Personalized PageRank Vectors for Tag Recommendations: Inside FolkRank

Heung-Nam Kim
University of Ottawa
800 King Edward, Ottawa,
Ontario, K1N 6N5, Canada
hnkim@mcrlab.uottawa.ca

Abdulmotaleb El Saddik
University of Ottawa
800 King Edward, Ottawa,
Ontario, K1N 6N5, Canada
abed@mcrlab.uottawa.ca

ABSTRACT
This paper looks inside FolkRank, one of the well-known folksonomy-based algorithms, to present its fundamental properties and promising possibilities for improving performance in tag recommendations. Moreover, we introduce a new way to compute a differential approach in FolkRank by representing it as a linear combination of the personalized PageRank vectors. By the linear combination, we present FolkRank's probabilistic interpretation that grasps how FolkRank works on a folksonomy graph in terms of the random surfer model. We also propose new FolkRank-like methods for tag recommendations to efficiently compute tags' rankings and thus reduce expensive computational cost of FolkRank. We show that the FolkRank approaches are feasible to recommend tags in real-time scenarios as well. The experimental evaluations show that the proposed methods provide fast tag recommendations with reasonable quality, as compared to FolkRank. Additionally, we discuss the diversity of the top *n* tags recommended by FolkRank and its variants.

Categories and Subject Descriptors
H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – *Information filtering*

General Terms
Algorithms, Performance

Keywords
Folksonomy, FolkRank, Social Tagging, Tag recommendation

1. INTRODUCTION
With the current popularity of social tagging, a number of researchers have concentrated on recommender systems via folksonomies. Additionally, the proliferation of social tagging has created new topics for the research community. One of the emerging topics is a tag recommender system, supporting the suggestion of suitable tags during tagging processes [6]. To induce tags' full potential for facilitating better search results or item recommendations, tag recommender systems help users describe/organize their content with less effort. In general, modern social media services allow users to freely annotate their content with any kind of arbitrary words, e.g., context-based tags, content-based tags, subjective tags, self-referential tags, etc [3]. Since users' motivations and behaviors for tagging content differ from one another, recommending tags may create more challenges, compared to recommending items in traditional recommenders. Moreover, in terms of personalization, the tag recommenders should deal with three-dimensional data, i.e., users, tags, and items, generally leading to require more time-consuming tasks. Therefore, a number of studies have explored various techniques for personalized tag recommendations, such as collaborative filtering [6], link structure analysis [7], association rule mining [9], popularity models [11], tensor decompositions [15], probabilistic models [16], and so on.

One of the most frequently cited studies among folksonomy-based algorithms is FolkRank, which adapts the well-known PageRank for a folksonomy space [10][11]. A number of researchers have considered FolkRank as their performance benchmark or their related work. Nevertheless, there are few studies explicitly investigating what the salient properties behind FolkRank are. In this paper, we aim to look into FolkRank to disclose its fundamental properties and thus to present promising possibilities for improving the performance and overcoming existing drawbacks. We represent FolkRank as personalized PageRank vectors to explain a new probabilistic interpretation. We also introduce approximate FolkRank methods performing on dimensionality-reduced graphs, in turn leading to reduce computational cost. The main contributions of this paper are as follows: 1) We present and formalize the FolkRank model as a linear combination of the personalized PageRank vectors that can efficiently compute tags' rankings without any loss of those ranks; 2) we propose new FolkRank-like methods that can provide fast tag recommendations with reasonable quality; and 3) we empirically analyze how FolkRank and its variants are beneficial to tag recommendations. We also demonstrate a fundamental shortcoming of the FolkRank approaches in terms of the recommendation diversity.

The remainder sections are organized as follows: Section 2 reviews FolkRank in detail. In Section 3, we introduce possible alterations to the basic FolkRank. In Section 4, we analyze computational complexities, while Section 5 presents experiment evaluations comparing the proposed variants to FolkRank. Finally, we present conclusions and future work in Section 6.

Recsys'11, October 23–27, 2011, Chicago, Illinois, USA.

2. REVIEW OF FOLKRANK

We first review the main points of the FolkRank algorithm. Before going into further detail, we first introduce formal models of a folksonomy and the notations used in this paper. There are several ways to model a folksonomy such as a third-order tensor and a hypergraph. For a set of users $U = \{u_1, u_2,...,u_{|U|}\}$, a set of tags $T = \{t_1,t_2,...,t_{|T|}\}$, and a set of items $I = \{i_1, i_2,...,i_{|I|}\}$, a folksonomy is a tuple $F = (U, T, I, Y)$ where $Y \subseteq U \times T \times I$ is a ternary relationship, called *tag assignments*. With respect to the hypergraph, a folksonomy is represented as a three-uniform hypergraph $H = (V, E)$, where $V = U \cup T \cup I$ is the set of nodes and $E \subseteq U \times T \times I$ is the set of three-element subsets of V, called *hyperedges*. A folksonomy can also be viewed as a three-dimensional tensor space on users, tags, and items. Accordingly, the three-dimensional space can be projected onto three two-dimensional matrices. For convenience, we first define three matrices obtained by aggregating over items, tags, and users, respectively: a $|U| \times |T|$ user-tag matrix $\mathbf{M}_{UT}$, a $|U| \times |I|$ user-item matrix $\mathbf{M}_{UI}$, and a $|T| \times |I|$ tag-item matrix $\mathbf{M}_{TI}$. Each entry of the matrices represents the number of times that the corresponding row and column co-occurred in Y. For example, for a given user u and tag t, the entry's value in the user-tag matrix $\mathbf{M}_{UT}$ implies the number of items that user u has tagged with tag t. In this paper, matrices are denoted by using boldface, upper-case letters, such as $\mathbf{A}$; whereas the corresponding italicized, upper-case letters with two subscript indices represent entries in the matrices, such as $A_{u,t}$. We always use boldface lower-case letters, such as $\mathbf{r}$, to denote vectors, and denote by $\mathbf{r}(t)$ the tth element of the vector $\mathbf{r}$.

