User Modeling in Folksonomies: Relational Clustering and Tag Weighting

Takuya Kitazawa*
k.takuti@gmail.com

Masahide Sugiyama
sugiyama@u-aizu.ac.jp

School of Computer Science and Engineering
The University of Aizu, Fukushima, Japan

ABSTRACT

This paper proposes a user-modeling method for folksonomic data. Since data mining of folksonomic data is difficult due to their complexity, significant amounts of preprocessing are usually required. To catch sketchy characteristics of such complex data, our method employs two steps: (1) using the infinite relational model (IRM) to perform relational clustering of a folksonomic data set, and (2) using tag-weighting to extract the characteristics of each user cluster. As an experimental evaluation, we applied our method to real-world data from one of the most popular social bookmarking services in Japan. Our user-modeling method successfully extracted semantically clustered user models, thus demonstrating that relational data analysis has promise for mining folksonomic data. In addition, we developed the user-model-based filtering algorithm (UMF), which evaluates the user models by their resource recommendations. The F-measure was higher than that of random recommendation, and the running time was much shorter than that of collaborative-filtering-based top-n recommendation.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*Data mining*; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Clustering, Information filtering*

General Terms

Algorithms, Experimentation

Keywords

Folksonomy, User Modeling, Recommendation

*Current affiliation is Graduate School of Information Science and Technology, The University of Tokyo, Japan.

WIMS '15, July 13 - 14, 2015, Larnaca, Cyprus

DOI: http://dx.org/10.1145/2797115.2797129

1. INTRODUCTION

Folksonomy, also known as social tagging, is the result of a social networking, and it can be used to share a wide variety of content through the use of user-defined tags. Fig. 1 shows an example of a social network creating a folksonomy. In this figure, each user (1, 2, 3, and 4) demonstrates different relationships with different resources (a, b, c, d, and e), each adds their own choices of tags (V, W, X, Y, and Z) to the resource items.

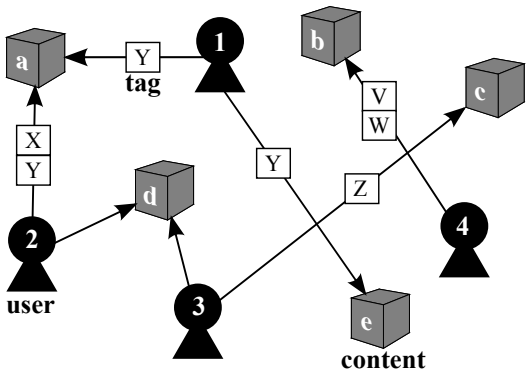

Figure 1: Social network creating a folksonomy

For instance, Flickr[1] and Delicious[2] are well-known services that make use of folksonomy; the former supports photo sharing, and the latter provides social bookmarking. While these services are indeed flexible for users, the analysis of real-world folksonomic data sets poses difficulties. This is due the data being very sparse and tags fluctuating. Tag fluctuations occur because people share the same content with similar, but different, tags; for example, to the same content, one user may add the tag "MUSIC," while another adds the tag "music." Thus, it has been necessary to develop algorithms to normalize the data sets prior to further processing.

The primary reasons for mining folksonomy systems are to find clusters and to develop recommendations. Thus, the principal purpose and expected applications of this study is the same as those of previous studies [2, 9, 11, 12]. However, for the reasons that we mentioned above, the methods presented in previous studies usually require that the data sets undergo careful normalizations and computations. Even though careful data mining improves accuracy in some ways, if folksonomic data can be mined more roughly, there is hope that running time can be reduced, and unpredictable quality of extracted information can be preserved.

More importantly, since these data sets are continuously chang-

[1]Flickr. http://www.flickr.com/
[2]Delicious. http://delicious.com/

ing, time-consuming algorithms are not realistic. For example, the recommender system proposed by Niwa et al. [9] may be impracticable because the system requires many tiny steps. Unfortunately, as long as researchers use the vector space model for folksonomy data sets, it will be necessary to include these extra steps. Even though Hothho et al. [3] established a formal model for retrieving information from folksonomies and contributed greatly to the understanding of folksonomy, their algorithm also uses vectors. Currently, there are many different data mining techniques, and it is straightforward to extract knowledge, such as user preferences and similarities, from folksonomic data, without the need to use a vector space model.

This paper proposes an unsubtle approach to the mining of folksonomic data. In particular, we focus on clustering to determine the relationship between users and resources. Our method is based on the infinite relational model (IRM) [5], which is a nonparametric Bayesian model, and the use of tag weighting to extract user models. User models for services that use folksonomy have a high potential for providing practical personalized applications, such as recommender systems. By demonstrating semantically clustered user models and considering their applications, we show in this paper how relational analysis of folksonomic data can effectively catch the momentary characteristics of users.

