https://doi.org/10.20965/jaciii.2012.p0266

Paper:

A Recommendation System with the Use of Comprehensive Trend Indication Based on Weighted Complete Graph

Takuya Sugimoto*, Tetsuya Toyota*,, and Hajime Nobuhara***

*Department of Intelligent Interaction Technologies, University of Tsukuba
1-1-1 Tenodai, Tsukuba Science City, Ibaraki 305-8573, Japan
E-mail: sugimoto@cmu.iit.tsukuba.ac.jp
**Japan Society for the Promotion of Science
Sumitomo Ichibancho FS Bldg., 8 Ichibancho, Chiyoda-ku, Tokyo 102-8472, Japan
[Received August 10, 2011; accepted October 28, 2011]

Recently, Internet shopping has become widespread, websites of which are equipped with a recommendation system to help users easily find their target items from among vast product information. As a typical method to create recommendation information, collaborative filtering is used but it has a problem that recommendation results tend to be biased toward the same category. Since this study intends recommendation with a high discoverability from a large point of view of category, we define dissimilarity between products based on information on Browse Node ID held by some products in Amazon and use *k*-medoids to newly categorize the products. Moreover, we create a weighted complete graph with those categories as nodes and indicate the trend across different categories. The proposed system estimates and recommends a category strongly related to a category that is thought to be unknown to the user but the user will like based on information of the weighted complete graph. We evaluate the effectiveness of the proposed system through experiments with 9 undergraduate students, 12 graduate students, and 2 office workers as subjects and show that the proposed system is better in recommending unknown products to the user than existing recommendation systems.

Keywords: recommendation system, graph theory, cluster analysis

1. Introduction

Recently, Internet shopping has become widespread because of its advantages in convenience and wide selection of products. In Internet shopping, sellers do not necessarily have brick-and-mortar shops, set up their shops with a little initial funding, and can effectively develop the market because they target many Internet users as clients. Buyers too have great advantages in price reduction due to cuts in distribution middlemen and opportunities for purchase of products available in shops difficult for the user to go to. The fact that the number of users of Amazon, which operates websites in six countries, rose by about 25% in a year from 2008 to 2009 [1] indicates that Internet shopping is becoming a common means of shopping. Such shopping sites are equipped with a recommendation system so that the user can easily find the target product from vast amounts of product information. One of the commonly used methods to create recommendation is collaborative filtering [2]. This enables accurate recommendations to the user by calculating users' similarities based on each user's taste information and using this to estimate predicted rating scores with respect to unknown items to the user. However, recommendation using collaborative filtering requires profiles such as users' taste information, which places a heavy burden on the user. In addition, there is a problem that recommendation results are biased toward the same category.

This study proposes a system that presents recommendation information that allows the user to find a new taste in addition to recommendation information of Amazon. By trend indication using complete graphs with categories as nodes, the proposed system creates recommendation information with an improved discoverability from a comprehensive point of view and presents it to the user. In addition, by indicating trends using complete graphs created in advance, the proposed system intends to build a highly real-time system without requiring the users' individual profiles.

This paper consists of the following sections. Section 2 includes prior studies on collaborative filtering and variation of recommendation and their problems. Section 3 proposes a system to present recommendation information that solves those problems and allows the user to find new tastes in addition to recommendation by existing methods. Section 4 presents a rating experiment to verify the effectiveness of the proposed system. Lastly, Section 5 offers discussions and future issues.

2. Prior Studies on Collaborative Filtering and Variation of Recommendation

Collaborative filtering is a method to estimate predicted rating scores of unknown items for the user and recom-

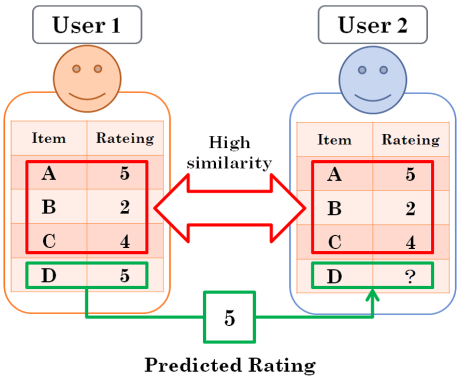

Fig. 1. Example of rating score prediction by collaborative filtering.