2.1 Adapting PageRank for Folksonomy

Hotho *et al*. [10] converted the folksonomy hypergraph into an undirected tripartite graph to propose a graph-based ranking algorithm, called *FolkRank*. Let $G_{TRI} = (U \cup T \cup I, E)$ denote an *un*directed *tri*partite graph whose *nodes* can be partitioned into three disjoint sets, the set of user nodes U, the set of tag nodes T, and the set of item nodes I, such that every node of each set is adjacent to at least one node in the each other set. That is, each *edge* in E connects the corresponding row and column of nonzero entries in the matrices, $\mathbf{M}_{UT}$, $\mathbf{M}_{UI}$, and $\mathbf{M}_{TI}$. Therefore, the $N \times N$ weighted adjacency matrix[1] $\mathbf{A}$ of the graph G_{TRI} is represented by:

$$\mathbf{A} = \begin{pmatrix} \mathbf{0}_{UU} & \mathbf{M}_{UT} & \mathbf{M}_{UI} \\ \mathbf{M}_{TU} & \mathbf{0}_{TT} & \mathbf{M}_{TI} \\ \mathbf{M}_{IU} & \mathbf{M}_{IT} & \mathbf{0}_{II} \end{pmatrix} \quad (1)$$

where $\mathbf{0}_{UU}$, $\mathbf{0}_{TT}$ and $\mathbf{0}_{II}$ are $|U| \times |U|$, $|T| \times |T|$, and $|I| \times |I|$ zero matrices, respectively. Additionally, $\mathbf{M}_{TU}$, $\mathbf{M}_{IU}$, and $\mathbf{M}_{IT}$ are the transpose matrices of $\mathbf{M}_{UT}$, $\mathbf{M}_{UI}$, and $\mathbf{M}_{TI}$, respectively.

From the tripartite graph, FolkRank employs a PageRank's random surfer model to rank tags (or items) in a tag recommendation scenario (or a tag-based search scenario). In this paper, we focus on ranking tags for given user-item pairs.

Formally, a PageRank (column) vector $\mathbf{r}$ is defined as the solution of the following equation:

[1] For the unweighted graph, we can set all non-zero entries of matrices to one.

$$\mathbf{r} = d\tilde{\mathbf{A}}\mathbf{r} + (1-d)\mathbf{p} \quad (2)$$

where $d \in (0, 1)$ is a damping factor and $\tilde{\mathbf{A}}$ is a *column* stochastic matrix of the adjacency matrix $\mathbf{A}$, i.e., the *transition probability matrix* where each column sums to 1. In addition, $\mathbf{p}$ is a preference vector that enables not only forcing the strongly connected graph for G_{TRI}, but also computing the personalized PageRank vector $\mathbf{r}$. If $\mathbf{p}$ is the uniform distribution vector $\mathbf{p} = [1/N, \ldots, 1/N]^T$ such that $N = |U| + |T| + |I|$, then the solution $\mathbf{r}$ is the *global PageRank vector*. Otherwise, the solution $\mathbf{r}$ refers to as the *personalized PageRank vector* for the corresponding preference vector $\mathbf{p}$ [12]. We later discuss the preference vector in more detail.

More precisely, for a given tag t, the PageRank score of t, $\mathbf{r}(t)$, is computed by:

$$\mathbf{r}(t) = d\sum_{v \in I(t)} \frac{A_{v,t} \cdot \mathbf{r}(v)}{\sum_{j=1}^{N} A_{v,j}} + (1-d)\mathbf{p}(t) \quad (3)$$

where $I(t)$ is the set of in-neighbors, i.e., nodes that link to tag t, and $A_{v,t}$ refers to the value of the tth column of the vth row in $\mathbf{A}$.

2.2 Computing the FolkRank Vector

PageRank basically exploits the hyperlink structure of the Web with a Markov chain model to compute the stationary (steady-state) distribution vector $\mathbf{r}$ [5]. Unlike the general Web graph, however, the graph G_{TRI} used in a folksonomy is an undirected, weighted tripartite. Hotho *et al*. [10] observed that a fraction of the probability of moving from any node i to any node j at a given step returned back at the next step due to the nature of G_{TRI}. To solve this problem, FolkRank employed a *differential* approach, capturing how PageRank with the non-uniform preference vector (i.e., the personalized PageRank vector) differs from PageRank with the uniform distribution vector (i.e., the global PageRank vector). Formally, the FolkRank vector of a given user $u \in U$ for a given item $i \in I$ is obtained as follows:

$$\mathbf{f} = \mathbf{r}^{(1)} - \mathbf{r}^{(0)} \quad (4)$$

where $\mathbf{r}^{(1)}$ is the personalized PageRank vector obtained with the non-uniform vector $\mathbf{p}$ where $\mathbf{p}(u) = 1 + |U|$, $\mathbf{p}(i) = 1 + |I|$, and $\mathbf{p}(v) = 1$ for all v if $v \neq u$ and $v \neq i$. And $\mathbf{r}^{(0)}$ is the global PageRank vector obtained with the uniform vector $\mathbf{p} = \mathbf{1}$, where $\mathbf{1}$ is the vector consisting of all ones, $\mathbf{1} = [1, \ldots, 1]^T$. The resulting vector $\mathbf{f}$ is called the FolkRank vector; an entry tag t, $\mathbf{f}(t)$, is called the FolkRank of tag t for given user-item pair (u, i).

In the early paper describing FolkRank [10], $\mathbf{r}^{(0)}$ was obtained without the preference vector (i.e., $d = 1$) in Equation (2), which may result in the different FolkRank vector $\mathbf{f}$ described above. The paper might assume that every node was reachable from every other node in a folksonomy space (an *irreducible* graph) and there was no a *rank sink* problem defined in [4]; more generally, the transition matrix $\tilde{\mathbf{A}}$ of the Markov chain is *irreducible* and *aperiodic*. By definition, the graph G_{TRI} is aperiodic. In our point of view, however, a folksonomy graph is not necessarily strongly connected similar to the Web graph although the folksonomy graph, by definition, does not have *dangling links* that point to any node with no out-neighbors [4]. Figure 1 (a) illustrates a simple example of an *irreducible* folksonomy graph, whereas Figure 1 (b) shows a *reducible* one. If

$\tilde{\mathbf{A}}$ is not irreducible, the PageRank vector with $d = 1$ may not be the unique dominant eigenvector [5]; the power iteration applied to the transition matrix $\tilde{\mathbf{A}}$ may converge to multiple outcomes depending on the choice of an initial vector for $\mathbf{r^{(0)}}$. In addition, $\mathbf{r^{(0)}}$ could contain some zero rank entries and the final rank vector $\mathbf{f}$ could contain negative rank values. To force irreducibility on the Web graph, PageRank employs necessarily the vector $\mathbf{p}$ with the damping factor $0 < d < 1$ [13].

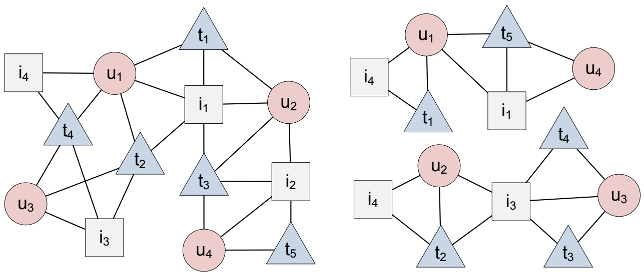

Figure 1. (a) An irreducible graph in which every node is reachable from every other node and (b) a reducible graph in which some nodes cannot move to some other nodes.

