to classify items. Since an impression that each user feels on a book is not known explicitly, a method to infer such impressions from tags data was developed based on an assumption that tags are selected according to impressions users feel on books. An inferred impression is called a “concept” in this study (Fig.3).

2.4 Proposed method

In this section, books recommendation is used to explain the proposed method (recommendation of books is also used in the experiments mentioned in the next section).

In the proposed method, a concept is expressed in the form of a vector of degrees of relevance from the concept to tags (see 2.4.3). At first, in a case that two tags are given to same books many times, it is assumed that the impression of the users who gave the one of the two tags is same with the impression of users who gave the other tag. Under this assumption, concepts are generated according to the steps as follows. First, “degrees of similarity” between two tags were calculated (see 2.4.1). Next, similar tags are gathered and clusters are generated (see 2.4.2). Such a cluster corresponds to a concept.

Using concepts obtained above, “degrees of relevance” from each book to concepts and “degrees of relevance” from each user to concepts are calculated (see 2.4.4 and 2.4.5). Then, characteristics of each book or each user are represented by a vector of degrees of relevence. The recommender system recommends books that whose characteristics are similar with the tastes of the user.

2.4.1 Degree of similarity between two tags

It is assumed that two tags a and b are used reflecting a same concept in case that tag a and b are both attached to same item in many cases. In this study, such a and b are regarded to be similar. Books are classified into four types listed below with tag a and b:

(I) both a and b are attached to the book,

(II) neither a or b is not attached to the book,

(III) only a is attached to the book and b is not.

(IV) only b is attached to the book and a is not.

In case that a sum of percentages of (I) and (II) is large, the similarity between a and b is assumed to be high. On the other hand, the similarity between a and b is assumed to be low when a sum of percentages of (III) and (IV) is large. In this study, "a degree of similarity" between two tags is represented by AEMI (Augmented Expected Mutual Information) [2].

$$
\begin{aligned}
AEMI(\ \ ,\ \ ) \quad = & \sum_{(A=\alpha,B=\beta),(A=\bar{\alpha},B=\bar{\beta})} MI(A,B) \\
& \sum_{(A=\bar{\alpha},B=\beta),(A=\alpha,B=\bar{\beta})} MI(A,B)
\end{aligned}
$$

where is cases that tag a is attached to books and is cases that tag b is attached to books. $^{-}$ represents cases that tag a is not attached to books. $MI(A,B)$ is mutual information that measures co-occurrence of A and B:

$$MI(A,B) = P(A,B)\log\frac{P(A,B)}{P(A)P(B)}$$

where $P(A)$ is the occurrence frequency of A, and $P(A,B)$ is the concurrence frequency of A and B.

2.4.2 Tag clustering

This section explains the process to generate clusters that are composed of similar tags.

1. An empty set $Cset$ is prepared, which is used as the set of created clusters.

2. All tag pairs (Tag_i,Tag_j) ($i \neq j$) are created and sorted in the order of their similarity. Then tag pairs that have higher similarity than the threshold V_t introduced in advance are selected and stored in a list.

3. Tag pairs in the list are processed as follow in the higher order of their similarities:

   (a) A copy of $Cset$ is created ($CopySet$).

   (b) Clusters that include both Tag_i and Tag_j are removed from $CopySet$.

   (c) For each cluster (Cl_k) in $CopySet$,

   - if Tag_i is not included in Cl_K, do step i,
   - if Tag_j is not included in Cl_K, do step ii0.

     i. A degree of similarity between Cl_k and Tag_i is calculated. Similarity of a tag T and a cluster C is the average of similarities between T and all tags included in C. If the value is greater than V_t, Tag_i is added to Cl_k.

     ii. A degree of similarity between Cl_k and Tag_j is calculated. If the value is greater than V_t, Tag_j is added to Cl_k.

   (d) $Cset$ is updated. Concretely, clusters in $CopySet$ substitute corresponding clusters in $Cset$.

   (e) If $Cset$ does not include a cluster that includes both Tag_i and Tag_j, a new cluster is created that includes two tags (Tag_i and Tag_j) and added to $Cset$.

2.4.3 Creation of concepts

A concept (Co_i) is represented by a vector of degrees of relevance from a cluster (Cl_i) obtained above to all tags. A degree of relevance from a cluster Cl_i to a tag Tag_i is calculated as follows:

$$rel(Tag_i, Co_1) = \frac{t(Tag_i, Cl_1)}{\sum_j t(Tag_i, Cl_j)}$$

where $t(Tag_i, Cl_1)$ is an average of degrees of similarity between Tag_i and all tags in a cluster Cl_i.