Fig. 2. Example of recommendation by Amazon (recommendation based on previously purchased products).

mend them based on similarities between rating score vectors with respect to the products. **Fig. 1** shows an example of rating score prediction by collaborative filtering. Since both the users rate the products A, B, and C at the same rating scores, these two are thought to have a high similarity in tastes. Accordingly, for the product D, which has been rated only by user 1, user 2 is predicted to give a similar rating score. The recommendation system using collaborative filtering rates a vast amount of products with this predicted rating score and recommends products with a high score.

Representative systems using collaborative filtering include GroupLens [3] by Paul Resnick et al. and Ringo [4] by Upendra Shardanand et al. GroupLens was developed to classify Web-based news articles and was the first study that strictly formulated the idea of collaborative filtering as an algorithm. Ringo is a collaborative filtering system for music that is characterized by using a Pearson's product-moment correlation coefficient for calculation of user similarities. In addition, there is a study to apply collaborative filtering to recommendation of unknown useful functions of software [5], which is used in various fields. However, there is a problem that GroupLens and Ringo cannot recommend new articles that nobody has ever read and music that nobody knows. In addition, daily change in users' taste causes their rating score to fluctuate, and thus it is very difficult to accurately comprehend taste information. Intending to improve variation in recommendation, Ziegler et al. proposed topic diversification to recommend items of wider fields even at the expense of prediction accuracy [6]. Shimizu et al. [7] proposed multiple algorithms that use profile information about whether items are known or unknown to the user in addition to the user profile about taste to recommend products unknown to the user. However, use of the user profile has a problem of increase in burden on the user. **Fig. 2** is an example of item recommendation by Amazon. The main recommendation method by Amazon is collaborative filtering using the user's purchase history and the like, which may have the problem described above.

This study focuses on a problem that recommendation information becomes similar when an existing method is used and proposes a system to solve this problem without increasing burden on the user. To achieve that, we obtain product information of Amazon.co.jp, define the categories by clustering them, and indicate relationships between the categories as a complete graph. By preparing the complete graph in advance, a recommendation is created after knowing a comprehensive trend. This allows individual profiles to be unnecessary and burden on the user to be reduced. This is also an effective means for cold start problems. In addition, this allows the user to find a new taste not by creating recommendation information per product but by getting recommendation from among a large-order product group such as per category.

3. Proposed System

3.1. Amazon Web Services

Amazon Web Services [8] are an API provided by Amazon.com, which offers diverse information such as product reviews on receiving URL requests transmitted by the users. This study targets data of domestic products in Japan and uses Amazon Web Services to obtain product information of about 8000 items of Amazon.co.jp.

3.2. Category Definition

Amazon has a category structure with a hierarchical arrangement from high order categories such as "Books" and "DVDs" to low order categories such as "Authors with the name starting with 'A'" and "Japanese Literature," in which different categories are given depending on media.

However, Amazon has many products in their line of business and their related products in different media such as soundtrack CDs of films. If the categories defined by Amazon are used as they are, those products may be added into recommendation information as products of different categories. However, those may be well known to the users due to the relevance in works. So, items that come in as different media but have relevance in works are handled as items in the same category, and recom-

Browse Node	Browse Node ID
Books	465392
Literature & Fiction	548206
Short Stories	467250
...	...

Fig. 3. Browse Node and Browse Node ID.

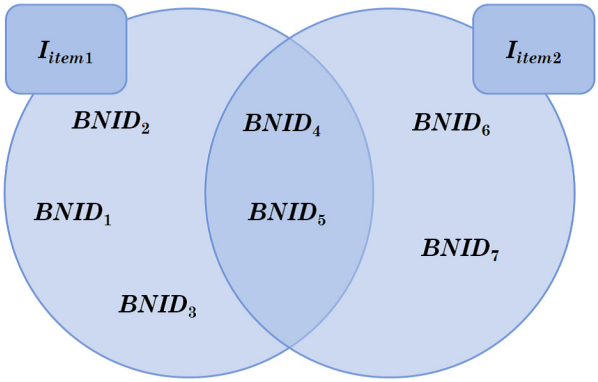

Fig. 4. Example of inclusion relation of products and Browse Node ID.

mendation information is created from items in different categories.