3. APPROXIMATE FOLKRANK

As referred to earlier, FolkRank performs on the graph with three disjoint sets: the user set U, the tag set T, and the item set I. In general, the power iterations can effectively compute the eigenvector by using a extreme sparse matrix such as $\tilde{\mathbf{A}}$, i.e., sparse matrix-vector multiplication [13]. With large-scale data, PageRank quickly converged to a tolerance level with small iterations [4]. Nevertheless, the folksonomy graph G_{TRI} still contains nodes of considerable dimensions. Moreover, the FolkRank vector includes unnecessary ranking elements in a situation of recommending tags, such as users' rankings and items' rankings. To overcome these problems, we propose approximate FolkRank algorithms working on dimensionality-reduced graphs so that they can accelerate to provide tag recommendations for users.

3.1 FolkRank as Personalized PageRank Vectors

First, we interpret FolkRank as the personalized PageRank vector. When the preference vector $\mathbf{p}$ is a non-uniform distribution vector, Jeh and Widom [12] explained the random surfer model with a *teleportation* probability $(1 - d)$: if a certain surfer is looking at page p, at the next time step, he/she looks at a random out-neighbor page of p with probability d, or jumps back to a random *favor* page among their *preference pages* with probability $(1 - d)$.

For representing FolkRank as the personalized PageRank vector, let $\mathbf{p^{(1)}}$ and $\mathbf{p^{(0)}}$ be the preference vectors for $\mathbf{r^{(1)}}$ and $\mathbf{r^{(0)}}$, respectively. When rewriting Equation (4), we have that

$$\begin{aligned} \mathbf{f} &= (d\tilde{\mathbf{A}}\mathbf{r^{(1)}} + (1 - d)\mathbf{p^{(1)}}) - (d\tilde{\mathbf{A}}\mathbf{r^{(0)}} + (1 - d)\mathbf{p^{(0)}}) \\ &= (d\tilde{\mathbf{A}}\mathbf{r^{(1)}} - d\tilde{\mathbf{A}}\mathbf{r^{(0)}}) + ((1 - d)\mathbf{p^{(1)}} - (1 - d)\mathbf{p^{(0)}}) \\ &= d\tilde{\mathbf{A}}(\mathbf{r^{(1)}} - \mathbf{r^{(0)}}) + (1 - d)(\mathbf{p^{(1)}} - \mathbf{p^{(0)}}) \\ &= d\tilde{\mathbf{A}}\mathbf{f} + (1 - d)(\mathbf{p^{(1)}} - \mathbf{p^{(0)}}) \end{aligned} \tag{5}$$

Let $\mathbf{p}' = \mathbf{p^{(1)}} - \mathbf{p^{(0)}}$. Recall that the entries of $\mathbf{p^{(1)}}$ are set to 1 except for the entries of the target user u ($\mathbf{p^{(1)}}(u) = 1+|U|$) and item i ($\mathbf{p^{(1)}}(i) = 1 + |I|$), whereas $\mathbf{p^{(0)}}$ consists of all ones. Consequently, the entries of the vector $\mathbf{p}'$ are $\mathbf{p}'(u) = |U|$ for user u, $\mathbf{p}'(i) = |I|$ for item i, and 0 for all the other nodes. Since $\mathbf{p}'$ is a non-negative vector, we can rewrite Equation (5) as:

$$\mathbf{f} = d\tilde{\mathbf{A}}\mathbf{f} + (1 - d)\mathbf{p}' \tag{6}$$

such that $\| \mathbf{f} \|_1 = |U| + |I|$. As we can see the above equation, the global PageRank vector is not necessarily needed to compute the FolkRank. Without the differential process, we can directly compute the FolkRank. Analogous to the standard PageRank, we can make the sum of all rankings be one, $\| \mathbf{f} \|_1 = 1$, as follows:

$$\mathbf{f} = d\tilde{\mathbf{A}}\mathbf{f} + \frac{(1-d)}{|U|+|I|}\mathbf{p}' \tag{7}$$

Since we are interested in ranking orders according to the values of $\mathbf{f}$, rather than the exact values of it, the resulting ranking order from Equation (6) (or Equation (4)) is exactly identical to that from Equation (7). The solution vector $\mathbf{f}$ may be unique for a given vector $\mathbf{p}'$ regardless of starting from any initial vector for $\mathbf{f}$. In some cases, however, $\mathbf{f}$ resulted from Equation (4) or (7) may contain some zero entries depending on the graph structure and a given user-item pair, i.e., $\mathbf{f}$ could be a non-negative vector. In this situation, only limited tags could be recommended to a target user, which might be an undesirable result. For example, when the target user is u_4 and the target item is i_4 in the example graph of Figure 1 (b), only two tags t_1 and t_5 have positive ranks. The rankings of the other tags such as t_2, t_3, and t_4 are zeros. This problem is also known as rank sinks in PageRank [4]. In terms of the Markov chain, it is *absorbing*. However, for given user-item pair (u_2, i_1), $\mathbf{f}$ is a positive vector, implying all values of $\mathbf{f}$'s entries are greater than zero, even though the example graph is not irreducible; at each step, a surfer is able to teleport from one irreducible closed subset to the other one.

We now represent FolkRank as a combination of some individual personalized PageRank vectors. Let $\mathbf{r}_p$ denote the personalized PageRank vector for a preference vector with a *single* nonzero entry at p. Jeh and Widom called $\mathbf{r}_p$ a *basis vector* [12]. An entry j of $\mathbf{r}_p$, $\mathbf{r}_p(j)$, is j's importance in p's view. For any preference vector $\mathbf{p}_a$ and $\mathbf{p}_b$, if $\mathbf{r}_a$ and $\mathbf{r}_b$ are the two corresponding personalized PageRank vectors, then for any constants α_a, α_b such that $\alpha_a + \alpha_b = 1$, the following *linearity* holds [12]:

$$\alpha_a\mathbf{r}_a + \alpha_b\mathbf{r}_b = d\tilde{\mathbf{A}}(\alpha_a\mathbf{r}_a + \alpha_b\mathbf{r}_b) + (1-d)(\alpha_a\mathbf{p}_a + \alpha_b\mathbf{p}_b) \tag{8}$$