2. PROBLEM FORMULATION

2.1 Input and Output

Let U be a set of users, let C be a set of resource items, and let T be a set of tags in a folksonomy. Thus, for example, based on Fig. 1, $U = \{1, 2, 3, 4\}$, $C = \{\mathrm{a}, \mathrm{b}, \mathrm{c}, \mathrm{d}, \mathrm{e}\}$, and $T = \{\mathrm{V}, \mathrm{W}, \mathrm{X}, \mathrm{Y}, \mathrm{Z}\}$. Under this setting, we define a binary relational matrix R on $U \times C$; the i, j-th entry has the value of 1 if the i-th user has a connection to the j-th item, and it is 0 otherwise. Hence, Fig. 1 can be represented as:

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{ccccc} \mathrm{a} & \mathrm{b} & \mathrm{c} & \mathrm{d} & \mathrm{e} \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix} \end{array}, \tag{1}$$

in which the i-th user corresponds to the i-th row, and the j-th item corresponds to the j-th column. Note that when we say a user is connected to a resource item, we mean that the user has bookmarked the item (such as a web page) or has marked it as a "favorites" (such as a shared photo). Each resource item is also associated with added tags and the frequency of each tag, as shown in Table 1.

Table 1: Tag information for each item in Fig. 1

Content	Added tags and frequency
a	{(V, 0), (W, 0), (X, 1), (Y, 2), (Z, 0)}
b	{(V, 1), (W, 1), (X, 0), (Y, 0), (Z, 0)}
c	{(V, 0), (W, 0), (X, 0), (Y, 0), (Z, 1)}
d	{(V, 0), (W, 0), (X, 0), (Y, 0), (Z, 0)}
e	{(V, 0), (W, 0), (X, 0), (Y, 1), (Z, 0)}

The above information comprises the input for user modeling, and an example of the output is shown in Table 2. According to this table, several user clusters are found, and we can infer the characteristics of the each of these clusters from the weighted tags. In the table, $w_k(t)$ indicates the weight of tag t for the k-th user cluster.

In this paper, we will primarily focus on the characteristics and preferences of users, since this information leads to practical personalized applications. In this context, the term "user modeling" refers only to their tagging of content, and not to personal attributes, such as gender or age.

Table 2: Example of output from user modeling, using the data from Fig. 1

k	User cluster	Tag weights
1	{1, 2}	$w_1(\mathrm{V}), w_1(\mathrm{W}), w_1(\mathrm{X}), w_1(\mathrm{Y}), w_1(\mathrm{Z})$
2	{3}	$w_2(\mathrm{V}), w_2(\mathrm{W}), w_2(\mathrm{X}), w_2(\mathrm{Y}), w_2(\mathrm{Z})$
3	{4}	$w_3(\mathrm{V}), w_3(\mathrm{W}), w_3(\mathrm{X}), w_3(\mathrm{Y}), w_3(\mathrm{Z})$

2.2 Infinite Relational Model

First, we will introduce the IRM because it plays a central role in our approach. The IRM is a stochastic model that is used to determine clusters in relational matrices; IRM-based clustering has been used to discover useful structures in real-world relational data, as discussed in previous studies [4, 5]. As a simple example, Fig. 2 demonstrates clustering for a relational matrix in which the entries are 0 (white) or 1 (black) to show the connections between data sets S_1 and S_2. The figure shows three clusters.

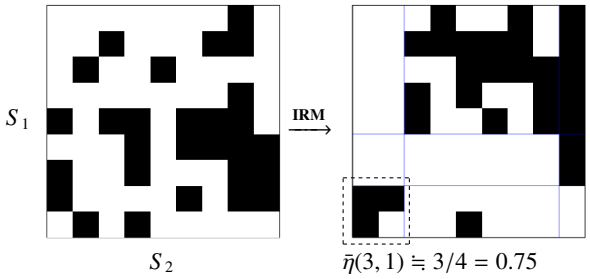

Figure 2: Example of IRM-based clustering

The IRM assumes that the relational matrices, such as the left-hand matrix in Fig. 2, create dense or sparse sub-blocks in it. For a better understanding, cluster assignments for all elements $\mathbf{z}^1$, $\mathbf{z}^2$, and inter-cluster density $\eta(k, \ell)$ are shown in Fig. 3. Importantly, this is just how the IRM assumed the data have been generated. In practice, the relational data R is observed, and the estimation of the optimal cluster allocation $\mathbf{z}^1$ and $\mathbf{z}^2$ is the main task[3].