Due to this, the categories are defined again using category names given to items by Amazon and their identification number Browse Node ID (**Fig. 3**) and the same category is given to relevant product groups.

3.3. Category Creation by Clustering

One widely used clustering method to classify multiple data is the k-means method [9]. The k-means method is a method to carry out clustering by minimizing the sum of squares of the distance between the centroid of cluster and each vector data point in the cluster.

The use of the k-means method has an advantage in that the variable k, that is, the final number of categories, can be defined arbitrarily. In the k-means method, the dimensions of all of the target vector data have to be the same. However, the numbers of the Browse Node IDs held by the products are often different from one another, and thus the k-means method cannot be applied. So, this study uses k-medoids [10], to which the k-means method is applied, to carry out clustering, and defines the obtained clusters as categories. The largest difference between the k-means method and the k-medoids lies in the type of data to be used. While the centroid represents each cluster in the k-means method, the medoid represents the cluster in the k-medoids method. The medoid is determined so as to minimize the total sum of dissimilarities between one data point and others in a cluster to which the former belongs. This enables clustering to be carried out by defining the dissimilarities between each point with a certain method even to multiple data with a different dimension p. This study lets a set that has the Browse Node ID held by the product x as an element be I_x and lets the dissimilarity $dissim_{ij}$ with respect to the products i, j be

$$dissim_{ij} = 1 - \frac{|I_i \cap I_j|}{|I_i \cup I_j|} (\in [0,1]). \quad \ldots\ldots \quad (1)$$

The second term of the right-hand side of the expression (1) is called Tanimoto distance [11], which expresses similarity between sets. So, the higher the similarity between i and j is, the smaller value the $dissim_{ij}$ has. **Fig. 4** shows the inclusion relation between the product and the Browse Node ID. I_{item1} is a set that has the Browse Node ID held by the item 1 as an element, to which $BNID_1$ to $BNID_5$ belong. Similarly, $BNID_4$ to $BNID_7$ belong to I_{item2}, where $BNID_4$ and $BNID_5$ belong to both the sets. A calculation example of $dissim_{item1,item2}$ with respect to such sets I_{item1} and I_{item2} is shown below.

$$\begin{aligned} |I_{item1} \cap I_{item2}| &= 2 \\ |I_{item1} \cup I_{item2}| &= 7 \\ dissim_{item1,item2} &= 1 - \frac{2}{7} \\ &= \frac{5}{7} \end{aligned}$$

In addition, with respect to the cluster J_i to which the data point x, y belongs, the medoid is determined by

$$medoid_i = \operatorname*{argmin}_{x \in J_i} \sum_{y \in (J_i - \{x\})} dissim_{x,y}. \quad \ldots \quad (2)$$

That is, the medoid is one of the clustering target data points, and each data point belongs to a cluster that is represented by the medoid with the smallest dissimilarity with it. k clusters set by the k-medoids are defined as the categories $J_1, J_2, \ldots, J_k$. This study relates these k clusters to nodes of a weighted complete graph and indicates the trend of users who give ratings in Amazon.co.jp.

The procedure of the k-medoids is presented below:

1. Select k pieces of data at random and set to the initial *medoid*.
2. Calculate dissimilarity between each piece of data and the *medoid*.
3. Set the cluster of the *medoid* with the smallest dissimilarity as a cluster of each piece of data.
4. Set a new *medoid* so as to minimize the total sum of the distance between the *medoid* and the data point in each cluster.
5. Repeat the procedures 2 to 4 until the clusters do not change.

3.4. Complete Graph

This study uses the properties of complete graphs to indicate trends across different categories. A complete graph is a graph with an edge between each node and all other nodes [12]. The graph may be shaped so that the edge is weighted to more specifically indicate the relationship between nodes, which is called a "weighted complete graph."

For products the user searched, the proposed system presents, in addition to existing recommendation, recommendation information with improved discoverability related to the category that is auxiliary for the existing recommendation. At this time, let the nodes of the graph be the category J created by clustering. The use of the weighted complete graph can indicate the relationship between each category and all other categories, thereby enabling recommendation information to be created regardless of whatever category the user searches for products of.