By the above linearity, Equation (7) can be regarded as a linear combination of two basis vectors, $\mathbf{r}_u$ for user u and $\mathbf{r}_i$ for item i:

$$\mathbf{f} = \alpha_u\mathbf{r}_u + \alpha_i\mathbf{r}_i \tag{9}$$

where $\alpha_u = |U| / (|U| + |I|)$ and $\alpha_i = |I| / (|U| + |I|)$. Therefore, for a given user-item pair (u, i), the FolkRank of a certain tag $t \in T$, $\mathbf{f}(t)$, is t's importance in user u's view (i.e., $\mathbf{r}_u(t)$) and t's importance in item i's view (i.e., $\mathbf{r}_i(t)$). We refer to FolkRank defined by Equation (9) (or Equation (7)) as *Tripartite FolkRank* (*TriFolkRank* for short). With respect to the probabilistic interpretation, $\mathbf{f}(t)$ is the score that combines the long-term probability of a random walk starting from user u and ending at tag t with that of a random walk starting from item i and ending at tag t. When $\alpha_i = 1$ and hence $\alpha_u = 0$, the resulting vector $\mathbf{f}$ has rankings only for a given item. Accordingly, it is a non-personalized tag recommendation. TriFolkRank gives an advantage over the standard approach because if some basis vectors are available for some preference vectors, then the ranking vector can be easily obtained from any combination of pre-computed basis vectors according to the linear property. For example, once we have $\mathbf{r}_i$ for item i, it can be used in computing $\mathbf{f}$ for any other user. Similarly, $\mathbf{r}_u$ for user u can combine with any other item to recommend tags for the corresponding user.

3.2 FolkRank on a Bipartite Graph

In terms of the personalized PageRank vector, TriFolkRank periodically restarts a random walk from a given user-item pair. It was empirically observed that the tags linking from either a given user or a given item tended to obtain higher probabilities, as compared to other tags. This result is also affected by a key characteristic of the undirected, tripartite graph. Inspired by the observation, we remove either the user set U or the item set I from G_{TRI}. Subsequently, we build two bipartite graphs: a bipartite graph with U and T and a bipartite graph with T and I. This section presents a tag recommendation method using the tag-item bipartite graph in detail (due to the space consideration). We name this method *Bipartite FolkRank* (*BiFolkRank* for short).

Let $G_{BI} = (T \cup I, E')$ denote an *un*directed *bi*partite graph whose nodes can be partitioned into two disjoint sets such that $E' \subseteq T \times I$ and $E' \subset E$, i.e., G_{BI} is a subgraph of G_{TRI}. Analogous to G_{TRI}, G_{BI} has the following weighted biadjacency matrix:

$$\mathbf{B} = \begin{pmatrix} \mathbf{0}_{TT} & \mathbf{M}_{TI} \\ \mathbf{M}_{IT} & \mathbf{0}_{II} \end{pmatrix} \tag{10}$$

Additionally, we can express BiFolkRank as the solution to:

$$\mathbf{b} = d\widetilde{\mathbf{B}}\mathbf{b} + (1-d)\mathbf{p} \tag{11}$$

where $\widetilde{\mathbf{B}}$ is a column-stochastic matrix of $\mathbf{B}$. When computing BiFolkRank, in place of the TriFolkRank's preference vector $\mathbf{p}'$ described in Section 3.1, we use another preference vector for a given user-item pair (u, i).

Let $\mathbf{p}_x$ denote a *basis preference vector* in which the xth entry is 1 and all other entries are 0. Let $\mathbf{b}_x$ be the basis vector for $\mathbf{p}_x$ as the solution to Equation (11). If $x \in T$ is a tag, $\mathbf{b}_x$ refers to as a *tag basis vector*, and if $x \in I$ is an item, it refers to as *an item basis vector*. For given user u, we aggregate the tag basis vectors $\mathbf{b}_t$ for that user. Formally, we can express the user u's BiFolkRank vector in the tag space as the sum of some tag basis vectors with those non-negative coefficients as follows:

$$\mathbf{b}^{(\mathbf{u})} = \sum_{t \in T(u)} \omega_t \mathbf{b}_t \tag{12}$$

where $T(u)$ is the set of tags used by user u. We maintain the sum of all coefficients with 1, $\sum_t \omega_t = 1$; thus $\| \mathbf{b}^{(\mathbf{u})} \|_1 = 1$. For given tag t, the coefficient for user u is obtained by:

$$\omega_t = \frac{\mathbf{M}_{u,t}}{\sum_{j=1}^{|T|} \mathbf{M}_{u,j}} \tag{13}$$

where $\mathbf{M}$ represents the user-tag matrix $\mathbf{M}_{UT}$. To recommend tags to user u for item i, we compute the final BifolkRank vector by adding item i's basis vector $\mathbf{b}_i$ to user u's $\mathbf{b}^{(\mathbf{u})}$ with appropriate weights.

$$\mathbf{b} = \alpha_T \mathbf{b}^{(\mathbf{u})} + \alpha_I \mathbf{b}_i \tag{14}$$

such that $\alpha_T + \alpha_I = 1$ (for $\| \mathbf{b} \|_1 = 1$). Similar to TriFolkRank, we assign the weight to the each vector according to proportion of the each node set to the total node set in G_{BI}, i.e., $\alpha_T = |T| / (|T| + |I|)$ and $\alpha_I = |I| / (|T| + |I|)$. For recommending tags, the elements of tags are ranked according to those ranking scores contained in the resulting BiFolkRank vector $\mathbf{b}$. As shown in the proof of [8][12], by the linear property, Equation (14) is equivalent to:

$$\mathbf{b} = d\widetilde{\mathbf{B}}\mathbf{b} + (1-d)\left(\alpha_T \sum_{t \in T(u)} \omega_t \mathbf{p}_t + \alpha_I \mathbf{p}_i \right) \tag{15}$$

which satisfies Equation (11) when we combine the basis preference vectors together, i.e., $\mathbf{p} = \alpha_T \sum_t \omega_t \mathbf{p}_t + \alpha_I \mathbf{p}_i$. Thus, this computation implies the following probabilistic interpretation in terms of the random surfer model: with probability $\alpha_T(1 - d)$, a random tagger teleports to one of their personal tags where the particular tag is chosen non-uniformly at random. In addition, with probability $\alpha_I(1 - d)$, the tagger instead jumps to particular item i. And then at the next step he/she keeps moving to out-neighbor tags of i with probability d where the particular out-neighbor tag is chosen uniformly at random. In this context, the BiFolkRank of a certain tag x, $\mathbf{b}(x)$, is the long-term probability of the random walk that ends at the tag x.

