Adaptive and Multiple Interest-aware User Profiles for Personalized Search in Folksonomy: A Simple but Effective Graph-based Profiling Model

Keejun Han, Juneyoung Park, Mun. Y. Yi
Department of Knowledge Service Engineering
Korea Advanced Institute of Science and Technology
Daejeon, Republic of Korea
Email: {keejun.han, j.park89, munyi}@kaist.ac.kr

***Abstract*—The data derived from the social tagging system, known as folksonomy, is a potentially useful source for understanding users' intentions. This study seeks to uncover some of the unexplored areas of folksonomy and examine the plausibility of new ideas for the improvement of personalized search. In particular, we challenge several state-of-the-art algorithms by exploiting folksonomy network structures used in creating user profiles that are adaptive and aware of multiple interests of a user, for the personalization of search results. The results obtained from the proposed approach shows a unanimous increase in the performance of personalization when compared to other state-of-the-art algorithms.**

Keywords—folksonomy; personalized search; user profiles; resource profiles; collaborative systems

I. Introduction

Incorporating social tag data, more commonly known as folksonomy [7], into user profile-building has been proven to be an effective way to acquire a better result in personalized search [8]. The fact that users annotate the resources themselves, not only provides a direct connection between the resource and the user, but also creates a context that reveals user's intention with regard to the resource. This nature of folksonomy data, the accuracy and diverse perspectives in understanding user intentions, represents an immense potential for creating user profiles based upon the folksonomy data that can result in a significant improvement of personalized search. Prior studies that have challenged to create user profiles for better personalization results by using folksonomy data mainly focused on how to adapt existing traditional IR techniques, such as Vector Space Model (VSM) [18] and bm25 [15].

Despite the apparent advantage of using folksonomy in personalization, little research has been conducted on incorporating user's actual behaviors with regard to time and multiple interests into personalized search - issues that have the potential to markedly improve personalized search performance. First, it is natural for a user's interest to change over time [10]. For instance, if a user who used to be interested in HCI is now interested in Information Retrieval, the user profile should be adaptive in order to stay relevant to the current interest that is Information Retrieval, rather than the previous interest, HCI. Second, in real life, a user's interest cannot simply be confined into a single subject [1]. Basic human characteristics would in fact suggest that it is more plausible to assume that users are interested in multiple subjects. For example, a user who is interested in Information Retrieval is also likely to be interested in Data Mining and Recommendations, as the techniques in those fields are partially inclusive to Information Retrieval. Lastly, the noted user's behaviors stated above are not mutually exclusive but occur simultaneously, implying that folksonomy-based personalization techniques should be able to solve both issues at once by dynamically adapting to drifts in user's multiple interests. Prior work has considered the user's multiple interests [1] and the drifts in user's interests [10]. However, to the best of our knowledge, none of the prior studies have attempted to solve both issues at once, neither did they utilize it for personalization.

In order to tackle these issues, we present graph-based profiling techniques that can be aware of adaptive and multiple interests of users. Despite the promising performance of VSM-based personalization techniques dealt with in previous studies, those techniques are still considered an oversimplification of the problem, bearing in mind the significant amount of semantics in a resource and a user that can be lost when only a set of paired keywords and their weights are used [9]. Likewise, these semantics can be reserved and utilized by adopting a graph-based approach [5], which can also be simply yet effectively further developed in order to find out the drifting of user's multiple interests. The results obtained from this study are in clear support of the suggested algorithm, with a significant increase in the performance of personalization compared to other state-of-the-art algorithms [4, 11, 15, 18], without exception.

The remainder of this paper is structured as follows: In section 2, we review the state-of-the-art algorithms used in folksonomy-based personalization. In section 3, our graph-based profiling method is proposed. In section 4, we demonstrate the effectiveness of the algorithm through an experiment and present the results. In section 5, we conclude the study and suggest future research directions.

Authorized licensed use limited to: ETRI. Downloaded on March 18,2021 at 06:19:13 UTC from IEEE Xplore. Restrictions apply.

II. Related Work

In this section, we discuss some of the state-of-the-art profiling algorithms that are available and examine their limitations, whilst also introducing previous studies in order to fully explore the network structure in folksonomy and put into perspective our proposed graph-based approach of learning user and resource profiles.