2.4.4 Representation of characteristics of books

Characteristics of a book are represented by a vector of degrees of relevance from the book to all concepts obtained above. A degree of relevance from a book $Book_1$ to a concept Co_1 is represented as follows:

$$rel(Book, Co_1) = \sum_i rel(Tag_i, Co_1)$$

2.4.5 Representation of characteristics of users

First, a degree of preference from a user ($User$) to a concept (Co_1) is introduced as follows:

$$pre(User, Co_1) = \sum_i rel(Book_i, Co_1)$$

where $Book_i$ is a book that $User$ likes.

Characteristics of a user is represented by a vector of a $pre(User, Co_1)$ for all concept obtained above.

There are several methods to find books that a user likes: questionnaires, interviews, referring records of web viewing, and so on. In the experiments of this paper, examinees indicate their favorite books in direct.

2.4.6 Selection of books for recommendation

In this study, books whose characteristics vectors are similar with that of a target user are selected for recommendation. Concretely, an inner product of characteristics vectors of a book and a user is adopted here.

3 EXPERIMENTS

A book recommender system that is based on the proposed method was implemented by java language. A series of experiments was carried out using the system in order to confirm the validity of the proposed method.

3.1 Acquisition of data for experiments

Data used in the experiments are collect from Booklog (http://booklog.jp). Booklog is a web service that provides virtual book shelves. Over 500 thousands users are registered at the site and over 33 million items (books, CDs, and so on) are stored as of Jan. 2012. Web pages have been created for every book. Each of the pages provides information about the corresponding book. Each page also includes the links to the pages of the relating books. Booklog adopt folksonomy and its users can attach tags to books arranged on their book shelves.

Books in the top list for 2011 and books linked from them were selected. And 18,922 tags attached to the 6,717 books were obtained.

3.2 Methods of experiments and evaluation

Concepts were generated and specific vectors were calculated from the obtained data by the proposed method. The concepts and the vectors were embedded into the developed recommender system. The system recommended 10 books to each of 50 examinees.

A characteristics vector of each examinee was calculated from favorite books which he/she listed up. Ten books were selected for each examinee according to his/her characteristics vector. Next, the examinees answered the three inquiries listed below about each of the recommended books:

Q1 Are you interested in the book?
(1) Yes. (2) A little. (3) Little. (4) No.

Q2 Do you know the book well?
(1) Yes, I have read it. (2) Yes, but unread.
(3) Only its title. (4) No.

Q3 Do you think the recommendation validates?
(1) Yes. (2) I don’t know. (3) No.

3.3 Results

The 500 answers (10 answers from each of the 50 examinees) for the three inquiries were obtained. The numbers of each of the answer are shown in Table 1. Positive answers ((1) or (2)) were selected for 309 recommended books in Q1 (61.8%). This result is as good as the result of the existing research [3] . 200 books that the examinees did not know well ((3) or (4) for Q2) were involved in the 309 books. 38 books that the examinees evaluated to be surprised ((3) for Q3) were involved in the 200 books. This means that about 12 % books were serendipitous books. Since the situation such that almost all recommended books are surprising is not adequate, it is considered that the obtained results, most books are proper and the rest are serendipitous, was appropriate recommendation.

Table 1. The results of the recommendation experiment.

	(1)	(2)	(3)	(4)	
Q1	156	153	149	42	500
Q2	65	55	77	303	500
Q3	228	147	125	N/A	500

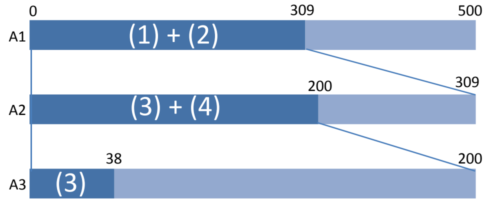

Fig. 4. The ratio of recommended books that were serendipitous.

4 CONCLUSIONS

In this study, the serendipity oriented recommendation method was proposed. Impressions that users felt from a book is extracted as a “concept” using tag data attached to books in the folksonomy style. In the proposed method, the concepts are used for selection of books recommended to users.

The recommender system was implemented and recommendation experiments were carried out using the system. It was confirmed that the recommender system based on the proposed method has enough recommendation accuracy and can recommend serendipituous books to users.

REFERENCES

[1] Kamishima T (2007-2008), Algorithms for Recommender Systems (in Japanese), Journal of JSAI, vol.22,no.6-vol.23,no.2

[2] Philip L. Cahn (1999), A non-invasive learning approach to building web user profiles, KDD.99 Workshop on Web Usage Analysis and User Profiling

[3] Niwa S, Doi T, Honiden S (20066), Web Page Recommender System based on Folksonomy Mining (in Japanese), IPSJ Journal, 47(5), pp.1382-1392