Alternatively, we can build another bipartite graph with U and T. In this case, for the user-item pair (u, i), we use the aggregate tag basis vector for the tags annotated in item i, $\mathbf{b}^{(\mathbf{i})}$, and a user basis vector $\mathbf{b}_u$ with appropriate weights.

3.3 Tag-Centric FolkRank

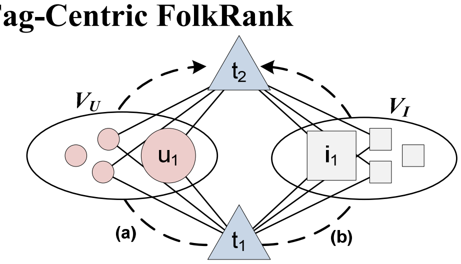

Figure 2. Transition from one tag to another tag (a) through the user set and (b) through the item set.

Even though BiFolkRank reduces one node set compared to TriFolkRank, it still has unnecessary ranking entries in a tag recommendations scenario such as items' rankings or users' rankings. In this section, we construct a graph G_T base solely on the tag set T; G_T is now neither a tripartite graph nor a bipartite graph. As illustrated in Figure 2, there are two possible transition probabilities from one tag (state) to another tag (state) in two steps: transitions through the user node set U and transitions through the item node set I. For example, when flowing through users as intermediate nodes, the transition probability from tag t_1 to tag t_2 is the sum of individual probabilities computed by the probability from t_1 to a user multiplied by that from the user to t_2.

In an analogous fashion, we can compute the transition probabilities flowing through the item set.

More formally, we define two tag-tag stochastic matrices as follows:

$$\overline{\mathbf{W}}_{TT}^{U} = \overline{\mathbf{M}}_{TU}\overline{\mathbf{M}}_{UT}, \quad \overline{\mathbf{W}}_{TT}^{I} = \overline{\mathbf{M}}_{TI}\overline{\mathbf{M}}_{IT} \tag{16}$$

where $\overline{\mathbf{M}}_{TU}, \overline{\mathbf{M}}_{UT}, \overline{\mathbf{M}}_{TI}$, and $\overline{\mathbf{M}}_{IT}$ are row-stochastic matrices of $\mathbf{M}_{TU}$, $\mathbf{M}_{UT}$, $\mathbf{M}_{TI}$, and $\mathbf{M}_{IT}$, respectively. When combining the two defined matrices, we can obtain a new transition matrix in the tag space:

$$\overline{\mathbf{W}} = \alpha_U \overline{\mathbf{W}}_{TT}^{U} + \alpha_I \overline{\mathbf{W}}_{TT}^{I} \tag{17}$$

such that $\alpha_U + \alpha_I = 1$. In our study, we use $\alpha_U = |U| / (|U| + |I|)$ and $\alpha_I = |I| / (|U| + |I|)$. The combined matrix $\overline{\mathbf{W}}$ is also a row-stochastic matrix. Let $\mathbf{t}$ denote the ranking vector working on G_T. When substituting $\overline{\mathbf{W}}^{\mathbf{T}}$ in place of $\widetilde{\mathbf{B}}$ shown in Equation (11), we can solve $\mathbf{t}$ with an appropriate preference vector $\mathbf{p}$ for a given user-item pair (u, i). We refer to this approach as *Tag-centric FolkRank* (*TFolkRank* for short).

Analogous to Equation (12), we aggregate tag basis vectors for user u and item i, respectively, as follows:

$$\mathbf{t}^{(\mathbf{u})} = \sum_{t \in T(u)} \omega_t \mathbf{t}_t, \quad \mathbf{t}^{(\mathbf{i})} = \sum_{t \in T(i)} \varpi_t \mathbf{t}_t \tag{18}$$

where $T(u)$ is the set of tags used by user u and $T(i)$ is the set of tags annotated in item i. In addition, ω_t and ϖ_t represent the coefficients for user u and item i, respectively; we obtain the coefficients using $\mathbf{M}_{UT}$ for the user and $\mathbf{M}_{TI}$ for the items in the manner mentioned in Equation (13). Finally, the TFolkRank vector $\mathbf{t}$ is computed by the weighted sum of the aggregate tag basis vector for user u, $\mathbf{t}^{(\mathbf{u})}$, and the aggregate tag basis vector for item i, $\mathbf{t}^{(\mathbf{i})}$, as follows:

$$\mathbf{t} = \alpha_U \mathbf{t}^{(\mathbf{u})} + \alpha_I \mathbf{t}^{(\mathbf{i})} \tag{19}$$

such that $\alpha_U + \alpha_I = 1$ (for $\| \mathbf{t} \|_1 = 1$). An equivalent form of the combined preference vector $\mathbf{p}$ for computing $\mathbf{t}$ is $\mathbf{p} = \alpha_U \sum_t \omega_t \mathbf{p}_t + \alpha_I \sum_t \varpi_t \mathbf{p}_t$. The intuition behind this preference vector is that, for a target user-item pair, the tags either used by the target user or annotated in the target item are important themselves. In addition, tags highly reachable from those important tags are also important regarding random walks.

4. COMPLEXITY ANALYSIS

In a practical point of view, we analyze the computational complexities of the FolkRank-based methods. For recommending top-n tags to any target user for any given item, Jäschke *et al*. [11] practically proved that *online* complexity of FolkRank is $\mathcal{O}(k(|E| + N))$, where k is the number of iterations, $|E|$ is the total number of edges in the tripartite graph G_{TRI}, and $N = |U| + |T| + |I|$. This online complexity is a stumbling block to real-time performance as already pointed out in [6], [11], and [15]. They additionally required *offline* complexity of $\mathcal{O}(k(|E| + N))$ for computing the global PageRank $\mathbf{r}^{(0)}$. To reduce real-time cost, we efficiently separate the offline complexity from the online complexity by utilizing a linear combination of basis vectors.