The available state-of-the-art algorithms are usually based on methods such as Vector Space Model (VSM) or bm25 scheme. Xu et al. [18] explored the terms tag frequency (tf), and inversed user frequency (iuf) in their study whereby tf-iuf is an adaptation of the tf-idf weighting scheme. Similarly, Vallet et al. [15] proposed the hybrid of tf-iuf and bm25 weighting scheme. The work presented by Noll et al. [11] is a simple but effective personalization approach to exploit tags for users and resources based upon term frequency. More specifically, they only used the user tag frequency values (tf_u) and set all the resource tag frequencies to 1, in order to empower the user profile when computing similarity. Similar to their work, another approach proposed by Cai et al. [4] suggested Normalized Term Frequency (ntf) weighting scheme that not only emphasizes the importance of tag frequency, but also minimized the bias on active users by adopting the proportion of user tags. The rationale for their approach was that if a user uses a particular tag more frequently, it implies the user is more interested in that tag. Similarly, Xie et al. [17] further developed the ntf scheme to reflect multiple views towards resources in accordance with the user's different preferences through a combination of LDA method while the adaptivity of user interests are overlooked. In this paper, instead of following traditional models, our model utilizes various different techniques to consider the network structure in folksonomy, as the graph-based approach enables the utilization of latent semantics of a tag. We shall evaluate our graph-based algorithm against other state-of-the art profiling techniques presented by the authors aforementioned.

There have been several attempts to explore network structures in folksonomy. Au- Yeung et al. [1] suggested that folksonomy data can be converted into a network structure to produce a better picture of the semantics among tags. Bao et al. [2] presented two algorithms: SocialSimRank and SocialPageRank, in order to exploit the latent semantics of tags and improve the performance of web search systems. On the other hand, Hotho et al. [9] proposed the FolkRank algorithm to acquire the popularity of documents, which also seemed to be better than the original PageRank algorithm, as it exploits user generated tags rather than implicit web links. Similar to the studies above, we too exploit the folksonomy structure, but focus primarily on providing personalized search by considering the adaptivity and multiple varieties of user interests simultaneously, rather than merely improving the overall rank of documents.

III. Proposed Method

In the subsequent subsections, we first present the overall procedure for our graph-based profiling model, called Adaptive and Multiple Interest-aware FolkRank (AMI-Frank). We then follow on to present a method of providing a final personalized search ranking score for a resource, by measuring the relevance between the profiles of a user and the resource.

A. AMI-Frank: Adaptive and Multiple Interest-aware User Profiling Model

A folksonomy F is a tuple $\mathrm{F} = \{\mathrm{U}, \mathrm{T}, \mathrm{R}, \mathrm{A}\}$, where U is a set of users, T is a set of tags, R is a set of resources, and A is a set of annotations, $\mathrm{A} \subseteq \mathrm{U} \times \mathrm{T} \times \mathrm{R}$. Furthermore, a subset of the folksonomy F for a single user U can be defined as Personomy $\mathbb{P}_\mathrm{u}$, whereby $\mathrm{A_u} = \{(\mathrm{t},\mathrm{r}) | (\mathrm{u},\mathrm{t},\mathrm{r}) \in \mathrm{A}$,$\mathrm{T_u} = \{\mathrm{t} | (\mathrm{t},\mathrm{r}) \in \mathrm{A}\}$, and $\mathrm{R_u} = \{\mathrm{r} | (\mathrm{t},\mathrm{r}) \in \mathrm{A}\}$ of a user m. We then can convert $\mathbb{P}_\mathrm{u}$ into an undirected weighted graph $\mathbb{G}_\mathbb{P} = (\mathrm{V},\mathrm{E})$, where $\mathrm{V} = \{\mathrm{v_1}, \mathrm{v_2}, \cdots, \mathrm{v_n}\}$ is the set of vertices which corresponds to the tags, and $\mathrm{E} = \{\mathrm{e_{v_1,v_2}}, \mathrm{e_{v_1,v_3}}, \cdots, \mathrm{e_{v_{n-1},v_n}}\}$ which corresponds to the set of edges. For every edge $\mathrm{e_{x,y}} \in \mathrm{E}$, the weight $\mathrm{w_{e_{x,y}}}$ represents the number of times that tag x and tag y have been used by the user u.

Our algorithm, AMI-Frank, updates the graph $\mathbb{G}_\mathbb{P}$ into $\mathbb{G}^*{}_\mathbb{P}$ by using community clustering algorithm and evaporation technique based on ant algorithm [6], to address the issues of multiple user interest and drifts in the user interest. The algorithm proceeds in the following three steps: Profile Clustering, Graph Update, and Execution of FolkRank.