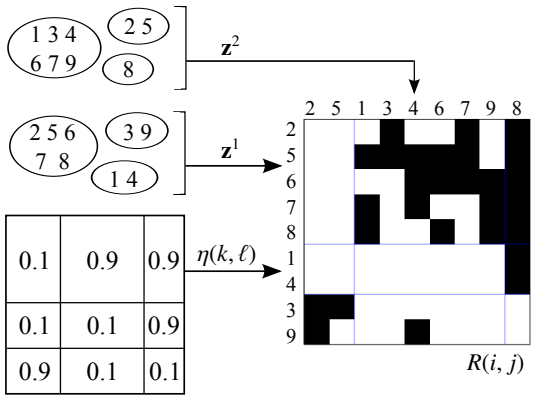

Figure 3: Relational data generative model of the IRM

[3]C code developed by Kemp et al. [5] is available at: http://www.psy.cmu.edu/ ckemp/code/irm.html.

Let N_1 be the number of S_1 clusters, and let N_2 be the number of S_2 clusters. As can be seen in Fig. 2, relational clustering enables the computation of the maximum a posteriori value of $\eta(k, \ell)$:

$$\bar{\eta}(k, \ell) = \frac{m(k, \ell) + \beta}{\bar{m}(k, \ell) + m(k, \ell) + 2\beta}, \quad (2)$$

where $m(k, \ell)$ and $\bar{m}(k, \ell)$ are, respectively, the number of 1s and 0s in the intersection of the k-th cluster ($k = 1, 2, ..., N_1$) with the ℓ-th cluster ($\ell = 1, 2, ..., N_2$), and β is a parameter of the IRM.

2.3 User-Modeling Algorithm

To create user models for a given data set, the following steps are followed: (1) preform relational clustering for the matrix R (a binary relational matrix of $U \times C$), and (2) use tag weighting to extract the characteristics of each user cluster. After the first step, user clusters and content clusters are generated simultaneously.

Next, the characteristics of each content cluster are determined by gathering the tag frequencies, as shown in Table 1. We will introduce a tag weighting technique in Section 4.1, and we will be interested in the tags with the highest weights.

Finally, we integrate the results from the above two steps, and the characteristics of all of the individual users are output as a set of weighted tags. Fig. 4 shows an example of the relationships between components in the user-modeling algorithm. In step (1), the relationship between U_k and C_ℓ is determined by $\bar{\eta}(k, \ell)$, and then in step (2), each content cluster C_ℓ is characterized by the set of tags with the highest weights.

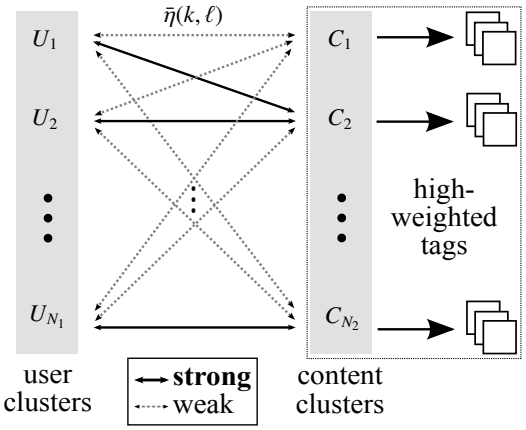

Figure 4: Relationships between components of the user-modeling algorithm

3. RELATIONAL CLUSTERING IN SOCIAL BOOKMARKING

As the first step in user modeling, we collected data sets[4] from Hatena Bookmark[5], one of the most popular social bookmarking services in Japan, and we then performed relational clustering for the data matrix, which consisted of 1,017 users U and 7,000 web pages C [6]. IRM-based clustering generated 163 user clusters ($N_1 = 163$) and 8 content (web page) clusters ($N_2 = 8$). Partial results are shown in Fig. 5, where T_ℓ is the set of tags given to C_ℓ items, with the frequency of each tag.