Offline complexities closely connect to the time required to compute basis vectors. As for TriFolkRank, we should first compute users' basis vectors $\mathbf{r}_u$, for all $u \in U$, and items' basis vectors $\mathbf{r}_i$, for all $i \in I$. The complexities of these steps are $\mathcal{O}(k|E||U|)$ and $O(k|E||I|)$, respectively; accordingly, the offline complexity for TriFolkRank becomes approximately $\mathcal{O}(k|E||U| + k|E||I|)$. In an analogous fashion, for BiFolkRank with T and I, each tag basis vector $\mathbf{b}_t$, for all $t \in T$, and each item basis vector $\mathbf{b}_i$, for all $i \in I$, should be pre-computed; these steps take $\mathcal{O}(k|E'||T| + k|E'||I|)$ time where $|E'|$ is the total number of edges in the bipartite graph G_{BI}. In the case of BiFolkRank with U and T, the complexity becomes $\mathcal{O}(k|E'||U| + k|E'||T|)$. In general, BiFolkRank costs less than do TriFolkRank due to $|E'| < |E|$. TFolkRank has slightly different cost. We should build the transition matrix $\overline{\mathbf{W}}$ derived from the sparse matrix multiplications. These computations roughly require $\mathcal{O}(m|T|)$ time for $\overline{\mathbf{W}}_{TT}^{U}$ and $\mathcal{O}(m'|T|)$ time for $\overline{\mathbf{W}}_{TT}^{I}$, where m and m' are the number of nonzero entries in $\mathbf{M}_{UT}$ and $\mathbf{M}_{TI}$, respectively. For computing each tag basis vector $\mathbf{t}_t$, for all $t \in T$, we additionally require $\mathcal{O}(km''|T|)$ where m'' is the number of nonzero entries in $\overline{\mathbf{W}}$. Therefore, the total complexity of TFolkRank is $\mathcal{O}(m|T| + m'|T| + km''|T|)$ during the offline step. To reflect up-to-date data, we should update the basis vectors on a regular basis. In that case, we may need not to update all basis vectors, but to update only the basis vector occurring big changes. In addition, each basis vector computation can be easily implemented with parallel computation because its computation is simply the sparse matrix-vector multiplication [13].

Online is more important than offline complexity regarding real-time scenarios [16]. Since we already compute the basis vectors for each algorithm, for any given user-item pair, each method can easily compute ranking scores by combining appropriate basis vectors with those weights. The complexity of the combination step is approximately $\mathcal{O}(|T|)$, which is the time required to add all tags' ranking scores contained in each basis vector. In parallel, we can select top-n ranked tags, which results in the complexity of $\mathcal{O}(n\log(n))$. Thus, the total complexities of TriFolkRank, BiFolkRank, TFolkRank for the top-n recommendation become $\mathcal{O}(n\log(n)|T|)$, which is an approximation to $\mathcal{O}(|T|)$ because n is much less than $|T|$ (usually $n = 10$). Note that we can also compute the rankings with combined preference vectors as in Equation (15), rather than with the pre-computed basis vectors. In this approach, for any given user-item pair, the online complexities of TriFolkRank and BiFolkRank become $\mathcal{O}(k|E|)$ and $\mathcal{O}(k|E'|)$, respectively, while there are no offline steps. As for TFolkRank, its online complexity becomes $\mathcal{O}(km'')$, whereas the offline is $\mathcal{O}(m|T| + m'|T|)$.

5. EXPERIMENTAL RESULTS

5.1 Dataset

The experimental data came from *CiteULike*, a free service for managing and discovering scholarly references. We downloaded the latest data snapshot of CiteULike in November 2010. The original dataset was too sparse to be used for experiments. Therefore, we cleaned the dataset to carry out experiments that were more meaningful. We selected (1) users who have tagged at least five items (papers) and used at least five tags; (2) items that have been labeled with at least five tags and posted by at least

Table 2. Comparisons of Precisions (%) and recalls (%) at the top-5 and top-10, shown with 95% confidence intervals.

	TriFolkRank.	BiFolkRank (TI)	BiFolkRank (UT)	TFolkRank	Popular
P@5	29.01 ± 1.62	30.11 ± 1.65	30.01 ± 1.65	28.08 ± 1.59	24.37 ± 1.58
R@5	52.92 ± 2.78	54.59 ± 2.92	54.46 ± 2.92	51.87 ± 2.67	42.12 ± 2.23
P@10	19.74 ± 1.40	20.23 ± 1.44	20.22 ± 1.42	19.26 ± 1.33	16.87 ± 1.16
R@10	67.62 ± 2.80	68.93 ± 2.86	68.85 ± 2.86	66.64 ± 2.77	55.62 ± 2.69

five users; and (3) tags that at least five users have used and that have labeled to at least five items. The cleaned dataset used in our study contained 161,395 tag assignments. Three two-dimensional matrices obtained from the tag assignments had approximately a sparsity level of 99%. Table 1 briefly summarizes the dataset used in our experiments.

Table 1. The CiteULike dataset.

$\|U\|$	$\|T\|$	$\|I\|$	$\|Y\|$	$\mathbf{M}_{UT}$	$\mathbf{M}_{UI}$	$\mathbf{M}_{TI}$
2,614	2,310	4,096	161,395	62,112	65,325	72,619

5.2 Evaluation Design and Methodology

The experiments were designed to answer the following questions:

- How different are the approximate FolkRank methods from (Tri)FolkRank in terms of the recommendation quality?
- How diverse are tags recommended by the FolkRank-based approaches (the recommendation diversity)?
- How efficient are the approximate approaches, compared to (Tri)FolkRank?

To this end, we compared the performance of BiFolkRank and TFolkRank against that of TriFolkRank. As mentioned earlier, the original FolkRank recommends tags the same as TriFolkRank does; our experimental results confirmed that as well. Accordingly, we reported the TriFolkRank results as the performance of FolkRank. During the experiments, we set the damping factor d to 0.7 according to earlier work [11]. Additionally, to solve the linear equations, we employed a non-iterative approach using algebraic expressions rather than an iterative approach because computation time via the former was also reasonable under the scale of our experimental data. Thus, in our experiments, we did not require the number of the iterations. However, it should be noted that non-iterative approaches would practically be prohibitive, particularly when applied to large-scale datasets [2][13].

To evaluate the recommendation performance, we followed the evaluation procedure described in [11]. For each user, we randomly withheld one of their items and their tags annotated in that item. Subsequently, that item was used as a target item for the user. The remaining tag assignments were employed as a training set for building the graphs. For example, if user u_1 annotated item i_1 with tags, t_1 and t_2, we hid the item and the tags from all tag assignments of user u_1. We then recommend tags to the user for item i_1 and then tried to find tags t_1 and t_2 within the top-n ranked results. To ensure that our results were not sensitive to a test item for each user, we conducted five different runs with different test items for each user. The experimental results reported values that were the averages of five runs. The average size of the sets of tags per user was 3.2.

5.3 Evaluation Metrics

To measure the recommendation quality, we first adopted *precision* and *recall*, which can judge how relevant a set of ranked results is for the user. For a given user-item pair (u, i), *precision* and *recall* at top-n are defined as:

$$P @ n(u) = \frac{|T_u \cap R_u^n|}{|R_u^n|}, \quad R @ n(u) = \frac{|T_u \cap R_u^n|}{|T_u|} \tag{20}$$

where R_u^n is the set of top-n recommended tags for the user-item pair, whereas T_u is the set of tags that user u tagged item i in the test data. The overall precision and recall for all users are computed by averaging the individual precisions and recalls, respectively.