1) Profling Clustering:

The first step is to find out which tags belong to which interests by applying the community clustering method on the graph $\mathbb{G}_\mathbb{P}$. The advantage of using community clustering method to discover multiple interests is that it is an unsupervised clustering technique, so the effects of parameters are minimized compared to other methods such as topic modeling or k-means clustering [17]. We adopted the modularity-based clustering algorithm [3] that detects highest modularity partitions of networks, which in comparison to other community clustering algorithms shows reliable performance in discovering communities.

The modularity-based algorithm is an iterative clustering method to maximize the modularity values for potential communities. By putting a node into every group, the algorithm iteratively calculates the value of gain for modularity. If the value is positive, the node is placed into the group where the value is the largest; otherwise, the node remains in the current group.

After clustering the items, each community discovered corresponds to a user's individual interests, indicating that for a certain tag $t_x \in T_u$, $I(t_x)$ is an integer value which corresponds to the interest ID which the tag t_x belongs to. This attribute is used to enable the following step to deal with the issue of user's multiple interest in social tagging systems.

2) Graph update:

We now then update the graph $\mathbb{G}_\mathbb{P}$ to $\mathbb{G}^*{}_\mathbb{P}$ by using the extended co-occurrence approach with the evaporation technique. This technique is to create a graph that considers the time of the tag in the user profile, by implementing an evaporation function- each time a new resource is added, the weight of each edge in the graph decreases slightly by removing a small percentage of its current value. Consider a user u annotating a new resource r tagged with tags $t_1, \cdots, t_n$,

Authorized licensed use limited to: ETRI. Downloaded on March 18,2021 at 06:19:13 UTC from IEEE Xplore. Restrictions apply.

the updating process occurs in the following steps. First, the current weights of edges on the current graph are changed by the evaporation formula below. Note that $\rho \in [0,1]$ is a constant and $w_{e_{x,y}}$ is the weight of edge connects between tag t_x and t_y. In this paper, ρ was set to 0.2 as it was the best performing value for our experiment.

$$Evaporation(w_{e_{x,y}}) = w_{e_{x,y}} - \rho \cdot w_{e_{x,y}} \quad (1)$$

Second, after the evaporation, the n tags from the resource r : $t_1, \cdots, t_n$ are added to the graph. For every combination of $\mathrm{t_x}$ and $\mathrm{t_y}$ where $x, y \in 1, \cdots, n$ and $\mathrm{x} < y$, the current edges are updated as follows:

$$w_{e_{x,y}} = \begin{cases} 1\,, & \text{if } w_{e_{x,y}} = 0 \text{ and } \mathrm{I(t_x)} \neq \mathrm{I(t_y)} \\ w_{e_{x,y}} + 1\,, & \text{if } w_{e_{x,y}} > 0 \text{ and } \mathrm{I(t_x)} \neq \mathrm{I(t_y)} \\ \left(w_{e_{x,y}} + 1\right) + \gamma, & \text{if } w_{e_{x,y}} > 0 \text{ and } \mathrm{I(t_x)} = \mathrm{I(t_y)} \end{cases} \quad (2)$$

Note that γ was simply set to 1 in order to minimize the noise caused by the idiosyncratic nature of datasets; and $I(t_x)$ and $I(t_y)$ are the integer values corresponding to user's interests that tag t_x and tag t_y belong to, which was obtained in the previous step.

Fig. 1 shows two partial graphs $\mathbb{G}_{\mathbb{P}}$ and $\mathbb{G}^*{}_{\mathbb{P}}$ of real user u, which was chosen from our dataset and consists of two main interests: Social networking and Information retrieval. Note that the final structure of the graph $\mathbb{G}^*{}_{\mathbb{P}}$ looks exactly identical to the graph $\mathbb{G}_{\mathbb{P}}$, as our algorithm only makes changes on the weight of edges within the graph. The edge weight shown in graph $\mathbb{G}_{\mathbb{P}}$ are either 1 or 2 and the simplicity of these values potentially imply that it may not accurately reflect the relatedness between tags for a user. Meanwhile, based upon the co-occurrence relations, the edge weights in the graph $\mathbb{G}^*{}_{\mathbb{P}}$ becomes more varied according to the bookmarking time. For instance, although the tags social, networking, and personalization are initially used together, indicating that the weights of edges among the tags are identical in phase (a), the final weights shown in phase (c) are differentiated. Additionally, the edge weights of each pair: social, personalization, and recommender are all 2 in graph $\mathbb{G}_{\mathbb{P}}$. The values start to become distinguishable in phase (c), as each edge has a different property in terms of adaptivity and multiplicity. This variance in edge weights results in the variance of the importance of tags, defined as v, and is applied in the graph-based profiling technique in the next step.