[4]These data were collected from Jun. 4, 2014 to Jun. 5, 2014.
[5]Hatena Bookmark. http://b.hatena.ne.jp/

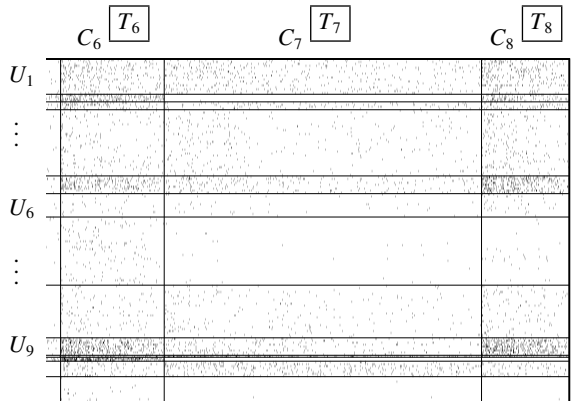

Figure 5: Part of the result of clustering $U \times C$

As can be seen in Fig. 5, the density of 1 in the intersection of U_9 and C_8 is high, as indicated by black dots. From Eq. (2), we find that $\bar{\eta}(9, 8) \fallingdotseq 0.14$ is large, and thus U_9 and C_8 have a relatively strong relationship. We can see that the density in the intersection of U_9 and C_7 is lower, and, as expected, $\bar{\eta}(9, 7) \fallingdotseq 0.02$ is smaller. The user-modeling method uses $\bar{\eta}(k, \ell)$ to find the characteristics of user clusters.

Fig. 6 shows the log-likelihood of a given relation and the number of clusters obtained in each iteration. The cluster assignments of all elements are individually reallocated in each step. Processing of the 1,017-by-7,000 matrix (97.6% sparse) took 54 minutes for 20 iterations. We note that the likelihood and the number of clusters converged at different rates.

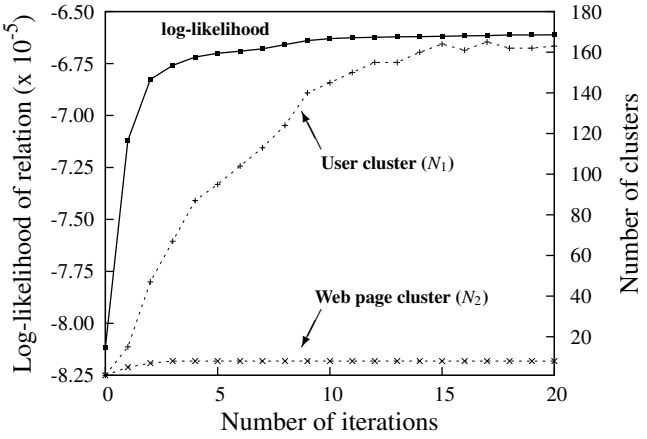

Figure 6: Convergence of the IRM-based clustering

4. USER MODELING BASED ON THE CLUSTERING RESULT

As mentioned in Section 3, $\bar{\eta}(k, \ell)$ indicates the strength of the relationship between the user clusters and the content clusters. In this section, we discuss the assignment of weights to each tag; this is done to determine the characteristics of each of the user clusters. We apply tag weighting and user modeling to the social bookmarking data sets used in the previous section.

4.1 Using Tag Weighting to Extract the Characteristics of User Clusters

The tag-weighting technique is similar to term frequency–inverse

document frequency tf-idf weighting, and our goal is to determine which tags have special significance for each user cluster. Thus, the characteristics of the user clusters correspond to the tags with the highest weights.

Let T_ℓ and tag $t \in T$ correspond, respectively, to a document and a term in tf-idf weighting. Hence, $\mathrm{tf}_{t,\ell}$ indicates the frequency of t in T_ℓ, and df_t denotes the number of T_ℓ such that $\mathrm{tf}_{t,\ell} \neq 0$. Let $\mathrm{idf}_t = \max\{0, \log((N_2 - \mathrm{df}_t)/\mathrm{df}_t)\}$, where N_2 denotes the number of content clusters [8]. Thus, the weight of t in T_ℓ is

$$w_{t,\ell} = \mathrm{tf}_{t,\ell} \cdot \mathrm{idf}_t. \tag{3}$$

In order to compare this with T_ℓ ($\ell = 1, 2, ..., N_2$), we then normalize Eq. (3) by dividing it by $w_0 = \sqrt{\sum_{t \in T_\ell} w_{t,\ell}^2}$:

$$w^*_{t,\ell} = \frac{w_{t,\ell}}{w_0}. \tag{4}$$

Next, our approach integrates Eq. (4) and $\bar{\eta}(k, \ell)$ to compute the weight of the tag for each user cluster. As shown in Eq. (2), $\bar{\eta}(k, \ell)$ is determined for each intersection:

$$\begin{array}{c} \\ U_1 \\ U_2 \\ \vdots \\ U_{N_1} \end{array} \begin{array}{c} \begin{array}{cccc} C_1 & C_2 & \dots & C_{N_2} \end{array} \\ \begin{pmatrix} \bar{\eta}(1,1) & \bar{\eta}(1,2) & \dots & \bar{\eta}(1,N_2) \\ \bar{\eta}(2,1) & \bar{\eta}(2,2) & \dots & \bar{\eta}(2,N_2) \\ \vdots & \vdots & \ddots & \vdots \\ \bar{\eta}(N_1,1) & \bar{\eta}(N_1,2) & \dots & \bar{\eta}(N_1,N_2) \end{pmatrix} \end{array}. \tag{5}$$