We also measured the Mean Average Precision (MAP) to provide insight regarding a single-value measure of the recommendation quality under a *precision-recall* curve [14]. Formally, MAP for all users is computed by:

$$MAP = \frac{1}{|U|} \sum_{u=1}^{|U|} \left(\frac{1}{|T_u|} \sum_{n=1}^{|T_u|} B_u^n \times P @ n(u) \right) \tag{21}$$

where B_u^n is a binary variable that is 1 if the tag with rank n in the recommended list appears in user u's tag set, T_u, and 0 otherwise.

5.4 Experiment Results

5.4.1 Recommendation quality

We first investigated the differences of the tag recommendation quality between TriFolkRank and the approximate methods. As the baseline, we also reported the performance of the simple method based on tags' popularity for a given item (denoted as *Popular*) [11], which is one of non-personalized tag recommenders.

Table 2 shows the results of precisions and recalls at the top-5 and top-10. As shown in the results, it is clear that the FolkRank-based recommenders performed significantly better than the baseline, *Popular*. When compared to TriFolkRank, the BiFolkRank methods obtained slightly improved precisions and recalls. TFolkRank also provided reasonable performance, compared to TriFolkRank. We examined the average number of the same tags recommended by TriFolkRank and the other approximate methods. As a result, among the top ten tags, on average, roughly 9 tags per user were the same tags. In addition, roughly 4.5 tags among the top five tags were the same. These results indicated that top n tags recommended by the proposed approaches working on dimensionality-reduced spaces were almost identical with those recommended by TriFolkRank (or FolkRank), particularly when n was small. However, we observed that those tags' ranks were slightly different.

Since users had, on average, 3.2 tags for their test pair, precision of each method could never be reached at 100% under cases where $n = 5$ or $n = 10$. Therefore, we measured the MAP values to provide a single-figure measure of quality across recall levels. Figure 3 shows the MAP results obtained by the five methods. As shown in the graph, two types of BiFolkRank obtained almost the same MAP values. The difference of MAP between them appeared statistically insignificant based on a two-tailed paired *t*-test. This meant that both BiFolkRank methods provided tag recommendations nearly having the same ranks for each user. The BiFolkRank approaches obtained approximately 1.8% and 2.7% improvements on MAP, compared to TriFolkRank and TFolkRank, respectively, even though the improvements seemed to be comparatively insignificant. In fact, MAP differences were statistically significant ($p < 0.01$) based on two-tailed paired *t*-tests. When comparing the results achieved by TriFolkRank and TFolkRank, we found that the former slightly outperformed the latter by 0.95%.

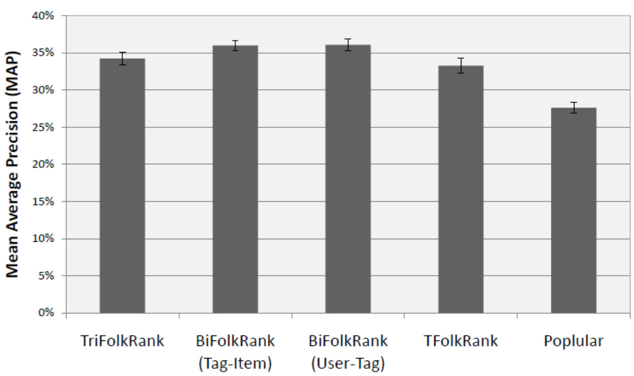

Figure 3. A comparison of MAPs shown with 95% CIs.

When analyzing the experimental data in more detail, we observed that, on average, 88.5% of the total tags occurred in each user-item test pair also appeared in the training tag set of the corresponding user or in the training tag set of the corresponding item. This observation implied that users did tend to annotate an item with their existing tags or, more particularly, with popular tags in that item. This causes the main difference between the tag recommender system and the item recommender system. The item recommender system recommends items not previously rated (or selected) by a given user. However, in the tag recommender system, we should also consider tags used by the user due to their personal tagging behavior. Consequently, the basic premise of the FolkRank-based approaches (including FolkRank) could be beneficial to recommend tags. In this situation, tag recommendations based simply on tags' popularity regarding a current user-item pair would also provide reasonable performance unless the user has used few tags and the item has been labeled with few tags. In some other scenarios, however, the performance would become worse as discussed in the following section.

5.4.2 Recommendation diversity

Most of earlier work for recommending tags conducted experiments as in the previous section. However, some users may annotate an item with new tags not previously labeled to that item. Analogously, a certain user may use new tags not previously used by the user when tagging an item. In consideration of these points, tag recommender systems should also discover some novel tags relevant to a given user-item pair. This is known as the diversity, which is often a desirable feature in recommender systems [1]. In this context, we investigated how the FolkRank-based approaches could recommend novel tags not only regarding a given user, but also regarding a given item. Since we could not find obvious evaluation metrics that measure the diversity of the tags recommended by algorithms, we designed the evaluation from a slightly different viewpoint. For each user-item pair in the test set, we first removed tags from the test set if the tags were previously either used by the user or labeled to the item in the training set. We then tried to discover the remaining tags for the each user-item pair within the top-*n* recommendation. That is, for a given user-item pair, the tags in the test set never co-occurred with the user in the training set and with the item in the training set. When an algorithm recommended such test tags within the top-*n* ranks, we assumed that the algorithm found novel tags different from a homogeneous set of a given user-item pair. In our test set, on average, 24.8% of users (roughly 648 users) had such tags.

Table 3. Precisions (%) and recalls (%) with respect to the diversity, shown with 95% confidence intervals.

	TriFolkR.	BiFolkR. (TI)	BiFolkR. (UT)	TFolkR.
P@10	1.23 ± 0.21	1.49 ± 0.20	1.01 ± 0.13	1.11 ± 0.15
R@10	8.93 ± 1.57	10.86 ± 1.39	7.44 ± 1.13	8.15 ± 1.25

Table 3 shows precisions and recalls with respect to the diversity recommendation. As a result, each algorithm has considerable performance disparity, compared to the top-10 results in Table 2. When recommending tags for a given user-item pair, we also excluded the user's personal tags and the item's tags from a recommendation list. Some algorithms that predict tags' rankings by merely utilizing the tags' popularity according to a given user and/or a given item, such as *Popular*, can never find such test tags any more. On the contrary, the FolkRank-based approaches can predict those tags' rankings, once folksonomy graphs ensure some connectivity. Therefore, they may provide high *coverage* in terms of measuring a percentage of hidden tags for which predictions can be formed. This is an advantage over the popular tag approaches.