The procedure to create a weighted complete graph that indicates the relationship between categories:

1. Obtain product information
   a. ASIN (Amazon Standard Identification Number: Individual number given to each product by Amazon).
   b. ID of the user who gives rating score.
   c. Rating score.
   d. Browse Node ID.
2. Categorization based on Browse Node ID
   a. Refer to the Browse Node ID of each product.
   b. Carry out clustering by k-medoids.
3. Create a weighted complete graph based on the number of users who gave high ratings in each category
   a. Set category to node.
   b. Calculate the number of users who gave high ratings in two arbitrary categories in common.
   c. Draw an edge with the calculated number of users as a weight and create a weighted complete graph (**Fig. 5**).

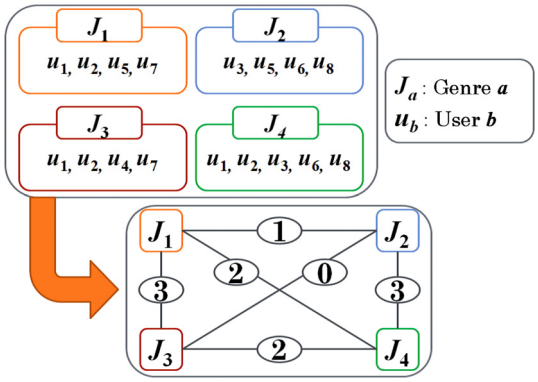

Fig. 5. Example of creation of a complete graph.

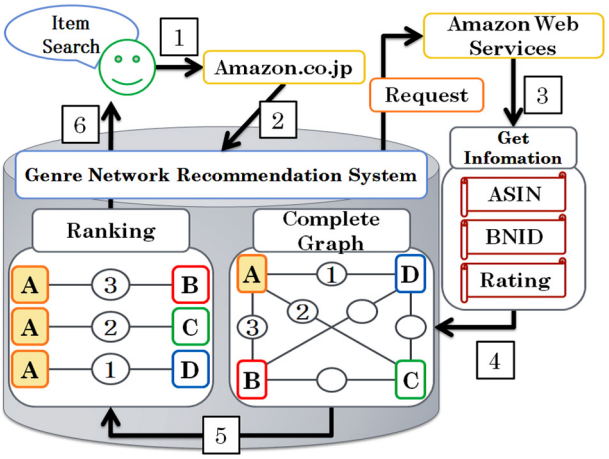

Fig. 6. Outline of recommendation procedure by the proposed system.

3.5. Category Network Recommendation System

A recommendation procedure by the "category network recommendation system," which is a recommendation system using a complete graph that defines the relationship between categories is presented below and its outline is presented in **Fig. 6**.

1. Product search by the user.
2. Obtain product name and ASIN.
3. Obtain Browse Node ID, rating user ID, and rating score.
4. Determine category (corresponding node of complete graph) from obtained information.
5. Create ranking of the categories.
6. Recommend products of highly ranked categories.

For product search by the user, the system first obtains the ASIN and the product name of the searched product. The system next uses the obtained ASIN to create a request to AWS and thus obtains the Browse Node ID, the rating user ID, and the rating score of each user. In addition, with reference to information of about 8000 products prepared in advance, the system calculates dissimilarity between all products and the searched product using the expression (1) similarly to the category definition. Then, the searched product is added to the category to which the product with the closest information of the held Browse Node ID belongs. After the category of the searched product is determined, the weights of the edges of the corresponding node and other nodes in the weighted complete graph are sorted in descending order and listed. A category in which more users give a high rating to a product in the category of the corresponding node is ranked higher in the list. In accordance with the ranking of this list, four products from the top category, three from the second cat-

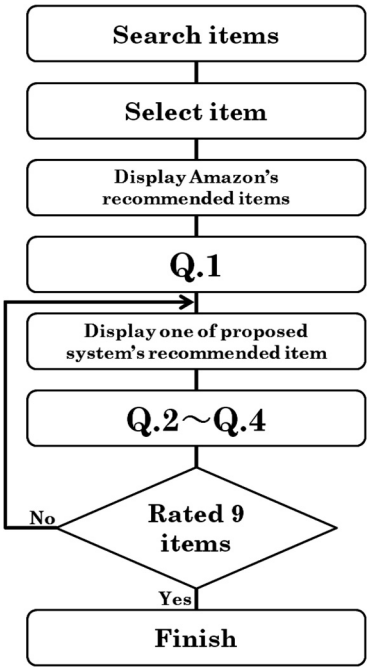

Fig. 7. Experiment procedure.

egory, and two from the third category are selected at random and presented to the user. By recommending more products from the higher categories, the trend indicated using the weighted complete graph is reflected in the recommendation information.