Since this paper focuses on the users, the sum of each row in Eq. (5) should be 1, as compared with C_ℓ; however, $\sum_{\ell=1}^{N_2} \bar{\eta}(k, \ell) \neq 1$. So, we normalize $\bar{\eta}(k, \ell)$:

$$\bar{\eta}^*(k, \ell) = \frac{\bar{\eta}(k, \ell)}{\bar{\eta}_0}, \tag{6}$$

where $\bar{\eta}_0 = \sum_{\ell=1}^{N_2} \bar{\eta}(k, \ell)$. The weight of the tag t for each user cluster U_k is

$$w_k(t) = \sum_{\ell=1}^{N_2} \bar{\eta}^*(k, \ell) \cdot w^*_{t,\ell}. \tag{7}$$

The solutions to Eq. (7) provide the U_k user characteristics for large values of $w_k(t)$.

4.2 User Modeling in Social Bookmarking

We applied user modeling to the clustering result discussed in Section 3. First, we used Eq. (4) to compute the tag weights; Fig. 7 shows the 20 tags with the largest weights (top-20) for each T_ℓ ($N_2 = 8$). In this figure, the x-axis and the y-axis indicate the rank of the tags and their weights, respectively, with the top-3 tags written in boxes. For example, note that all of the top-3 tags in T_1 are associated with topical news, and the tags of T_3 are terms related to engineering in the IT field.

As seen above, tag weighting allows us to infer the characteristics of each cluster. However, at the same time, the characteristics of some tag sets, such as T_8, are uncertain. Note that use of the modified idf formula $\mathrm{idf}_t = \max\{0, \log((N_2 - \mathrm{df}_t)/\mathrm{df}_t)\}$ is important in this experiment, since applying the conventional formula $\mathrm{idf}_t = \log(N_2/\mathrm{df}_t)$ leads to undesirable weighting, in which several of the tags end up in the top-3 of different T_ℓ sets.

As mentioned above, in Fig. 7, the height of the gray area in the background represents $\bar{\eta}(9, \ell)$. In this figure, we can see that the tags in T_1, T_6, and T_8 are characteristic of the users in U_9; this is confirmed by the top-20 $w_9(t)$, which are shown in Fig. 8. Most top-20 tags are rather general terms, but there are some heavily weighted technical tags, such as w_9("swift") and w_9("VBA"). Thus, U_9 users are likely to be primarily interested in general topics, but also interested to some degree in programming. This knowledge was obtained as an output of the user modeling. On the other hand, since it is difficult to obtain more-specific user preferences, it is necessary to further refine our proposed method.

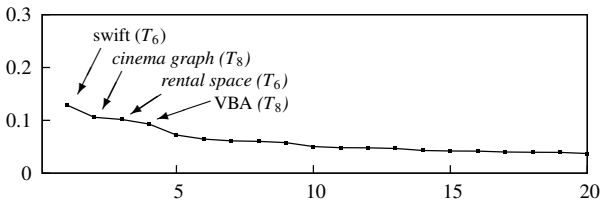

Figure 8: Top-20 $w_9(t)$ tags for U_9 users

5. EVALUATION AND DISCUSSION

In the previous sections, we discussed how to extract from folksonomic data the structure of groups and the characteristics of users. In this section, we will evaluate the correctness of the extracted user models, by using this information to make recommendations. We will use the results of user modeling to learn the data, create recommendations based on the user models, and then evaluate the accuracy of recommendations using test data. Importantly, in practical terms, the number of iterations for IRM-based clustering (IRM iterations) is a crucial factor for running time.

In this section, we will use relational tuples, instead of the relational matrix used above. For example, the relational matrix in Eq. (1) can also be represented as a set of relational tuples:

$$\{(1, \mathrm{a}), (1, \mathrm{e}), (2, \mathrm{a}), (2, \mathrm{d}), (3, \mathrm{c}), (3, \mathrm{d}), (4, \mathrm{b})\}, \tag{8}$$

where the tuple (X,Y) means that user X has a connection to item Y. Note that the size of the set in Eq. (8) is equal to the number of 1s in Eq. (1).