3) Execution of FolkRank:

FolkRank, originally suggested in [9], is the most powerful existing graph-based profiling model designed for folksonomy-based network. However, we had to make a few subtle adaptations to the original FolkRank algorithm for this paper to better fit the attribute of personalization, as the original algorithm mainly focuses on how to utilize folksonomy in improving general search rather than personalized search.

The adapted algorithm uses a PageRank to create profiles for personalization. The assumption of PageRank is that a Web page is important if it is referenced by an important page [12]. Likewise, in a folksonomy network, a tag is considered to be important if a tag is used together with an important tag. Note again that the formula to implement FolkRank in this paper differs from the original formula suggested in [9], as $\mathbb{G}_{\mathbb{P}}$ is not a tripartite graph that requires the aggregation of three different types of nodes for each iteration. FolkRank algorithm for user u and resource r profiles can be defined as follows, respectively:

$$v_{\mathrm{u}}(t_x) = (1-\mathrm{d}) + d * \sum_{t_y \in In(t_x)} \frac{w_{e_{x,y}}}{\sum_{t_k \in Out(t_y)} w_{e_{k,y}}} v_{\mathrm{u}}(t_y) \quad (3)$$

$$v_{\mathrm{r}}(t_x) = (1-\mathrm{d}) + d * \sum_{t_y \in In(t_x)} \frac{w_{e_{x,y}}}{\sum_{t_k \in Out(t_y)} w_{e_{k,y}}} v_{\mathrm{r}}(t_y) \quad (4)$$

Where d is a damping factor that is usually set to 0.85, it has the role of integrating into the model the probability of jumping from a given vertex to another random vertex in the graph. For folksonomy, this damping factor is used as a random suffer model, where a user annotates about a random topic of a given resource with a probability d, and jumps to a completely new topic with probability $1-d$. Starting from arbitrary values assigned to each node in the graph $\mathbb{G}_{\mathbb{P}}$, the computation iterates until convergence below a given threshold is achieved. We set the threshold value to 0.001, by using the Gephi application[1] that has 0.001 as the default value. After the halt, the score for each node represents the importance of the node in the graph.

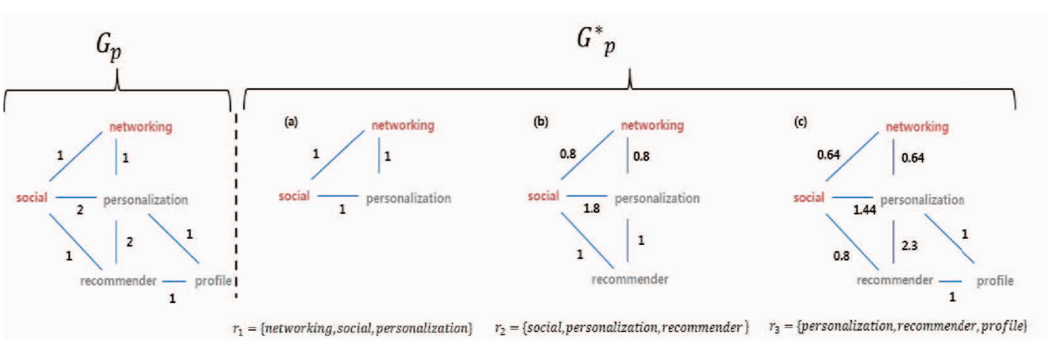

Fig. 1. Example of graph update process for real user u

[1] http://gephi.org

Authorized licensed use limited to: ETRI. Downloaded on March 18,2021 at 06:19:13 UTC from IEEE Xplore. Restrictions apply.

Continuing from the previous example, Table 1 shows the final values of the tags in descending order, after applying FolkRank on the graph $\mathbb{G}^*_{\mathbb{P}}$ in Fig. 1. To clearly observe how the final values change when adaptivity and multiplicity features are added at a time, we define FolkRank as the basic tag-based network propagation algorithm; Adaptive FolkRank (AFrank) as the FolkRank that only considers the adaptivity; and lastly our algorithm, AMI-Frank, that considers both variables simultaneously.