4. Rating Experiment

4.1. Experiment Outline

The proposed system creates recommendation information that does not alone satisfy all requests from the user but is intended to be used as supporting information that is auxiliary for existing recommendation and for the user to find a new taste. So, rating is conducted from the point of view of satisfaction as to whether an unknown product has been found and as to how much the recommended product was interesting. We conduct a rating experiment based on the procedure shown in **Fig. 7** targeting 23 subjects including undergraduate students, graduate students, and office workers. The experiment was conducted using a desktop PC with Intel (R) Core (TM) i5 (2.67 GHz + 2.67 GHz) + 2 GB memory + Windows 7 and a laptop PC with Intel (R) Core (TM) i7 (2.80 GHz + 2.80 GHz) + 4 GB memory + Windows 7.

At first, each subject searches five products of his own taste and recommends nine products for each of the searches. The weighted complete graph used for the experiment is created by the *k*-medoids with high rating users who gave rating scores of 4 or 5 on a five-point scale to products in Amazon and with 100 clusters (the number of categories) and 1500 trials. The number of clusters and the number of trials are determined by carrying out clustering with 50, 100, 500, 1500, and 2000 trials for each

Table 1. Preliminary experiment result.

		The number of trials				
		50	100	500	1500	2000
The number of clusters	5	0.985	0.987	0.986	0.986	0.985
	10	0.984	0.986	0.977	0.978	0.977
	15	0.981	0.976	0.976	0.983	0.975
	20	0.978	0.981	0.973	0.985	0.984
	50	0.971	0.969	0.974	0.978	0.972
	100	0.974	0.970	0.969	0.968	0.971
	200	0.969	0.969	0.971	—	0.971
	500	—	—	—	—	—
	1000	—	—	—	—	—

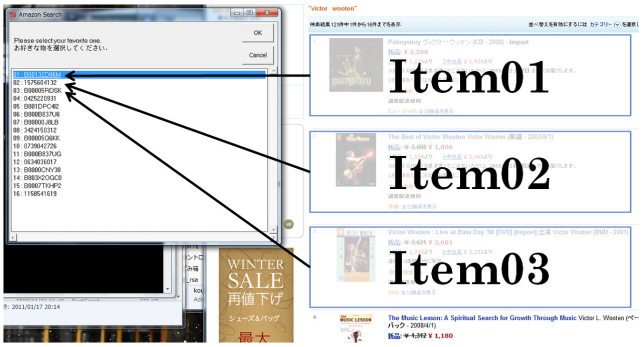

Fig. 8. Example of output of search result.

of 5, 10, 15, 20, 50, 100, 200, 500, and 1000 clusters in a preliminary experiment and selecting the combination with the lowest average of dissimilarity of clusters.

The preliminary experiment result is shown in **Table 1**. It is to be noted that cases in which there is a cluster whose element is only *medoid* are excluded from the target and described as "—." At this time, as a clustering tool, we used Pycluster, which is a Python binding module provided by Cluster 3.0 [13].

Four, three, and two recommended products were selected at random from the categories of the top to the third, respectively, of the ranking created from the weighted complete graph. Questions Q.1 to Q.4 are presented below.

Q.1. How many known products are included in the products recommended by Amazon? (0 to 9 items)

Q.2. Do you know the products recommended by the system? (0: I know, 1: I have heard of, 2: I do not know)

Q.3. Are you interested in the products recommended by the system? (0: Yes, 1: No)

Q.4. Do you want to have the products recommended by the system? (0: I want, 1: I rather want, 2: Neither, 3: I rather do not want, 4: I do not want)

Figure 8 is an example of output to the user search and **Fig. 9** is the input interface of Q.1. We used a similar interface for each of the other questions.