As mentioned in Section 3, the social bookmarking data set has 1,017 users and 7,000 web pages. We represented this data as relational tuples and evaluated them by 5-fold cross validation. Table 3 shows the numbers of tuples, 80% for learning and 20% for testing, in each cross validation step.

Table 3: Number of tuples in the social bookmarking data set

All	100%	172,365
Learning	80%	137,892
Test	20%	34,473

After dividing the data, we reconverted the learning tuples into a relational matrix and then performed user modeling, as shown in the previous sections.

IRM-based clustering is implemented in C, and we ran the program on Vine Linux 6.3 (32bit) with an Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz and 4GB RAM. After relational clustering, the results were handled by codes written in Python 2.7.8.

5.1 User-Model-Based Recommendation

We chose recommended web pages for every user cluster. As shown in Table 2, user cluster k has overall tag weights $w_k(t)$, so our recommendation algorithm can compute the *prediction degree* $P_k(p)$ of web page p for the k-th user cluster:

$$P_k(p) = \sum_{t \text{ added to } p} w_k(t). \tag{9}$$

To give an example, as a result of user modeling in the first cross validation step, the sorted $P_k(p)$ for the largest user cluster k in

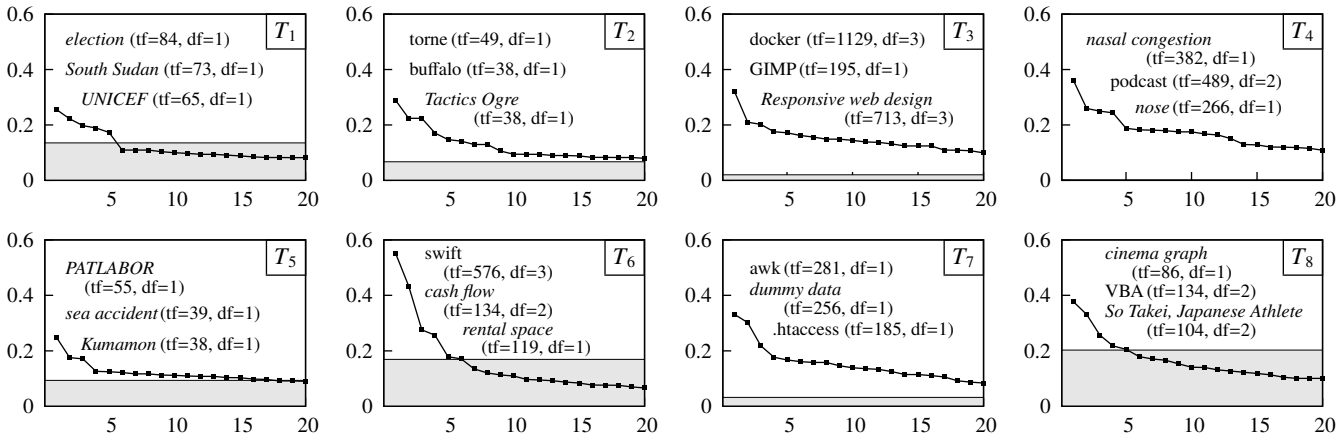

Figure 7: Top-20 $w^*_{t,\ell}$ in T_ℓ and top-3 tags; the height of the gray areas in the background indicates $\bar{\eta}^*(9,\ell)$. Words in *italics* are Japanese tags that were translated into English.

different IRM iterations are shown in Fig. 9. This figure clearly illustrates that only about 400 web pages (out of 7,000 total) are significant.

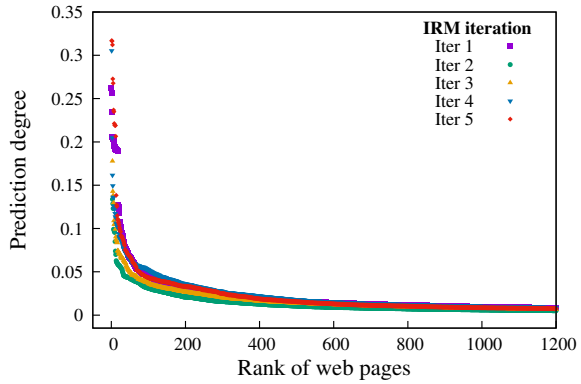

Figure 9: Sorted $P_k(p)$ in different IRM iterations

Fig. 9 also depicts that prediction degrees may not be affected by the number of IRM iterations. Hence, we can use earlier cluster assignments for fast, practical recommendations.

After computing the $P_k(p)$, web pages were recommended when the prediction degree exceeded the threshold θ, that is, $P_k(p) > \theta$ for web page p; then p was recommended to each user $u \in U_k$, excluding any p for which (u, p) was in the learning data. We call this recommendation the user-model-based filtering algorithm (UMF). The precise procedure of UMF is described as Algorithm 1.