With the consideration of adaptivity, the values of the two tags social and recommender becomes differentiated, as recommender was annotated more recently. Note that newly annotated tags weighs more in AFrank. Furthermore, the values change once more when the multiplicity factor is incorporated. For instance, the values for personalization and recommender increased by strengthening the relatedness between the tags, indicating that the user profile is now adapted to be more sensitive for the tags representing current interest rather than other tags. As seen from Table 1, different schemes produce different weighting score, thus providing different search results.

Table 1. Ranking schemes for user u

FolkRank		AFrank		AMI-Frank	
tag	value	tag	value	tag	value
personalize	0.319	personalize	0.319	personalize	0.328
social	0.218	recommend	0.237	recommend	0.250
recommend	0.218	social	0.198	social	0.188
networking	0.121	profile	0.141	profile	0.133
profile	0.121	networking	0.103	networking	0.098

B. User Models and Resource Relevance Measurement

Once the user and resource profiles are created, their relevance is then measured to rank the resources for individual users. There are several previous works that suggested methods in measuring the relevance between user interests and resources. In this paper, we chose the following relevance measurements to investigate whether our proposed profiling technique is robust regardless of the relevance measurement, as they are most common and effective.

In the widely used Vector Space Model (VSM), all the queries and resources are mapped to be vectors in a universal term space. The similarity between a query and resource is then calculated by using cosine similarity. Inspired by the VSM model, Xu et al., [18] measured user interests and resources as follows:

$$CosineSimilarity(u,r) = \frac{\sum_l (v_u(t_l)\ \cdot\ v_r(t_l))}{\sqrt{\sum_l \big(v_u(t_l)\big)^2} \cdot \sqrt{\sum_l \big(v_r(t_l)\big)^2}} \tag{5}$$

On the other hand, Vallet et al. [15], proposed another similarity measure called scalar similarity, which is an adaptation of the cosine similarity. In conventional measurements, the popularity of resources had been overlooked, since the length normalization factor was not considered. However, due to popularity factor being a good source of relevancy, the length normalization factor was removed from the conventional similarity model as follows:

$$ScalarSimiarlity(u,r) = \sum\nolimits_l (v_u(t_l) \cdot v_{\mathrm{r}}(t_l)\,) \tag{6}$$

The last similarity measure proposed in [15], between the user and resource is to assume that the user profile takes part as a query indicating the user's interests. We denote the measurement as user similarity, which can be defined as follows:

$$UserSimilarity(u,r) = \sum\nolimits_{(l|v_{\mathrm{r}}(t_l)>0)} v_u(t_l) \tag{7}$$

The final goal of personalized search is to provide resources that match both the query requirements and user's interests, indicating that the ranking aggregation process is required to obtain the final resource ranking for a particular user and query. The final relevance score between a resource r and a query q issued by user u is as follows:

$$FRank(u,r,q) = \mu \cdot r_{user-resource}(u,r) + (1-\mu) \cdot r_{query-resource}(q,r) \tag{8}$$

From the equation, $\mathrm{r_{user-resource}}$ is the rank of the resource r in the ranked list generated by user interest relevance, and $\mathrm{r_{query-resource}}(q,r)$ is the rank of the resource r in the ranked list generated by query relevance matching, and μ is the parameter value that manipulates the effect of each relevancy. In this paper, we simply set it as 0.5 to equally balance the two relevance scores.

IV. Experiments

In this section, we present an experimental evidence for the proposed graph-based profiling model. We will first begin by introducing the folksonomy dataset and other experimental settings. Then, we investigate the performance of the personalization approaches that fully considers both query-resource relevance, and user-resource relevance.

A. Experiment Setup

For our experiment, the following two popular folksonomy dataset was used:

The first dataset is CiteULike[2], a publicly available dataset that offers social bookmarking features to scientific papers. We only extracted the annotations of users who have more than 100 tags for at least more than a month, to focus on the cases where user interests have drifted as time passes. We then use 90% of the bookmarks to create the user profile, and the remaining 10% to be used for testing dataset. After removing unnecessary tags such as 'no-tag' and '-' which are automatically generated when the user inputs nothing for annotation, the test bed contained 124,236 documents and 63,559 distinct tags. The average unique number of tags per user is 64.82, while each resource has an average of 2.87 tags as its annotation. The small amount of resource tags indicates that this dataset is very sparse. Although data sparseness is known to be the main reason of reducing effectiveness of graph clustering [13], we can still verify our algorithm on this dataset to be robust, even though it suffers from the limited number of sources.

[2] http://www.citeulike.org

Authorized licensed use limited to: ETRI. Downloaded on March 18,2021 at 06:19:13 UTC from IEEE Xplore. Restrictions apply.