However, when we included the rankings of both the user's tags and the item's tags, precisions and recalls obtained by the FolkRank-based approaches were less than 0.1% and 1%, respectively. This implied that those approaches were having problems recommending novel tags having higher ranks within the top-10 recommendation. The top ten tags were barely different from anything that the user has previously used or that has already been labeled to the item. These monotonous recommendations can be a serious disadvantage with respect to the diversity. To overcome this limitation, we will consider some variations on the preference vector **p** in the future. For example, *k*-nearest neighbors could be incorporated into the random surfer model; for a given user-item pair, if a random surfer also teleports their neighbor's nodes with some probabilities instead of jumping to nodes only related to the given pair, it may be possible to provide a wide range of some novel tags having higher ranks for the user.

5.4.3 Runtime

Finally, we present the runtime comparison, showing how fast our approaches work, compared to FolkRank. To measure the runtime, we computed ranking scores with a combined preference vector for a given user-item pair. Table 4 summarizes the runtime results. The column labeled "RTime" shows the total execution time (seconds) required to compute tag recommendations for all users in the test set. And the next column shows the average time required to recommend top-10 tags to each user-item pair.

Table 4. Runtime for making tag recommendations (seconds).

	RTime	RTime per pair
Differential FolkRank	939.464	0.3593
TriFolkRank	492.318	0.1883
BiFolkRank (TI)	171.811	0.0657
BiFolkRank (UT)	67.284	0.0257
TFolkRank	43.679	0.0167

As shown in Table 4, TriFolkRank was 2 times faster than the differential approach in FolkRank. Moreover, the approximate methods recommended tags much faster than did FolkRank and TriFolkRank. For example, on average, TFolkRank made tag recommendations for a single user-item pair in 0.0167 seconds, whereas TriFolkRank took 0.1883 seconds, implying the former was 11 times faster than the latter. When looking at the results obtained by the two types of BiFolkRank, we could see that BiFolkRank with U and T was 2.5 times faster than one with T and I. This result was affected by the fact that the size of the item set $|I|$ was 1.5 times larger than that of the user set $|U|$. This means that we can be flexible in responding to implementations depending on datasets. When $|U| >> |I|$, we can adopt BiFolkRank with T and I. In the opposite case, we can use the other BiFolkRank. These results confirm that our approaches for tag recommendations support fast computation time while maintaining the recommendation quality.

6. CONCLUSION AND FUTURE WORK

In this paper, we look into FolkRank to present the salient concepts and properties behind it. We disclose that the differential approach in FolkRank can be represented simply as a linear combination of the personalized PageRank vectors. We also propose three variants of FolkRank working on the dimensionality-reduced graphs. Our experimental results with the CiteULike dataset demonstrate that the recommendation quality of the proposed methods is closely approximation to that of FolkRank. Particularly, the BiFolkRank methods outperform FolkRank. Moreover, the computation of our approaches is faster than that of FolkRank, leading to create desirable recommender systems.

In future work, we plan to compare our approach to other state-of-art methods using different datasets. To avoid monotonies in recommendation lists, we also plan to incorporate the nearest neighborhood into FolkRank-based approaches.

7. REFERENCES

[1] Adomavicius, G. and Tuzhilin, A. 2005. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. on Knowl. and Data Eng.* 17, 6, 734–749.

[2] Bianchini, M., Gori, M., and Scarselli, F. 2005. Inside PageRank. *ACM Trans. Internet Technol*. 5, 1, 92–128.

[3] Bischoff, K., Firan, C. S., Nejdl, W., and Paiu, R. 2008. Can all tags be used for search?. *In Proc. 17th ACM Conf. Inf. and Knowl. Manage.*, 203–212.

[4] Brin, S., Page, L., Motwami, R., and Winograd, T. 1999. *The PageRank citation ranking: Bringing order to the Web*. Technical Report, Stanford University.

[5] Farahat, A., LoFaro, T., Miller, J.C., Rae, G., and Ward, L.A. 2006. Authority Rankings from HITS, PageRank, and SALSA: Existence, Uniqueness, and Effect of Initialization. *SIAM J. Sci. Comput.* 27, 4. 1181–1201.

[6] Gemmell, J., Schimoler, T., Mobasher, B., and Burke, R. 2010. Hybrid tag recommendation for social annotation systems. In *Proc, 19th ACM Int. Conf. Information and Knowledge Management*, 829–838.

[7] Guan, Z., Bu, J., Mei, Q., Chen, C. and Wang, C. 2009. Personalized tag recommendation using graph-based ranking on multi-type interrelated objects. In *Proc. 32nd Int. ACM SIGIR Conf. Research and Development in Information Retrieval*, 540–547.

[8] Haveliwala, T. H. Topic-Sensitive PageRank: A context-sensitive ranking algorithm for Web search. *IEEE Trans. on Knowl. and Data Eng.* 15, 4, 784–796.

[9] Heymann, P., Ramage, D., and Garcia-Molina, H. 2008. Social tag prediction. *In Proc. 31st Int. ACM SIGIR Conf. Research and Development in Information Retrieval*, 531–538.

[10] Hotho, A., Jäschke, R., Schmitz, C., and Stumme, G. 2006. Information retrieval in folksonomies: Search and ranking. In *Proc. 3rd European Semantic Web Conf.*, 411–426.

[11] Jäschke, R., Marinho, L., Hotho, A., Schmidt-Thieme, L., and Stumme, G. 2008. Tag recommendations in social bookmarking systems. *AI Commun.* 21, 4, 231–247.

[12] Jeh, G. and Widom, J. 2003. Scaling personalized web search. In *Proc. 12th Int. Conf. World Wide Web*, 271-279.

[13] Langville, A.N. and Meyer, C.D. 2005. A Survey of eigenvector methods for Web information retrieval. *SIAM Rev.* 47, 1. 135–161.

[14] Manning, C.D., Raghavan, P., and Schütze, H. 2008. *Introduction to Information Retrieval*. Cambridge University Press.

[15] Rendle S. and Schmidt-Thieme, L. 2010. Pairwise interaction tensor factorization for personalized tag recommendation. *In Proc. 3rd ACM Int. Conf. Web Search and Data Mining*, 81–90.

[16] Song, Y., Zhuang, Z., Li, H., Zhao, Q., Li, J., Lee, W.-C., and Lee Giles, C. 2008. Real-time automatic tag recommendation. *In Proc. 31st Int. ACM SIGIR Conf. Research and Development in Information Retrieval*, 515–522.