Algorithm 1 Find all recommended pages, using a threshold θ

```
L ← ∅                                          ▷ list of recommended relational tuples
for k = 1, 2, ..., N_1 do
    for all {p | p ∈ C} do                     ▷ all web pages
        P_k(p) ← 0
        for all tag t added to p do
            P_k(p) ← P_k(p) + w_k(t)
        if P_k(p) > θ then                     ▷ filtering web page
            for all {u | u ∈ U_k and (u, p) ∉ Learning} do
                L ← L ∪ {(u, p)}
return L
```

5.2 Evaluation Based on the F-measure and Running Time

In this section, we will use the F-measure to evaluate the accuracy of the recommendations, and then we will discuss the effectiveness of the proposed user-modeling method. As a result of the UMF procedure, both the precision and the recall can be computed, as follows:

$$\text{Precision} = \frac{|L \cap \text{Test}|}{|L|}, \tag{10}$$

$$\text{Recall} = \frac{|L \cap \text{Test}|}{|\text{Test}|}. \tag{11}$$

We applied UMF with various values of θ in order to determine the optimal threshold value. As can be seen in Fig. 10, the F-measure peaked when $\theta = 0.025$ for every IRM iteration; we also note that for this value, the overall running time except for IRM-based clustering was acceptable. When $\theta < 0.025$, too many web pages were recommended, the running time was much longer, and the F-measure was lower. Thus, 0.025 is the optimal threshold value. Note that this is the result of 5-fold cross validation.

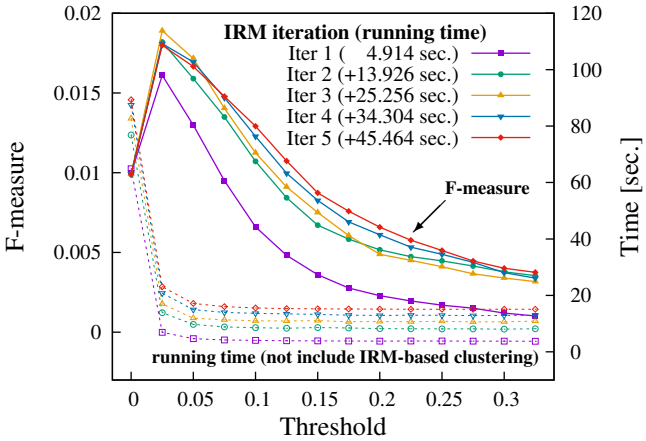

Figure 10: F-measure and running time (except for IRM-based clustering), for different thresholds and iterations

Actually, in Fig. 10, the F-measure increased from the first IRM iteration to the second IRM iteration, but it is clear that the number of iterations did not affect the accuracy after the second iteration. This fact can be also confirmed by Fig. 6 because, at the second iter-

ation, the IRM-based clustering already reached the nearly-optimal likelihood. That is, more iterations do not lead to better F-measures in our method. Hence, from here, we will only focus on the cluster assignments after the second IRM iteration (green line in Fig. 10) for efficient recommendation. To obtain the assignments, relational clustering takes 18.84 [s] before tag weighting (4.914 [s] in the first iteration and 13.926 [s] in the second iteration).

UMF for the second IRM iteration with $\theta = 0.025$ generated 333,951 recommended relations (tuples) on average of 5-fold cross validation. For comparison, we also considered 5-fold cross validation with 333,951 random recommendations as the worst cases. For a single user, we also examined the results of user-based collaborative-filtering (CF) and top-n recommendation, a straightforward and commonly used recommendation algorithm. With CF, we chose the Jaccard similarity coefficient to compute the similarities between the selected user and the other users. Table 4 shows the F-measures and running times for these three different methods.

Table 4: F-measure and running time obtained by the 5-fold cross validation with the three different methods

Method	F-measure	Time [s]
Random (average of five)	0.00891	—
CF and top-328 **for a single user**	0.10273	367.278
UMF ($\theta = 0.025$)	**0.01820**	**29.933** †

† This time is sum of 18.84 [s] for the IRM-based clustering and 11.093 [s] for UMF shown in Algorithm 1.

Since CF compares the similarities separately, the F-measure is not very low. However, despite the calculation of recommendations for only a single user, the running time was more than ten times longer than that of our proposed method. To make recommendations for 1,017 users like UMF, CF will compute scores for over one hour. The main reason for this is that we roughly accomplished user modeling, and the user characteristics were approximated. Naturally, these running times strongly depend on the particular implementation, but such a wide gap will persist. Note that N = 328 in the top-n recommendation comes from $333951/1017 \fallingdotseq 328$, because we considered a single user, and there are 1,017 users in total.