To further evaluate our method, we use another large dataset, MovieLens, which consists of 95,580 tuples of users, movies, and tags; this includes a breakdown of 71,567 users and 10,681 movies. While the user has an average of 10.6221 tags, each resource has 9.0884 tags, indicating that this dataset is a relatively denser dataset, compared to the CiteULike dataset.

To collect the results of $\mathrm{r}_{\mathrm{query-resource}}(q, r)$ in equation 8 for the CiteULike dataset, we used Google as a search platform because it is the most successful search system today. Thus, we can set the most restrictive baseline to be compared in section 4.2. We downloaded the top 100 documents in the result list, and discarded cases where the target resource is not found in the result list. After the download, the average position of the target resource on the result list is 12.43, which seems to be the most highly ranked baseline compared to other related studies [4, 15, 18]. On the other hand, for MovieLens, ranking functions are used to calculate the relevance between query and resources, instead of using ranking retrieved by web services.

We used tf-iuf [18], bm25 [15], hybrid [3][15], and ntf [4] as compared methods to construct user and resource profiles. We then compared their performance with respect to the two different metrics: imp (Ranking Improvement) [13], and Mean Reciprocal Ranking (MRR) [16]. MRR assigns a value for a target resource r of $1/p$, where p is the position of the resource r in the final result list. Ranking improvement measures the difference between $1/p$ and $1/p_0$, where p_0 is the position of the resource r in the initial result r_0.

Moreover, the performance of the two metrics (imp and MRR) are evaluated by using three state-of-the-art ranking measures, as denoted above, to compute the similarity between user and resource profiles: Cosine similarity, scalar similarity, and user similarity (denoted as CS, SS, US respectively for the rest of the paper). This is to examine if our approaches show consistent performance, regardless of the ranking function.

B. *Results of Folksonomy-based Personalization*

Table 2 shows the retrieval results for all data collections and metrics. Values in bold highlights the highest values for each metric. For imp metric, the values of r_0 are set to be the baseline. As a baseline, we used the initial result, where the user-resource relevance scores are not considered. The baseline for CiteULike is based on the initial ranking retrieved by Google.

On the other hand, the baseline for MovieLens is based on the initial ranking returned by each ranking function solely used to measure query-resource relevance. Thus, notice that the values of the baseline are the same regardless of the ranking functions for CiteULike collection, whilst the values are different depending on ranking functions in MovieLens collection.

The performance results of the state-of-the-art methods show that the ntf and the hybrid method perform better than the others, which is in accordance with the reports in [4, 15]. However, although Vallet et al. [15] reported that the ranking function SS and US perform better than CS, in our experiment, the performance does not show a statistical improvement.

A possible reason for this inconsistency is the difference between Vallet et al's and our evaluation setups. Vallet et al. used CombSUM method [14] that simply adds the relevance score of user-resource and query-resource. However, as Google does not show the scores for the resources in the search list, we alternatively used our own ranking function that exploits the position *p* of the target resource *r* in the initial and personalized list.

In Table 2, the improvements over FolkRank and AMI-Frank methods were statistically significant (using Wilcoxon test, $p < 0.05$) in all ranking functions and data collections. Although hybrid and ntf, the best performing state-of-the-art

Table 2. The performance of personalized search (r.f. denotes ranking function)

r.f.	Metric	Learning Profile Techniques on *CiteULike*						
		baseline	bm25	tf-iuf	hybrid	ntf	FolkRank	AMI-Frank
CS	MRR	0.4531	0.5292	0.5362	0.5340	0.5266	0.5474	**0.5519**
	imp.	-	0.0760	0.0830	0.0809	0.0734	0.0942	**0.0987**
SS	MRR	0.4531	0.5306	0.5305	0.5357	0.5228	0.5488	**0.5506**
	imp.	-	0.0774	0.0773	0.0825	0.0696	0.0956	**0.0974**
US	MRR	0.4531	0.5318	0.5245	0.5258	0.5296	0.5438	**0.5447**
	imp.	-	0.0786	0.0713	0.0726	0.0764	0.0906	**0.0915**
r.f.	Metric	Learning Profile Techniques on *MovieLens*						
		baseline	bm25	tf-iuf	hybrid	ntf	FolkRank	AMI-Frank
CS	MRR	0.3840	0.3881	0.3908	0.4069	0.3924	0.5262	**0.5950**
	imp.	-	0.0131	0.0225	0.0288	0.0228	0.0941	**0.1465**
SS	MRR	0.3805	0.3801	0.3835	0.3998	0.3998	0.5120	**0.5945**
	imp.	-	0.0108	0.0112	0.0224	0.0223	0.1008	**0.1512**
US	MRR	0.3907	0.3920	0.3937	0.4002	0.4228	0.5330	**0.6245**
	imp.	-	0.0078	0.0110	0.0221	0.0362	0.1202	**0.2022**