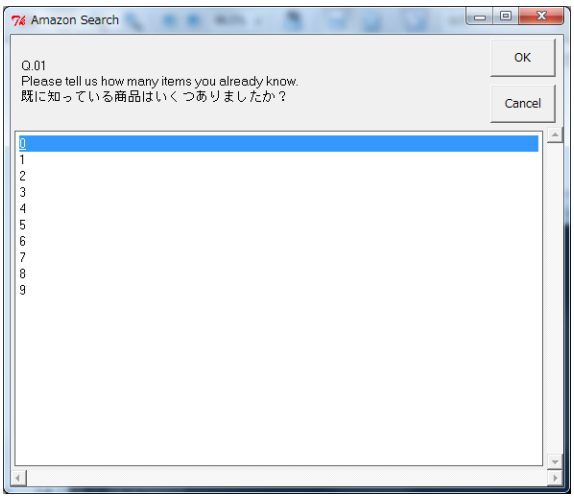

Fig. 9. Input screen for answer to Q.1.

Table 2. Rating experiment result (comparison by table).

Question		Average values
About recommendation by Amazon	Q.1	5.536
About recommendation by the system	Q.2	1.767
	Q.3	0.624
	Q.4	2.774

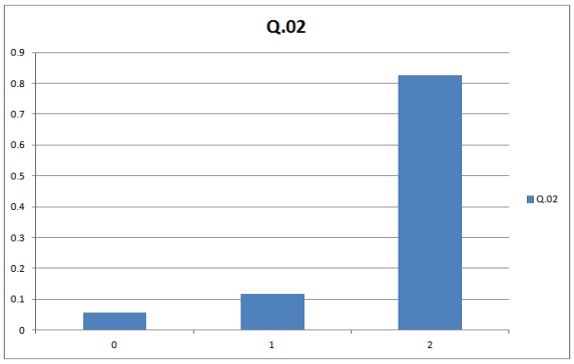

Fig. 10. Answers to Q.2.

4.2. Experiment Result

Table 2 presents averages of answers to each question and **Fig. 10** presents details of answers to Question 2. The result of Q.1 indicates that more than half of the recommended products of Amazon are known to the user. Some of the subjects know more than 70% of the products, which implies that the recommendation information of Amazon is poor in variation. On the other hand, in the answers to Q.2, many products are evaluated as "I do not know," which indicates that the proposed system has successfully recommended products unknown to the users. The score in Q.3 is 0.624, which shows that about 40% of the recommended products interested the users. However, about the half of answers are "I do not want" to Q.4, which shows that a few products got a rating that they actually want the products and clarifies that extraction of the user trend needs to be improved. This is thought to be because the population of the products recommended by the system is biased in terms of category. Since most of the about 8000 products obtained using AWS were books, the recommended products inevitably include many books, so that for those who are not very interested in books most of the recommended products by the proposed system are unnecessary. In addition, the Browse Nodes used for the categorization include inappropriate ones for categorization such as "By Artists" and "Refinements" and these are likely to reduce accuracy of the categorization. In future, the system needs to be improved so as to be responsive to a wider variety of users by removing unnecessary Browse Nodes in addition to consideration of biases in population when creating recommendation.

5. Conclusions

To create recommendation with a high discoverability from a large point of view of category, we indicated trends across different categories using the k-medoids and a weighted complete graph. In addition, we proposed a system that uses the recommendation as auxiliary to existing recommendation systems. The rating experiment result confirms that most recommendations by Amazon are known to the users and the proposed system is effective in recommending products unknown to the users. In addition, we observed many cases of recommending products that were deemed to suit the taste of the user regardless of the selected products. However, we did not obtain sufficient results in trend extraction. In future, in view of bias in population when creating recommendation, we will build a system that is responsive to more users.

References:

[1] Compete, "Compete Releases Top 25 Retail Web Sites for July 2009," 2009. http://www.competeinc.com/news_events/pressReleases/238/

[2] S. Alag, "Collective Intelligence in Action," Manning Publications, 2008.

[3] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "GroupLens: an open architecture for collaborative filtering of netnews," CSCW '94 Proc. of the 1994 ACM Conf. on Computer supported cooperative work, 1994.

[4] U. Shardanand and P. Maes, "Social information filtering: algorithms for automating "word of mouth"," CHI '95 Proc. of the SIGCHI Conf. on Human factors in computing systems, 1995.