5.3 Discussion

As can be seen in Table 4, UMF was more accurate than the worst cases, and the running time was acceptable. Therefore, our method successfully extracted practical user models. The collaborative filtering-based top-n recommendation was even more accurate, but that approach is time consuming. CF encountered many different problems, such as a cold start and sparsity [1].

It should be noted that Algorithm 1 requires users to choose a threshold value, but an optimal θ may depend on data sets. Even if $\theta = 0.025$ yields poor results in different data sets, searching an optimal threshold like Fig. 10 every time is certainly time consuming. In this case, top-n approach will be more promising because the prediction degree shows characteristic distribution as shown in Fig. 9. Folksonomic data generally suffers from a lot of garbage tags, one-time only tags. So, such specific distribution of $P_k(p)$ is probably general, and it makes easier to implement top-n recommender systems.

Currently, numerous recommendation algorithms have been proposed, but none is sufficiently versatile. For instance, even though matrix factorization [7] is considered a promising solution for the problem of dimensionality in CF, this technique does not work as expected on a very large, sparse, and binary matrix, such as the $U \times C$ used in this study; however, a new branched method has been proposed [13]. As this branching process is repeated, the outcomes will be increasingly unrealistic.

Hence, it is important to gain more sketchy knowledge. From this perspective, the motivation behind many recent studies on large social networks makes sense. Similarly, this study also satisfies this demand by providing relational-cluster-based user modeling. We note that user models can be used not only to provide recommendations, but also for other purposes, such as marketing.

6. CONCLUSION

In this paper, we have proposed a user-modeling method for folksonomy data sets. The method uses the results of IRM-based relational clustering. The experimental results showed that our method discovered semantically clustered tag groups, and thus user modeling was possible. Therefore, this paper has revealed that relational clustering works successfully for folksonomy data sets. Moreover, we made recommendations based on user models, and these performed well in terms of running time. In the future, further experimental supports are required because the Jaccard similarities in CF can be approximated by more efficient techniques like MinHash.

Our method does not use a vector space model and does not require preprocessing, and it is a relatively rough method for mining folksonomic data; however, the results have also revealed some limitations. As a next step, we intend to consider larger data sets and time series of $U \times C$. Furthermore, we intend to incorporate related work on relational data analysis, such as the work by Rendle and Schmidt [10], and to apply an advanced IRM [4] in order to obtain cleaner user models and more accurate recommendations.

7. REFERENCES

[1] J. Bobadilla et al. Recommender systems survey. *Knowledge-Based Systems*, 46:109–132, July 2013.

[2] J. Gemmell et al. Personalization in folksonomies based on tag clustering. In *Proc. of the 6th Workshop on Intelligent Techniques for Web Personalization and Recommender Systems*, pages 37–48, Apr. 2008.

[3] A. Hotho et al. *Information Retrieval in Folksonomies: Search and Ranking*, pages 411–426. Springer Berlin Heidelberg, 2006.

[4] K. Ishiguro et al. Subset infinite relational models. In *Proc. of AISTATS2012*, pages 547–555, Apr. 2012.

[5] C. Kemp et al. Learning systems of concepts with an infinite relational model. In *Proc. of AAAI2006*, pages 381–388, July 2006.

[6] T. Kitazawa and M. Sugiyama. Relational clustering in social bookmark. In *Proc. of ECEI2014*, 2A05, Aug. 2014.

[7] Y. Koren et al. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, Aug. 2009.

[8] C. D. Manning et al. *Introduction to Information Retrieval*. Cambridge Univ. Press, 2008.

[9] S. Niwa et al. Web page recommender system based on folksonomy mining. In *Proc. of ITNG2006*, pages 388–393, Apr. 2006.

[10] S. Rendle and L. Schmidt-Thieme. Pairwise interaction tensor factorization for personalized tag recommendation. In *Proc. of WSDM2010*, pages 81–90, Feb. 2010.

[11] A. Shepitsen et al. Personalized recommendation in social tagging systems using hierarchical clustering. In *Proc. of RecSys2008*, pages 259–266, Oct. 2008.

[12] Z. Xu et al. Towards the semantic web: Collaborative tag suggestions. In *Proc. of the Collaborative Web Tagging Workshop at WWW2006*, May 2006.

[13] Z. Zhang et al. Binary matrix factorization with applications. In *Proc. of ICDM2007*, pages 391–400, Oct. 2007.