Table 3. An average comparison result with state-of-the-art profiling models (vs. AMI-Frank) on MRR metric

	baseline	bm25	tf-iuf	hybrid	ntf
CiteULike	21.18%	3.49%	3.52%	3.24%	4.32%
MovieLens	57.03%	56.35%	55.31%	50.30%	49.30%

[3] As Vallet et al. suggested, CombSUM is used to make an aggregation of bm25 and tf-iuf.

Authorized licensed use limited to: ETRI. Downloaded on March 18,2021 at 06:19:13 UTC from IEEE Xplore. Restrictions apply.

Table 4. The performance of graph-based profiling models (r.f. denotes ranking function)

r.f.	Metric	*CiteULike*			*MovieLens*		
		FolkRank	AFrank	AMI-Frank	FolkRank	AFrank	AMI-Frank
CS	MRR	0.3149	0.3283	**0.3321**	0.2422	0.2844	**0.3177**
	imp.	0.0392	0.0548	**0.0570**	0.0388	0.1055	**0.1388**
SS	MRR	0.3149	0.3356	**0.3368**	0.2518	0.3012	**0.3288**
	imp.	0.0322	0.0532	**0.0539**	0.0366	0.0922	**0.1280**
US	MRR	0.3339	0.3430	**0.3461**	0.2639	0.3048	**0.3489**
	imp.	0.0392	0.0547	**0.0570**	0.0331	0.0988	**0.1305**

Table 5. An average comparison result of graph-based models (vs. AMI-Frank) on MRR metric

	FolkRank	AFrank
CiteULike	5.32%	0.8%
MovieLens	31.33%	11.79%

techniques, also show statistical improvement over the baseline, the improvement is relatively small. In particular, FolkRank and AMI-Frank show a dramatic improvement in MovieLens, as the collection is much denser. In CiteULike, those methods still show a promising robustness although it suffers from the limited number of tags to represent resources. Furthermore, our AMI-Frank algorithm presents the best performing values in all metrics and ranking functions. It indicates that the consideration of time and multiple interests of users positively affect user modeling. We can find that our profiling model outperforms the baseline by 9.58%, and 16.66% in CiteULike, and MovieLens respectively, on imp. Table 3 summarizes comparison results that reveal how AMI-Frank outperformed other state-of-the-art models on MRR, by indicating that AMI-Frank outperformed the baseline by 21.18% and 57.03% in CiteULike and MovieLens, respectively.

To gain further insight on the impact of adaptivity and multiplicity on user interests, we analyze the performance of the personalization approaches when only the relevance between user and resources is used to reorder the initial search results. This implies that the relevance between query and resource is not taken into account. We again denote here three different graph-based profiling techniques: FolkRank (naive graph-based model), AFrank (the graph-based model that only considers the adaptivity), and lastly our algorithm, AMI-Frank (the graph-based model that considers both adaptivity and multiplicity simultaneously).

The results in Table 4 show the personalization performance when each factor is added to the initial graph-based model. All improvements are statistically significant (using Wilcoxon test, $p < 0.05$) in all ranking functions and data collections. Values in bold highlights the highest values for each metric. For imp metric, the values of r_0 are set to be bm25 as it showed the lowest performance among other profiling models. It can be observed that this result is consistent with the above personalization result, in which our profiling model, AMI-Frank, outperforms other state-of-the-art profiling models including naive graph-based profiling models. According to Table 4, our method outperforms bm25 weighting scheme by 5.55% in CiteULike, and 13.24% in MovieLens on imp. Furthermore, in terms of MRR metric, it outperforms FolkRank by 5.32% in CiteULike and 31.33% in MovieLens, as shown in Table 5.

This result highlights two aspects: Despite multiple interests not being considered, the overall performances of AFrank increased at a significant level compared to FolkRank, proving the importance of considering time aspect in the algorithm. Finally, incorporating both factors, adaptivity and multiplicity, have shown a larger improvement than only incorporating adaptivity, as AMI-Frank outperforms AFrank.