[5] T. Akinaga, N. Ohsugi, M. Tsunoda, T. Kakimoto, A. Monden, and K. Matsumoto, "Recommendation of Software Technologies Based on Collaborative Filtering," Asia-Pacific Software Engineering Conf., pp. 209-216, 2005.

[6] C.-N. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen, "Improving recommendation lists through topic diversification," Proc. of the 14th Int. Conf. on World Wide Web, pp. 22-32, 2005.

[7] Y. Hijikata, T. Shimizu, and S. Nishida, "Discovery-oriented collaborative filtering for improving user satisfaction," Proc. of the 14th Int. Conf. on Intelligent user interfaces, pp. 67-76, 2009.

[8] Amazon.com, Inc., "Amazon Web Services." http://aws.amazon.com/jp/

[9] H. Takamura, "Introduction to Machine Learning for Natural Language Processing," Corona Publishing Co., LTD., 2010.

[10] S. Theodoridis and K. Koutroumbas, "Pattern Recognition, Third Edition," Academic Press, 2006.

[11] S. Theodoridis and K. Koutroumbas, "Pattern Recognition, Fourth Edition," Academic Press, 2008.

[12] D. Gries and F. B. Schneider, "A Logical Approach to Discrete Math (Monographs in Computer Science)," Springer, 1993.
[13] Laboratory of DNA Information Analysis Human Genome Center Institute of Medical Science University of Tokyo, "Open source Clustering software." http://bonsai.hgc.jp/ mdehoon/software/cluster/software.htm

Name:
Takuya Sugimoto

Affiliation:
Graduate School of Systems and Information Engineering, University of Tsukuba

Address:
1-1-1 Tenodai, Tsukuba Science City, Ibaraki 305-8573, Japan

Brief Biographical History:
2011- Graduate School of Systems and Information Engineering, University of Tsukuba

Main Works:
• "Diffelent Genres of Trend Indication Based on Complete Graph and Application for Advancement Discoverability of Collaborative Filtering," The 73rd National Convention of IPSJ, Mar. 2011.
• "Representing the Trend of Thought among Different Genres Based on Weighted Complete Graph and Its Application to Recommendation System," FSS2011 Fuzzy System Symposium in Fukui, Sep. 2011.

Membership in Academic Societies:
• Japan Society for Fuzzy Theory and Intelligent Informatics (SOFT)

Name:
Tetsuya Toyota

Affiliation:
Graduate School of Systems and Information Engineering, University of Tsukuba

Address:
1-1-1 Tenodai, Tsukuba Science City, Ibaraki 305-8573, Japan

Brief Biographical History:
2008- Graduate School of Systems and Information Engineering, University of Tsukuba
2011- Research Fellow, Japan Society for the Promotion of Science

Main Works:
• "Hierarchical Structure Analysis and Visualization of Japanese Law Networks Based on Morphological Analysis and Granular Computing," I-EEE GrC, 2009.
• "A Fast Learning Algorithm of Self-Organizing Map for Law Text Visualization," SCIS&ISIS, 2011.

Membership in Academic Societies:
• Japan Society for Fuzzy Theory and Intelligent Informatics (SOFT)
• The Institute of Electronics, Information and Communication Engineers (IEICE)
• Information Processing Society of Japan (IPSJ)

Name:
Hajime Nobuhara

Affiliation:
Department of Intelligent Interaction Technologies, University of Tsukuba

Address:
1-1-1 Tenodai, Tsukuba Science City, Ibaraki 305-8573, Japan

Brief Biographical History:
2002.4-2002.9 Post Doctoral Fellow, University of Alberta, Canada
2002.10- 2006 Assistant Professor, Tokyo Institute of Technology
2006 - Assistant Professor, University of Tsukuba

Main Works:
• "Fast Solving Method of Fuzzy Relational Equation and its Application to Image Compression/Reconstruction," IEEE Trans. on Fuzzy Systems, Vol.8, pp. 325-334, 2000.

Membership in Academic Societies:
• The Institute of Electrical and Electronics Engineers (IEEE)
• Japan Society for Fuzzy Theory and Intelligent Informatics (SOFT)
• The Institute of Electronics, Information and Communication Engineers (IEICE)