V. Discussions and Conclusions

In this study, we have presented a graph-based profiling scheme that exploits the structure of folksonomy and demonstrated that our graph-based approaches outperform the existing profile schemes. In particular, a graph-based personalization algorithm that adapts to the change in the user's interest over time has been shown as the most effective personalization technique among the other algorithms examined.

It should also be noted that we have not fully overcome the data sparseness issue when creating profiles for users and resources. Although the experiment shows that our approach represents a robust improvement with a folksonomy data, the improvement was somewhat limited by the sparseness of the data. In our future study, we need to further improve our algorithm to minimize the negative effects from data sparseness. Finally, we need to consider a scalable strategy if the proposed personalization algorithm is to be applied to a general Web search system. Despite these limitations, the proposed approaches show promising results for tapping into the potential of folksonomy for the enhancement of personal search.

Acknowledgements

We thank the anonymous reviewers for their helpful comments. This work was supported by ICT R&D Program of MSIP/ITTP (10044494).

References

[1] C.M. Au-Yeung, N. Gibbins, and N. Shadbolt, "A study of user profile generation from folksonomies," In *Proc. of WWW Workshop*, 2008.

[2] S. Bao, G. Xue, X. Wu, Y. Yu, B. Fei, and Z. Su, "Optimizing web search using social annotations," In *Proc. of WWW*, pp. 501-510, 2007.

[3] V.D. Blondel, J.L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of statistical mechanics: Theory and Experiment*, vol. 10, 2008.

[4] Y. Cai and Q. Li, "Personalized Search by Tag-based User Profile and Resource Profile in Collaborative Tagging Systems," In *Proc. of CIKM*, pp. 969-978, 2010.

Authorized licensed use limited to: ETRI. Downloaded on March 18,2021 at 06:19:13 UTC from IEEE Xplore. Restrictions apply.

[5] M. Daoud, L. Tamine, and M. Boughanem, “A personalized graph-based document ranking model using a semantic user profile,” In *Proc. of UMAP*, pp. 171-182, 2010.

[6] M. Dorigo and G.D. Caro, “New ideas in optimization,” pp. 11-32, McGraw-Hill, 1999.

[7] A. G. Golder and B.A. Huberman, “Usage patterns of collaborative tagging systems,” *Journal of information science*, vol. 32, no. 2, pp. 198-208, 2006.

[8] H. Kim, M. Rawashdeh, and A.E. Saddik, “Leveraging collaborative filtering to tag-based personalized search,” In *Proc. of UMAP*, pp. 195-206, 2011.

[9] A. Hotho, R. Jaschke, C. Schmitz, and G. Stumme, “Information retreival in folksonomies: search and ranking,” In *Proc. of ESWC*, pp. 411-426, 2006.

[10] E. Michlmayr and S. Cayzer, “Learning user profiles from tagging data and leveraging them for personal(ized) information access,” In *Proc. of WWW Workshop*, 2007.

[11] M.G. Noll and C. Meinel, “Web search personalization via social bookmarking and tagging,” In *Proc. of ISWC*, pp. 195-230, 2007.

[12] L. Page, S. Brin, R. Motwani, and T. Winograd, “The PageRank Citation Ranking: Bringing Order to the Web,” *Technical report*, Stanford Digital Library Technologies Project, 1998.

[13] A. Shepitsen, J. Gemmell, B. Mobasher, and R. Burke, “Personalized recommendation in social tagging systems using hierarchical clustering,” In *Proc. of RecSys*, pp. 259-266, 2008.

[14] J.A. Show and E.A. Fox, “Combination of multiple searches,” In *Proc. of TREC*, pp.632-648, 1993.

[15] D. Vallet, I. Cantador, and J. M. Jose, “Personalizing web search with folksonomy-based user and document profiles,” In *Proc. of ECIR*, pp. 420-431, 2010.

[16] E. Voorhees, “The TREC-8 question answering track report,” In *Proc. of TREC*, pp. 77-82, 1999.

[17] H. Xie, Q. Li, and Y. Cai, Y, “Community-aware profiling for personalized search in folksonomy,” *Journal of computer science and technology*, vol. 27, no. 3, pp. 599-610, 2012.

[18] S. Xu, S. Bao, B. Fei, Z. Su, and Y. Yu, “Exploring folksonomy for personalized search,” In *Proc. of SIGIR*, pp. 155-162, 2008.

Authorized licensed use limited to: ETRI. Downloaded on March 18,2021 at 06:19:13 UTC from IEEE Xplore. Restrictions apply.