Tag Recommendations in Social Bookmarking Systems

Robert Jäschke [a,c] Leandro Marinho [b,d]
Andreas Hotho [a] Lars Schmidt-Thieme [b]
Gerd Stumme [a,c]

[a] *Knowledge & Data Engineering Group (KDE), University of Kassel, Wilhelmshöher Allee 73, 34121 Kassel, Germany http://www.kde.cs.uni-kassel.de*
[b] *Information Systems and Machine Learning Lab (ISMLL), University of Hildesheim, Samelsonplatz 1, 31141 Hildesheim, Germany http://www.ismll.uni-hildesheim.de*
[c] *Research Center L3S, Appelstr. 9a, 30167 Hannover, Germany http://www.l3s.de*
[d] *Brazilian National Council Scientific and Technological Research (CNPq) scholarship holder.*

Abstract. Collaborative tagging systems allow users to assign keywords—so called "tags"—to resources. Tags are used for navigation, finding resources and serendipitous browsing and thus provide an immediate benefit for users. These systems usually include tag recommendation mechanisms easing the process of finding good tags for a resource, but also consolidating the tag vocabulary across users. In practice, however, only very basic recommendation strategies are applied.

In this paper we evaluate and compare several recommendation algorithms on large-scale real life datasets: an adaptation of user-based collaborative filtering, a graph-based recommender built on top of the FolkRank algorithm, and simple methods based on counting tag occurences. We show that both FolkRank and Collaborative Filtering provide better results than non-personalized baseline methods. Moreover, since methods based on counting tag occurrences are computationally cheap, and thus usually preferable for real time scenarios, we discuss simple approaches for improving the performance of such methods. We show, how a simple recommender based on counting tags from users and resources can perform almost as good as the best recommender.

Keywords: Folksonomies, Recommender Systems, Social Bookmarking, Ranking

1. Introduction

Folksonomies are web-based systems that allow users to upload their resources, and to label them with arbitrary words, so-called *tags*. The systems can be distinguished according to what kind of resources are supported. Flickr[1], for instance, allows the sharing of photos, del.icio.us[2] the sharing of bookmarks, CiteULike[3] and Connotea[4] the sharing of bibliographic references, and last.fm[5] the sharing of music listening habits. *BibSonomy*[6] allows to share bookmarks and BIBTEX based publication entries simultaneously.

In their core, these systems are all very similar. Once a user is logged in, he can add a resource to the system, and assign arbitrary tags to it. The collection of all his assignments is his *personomy*, the collection of all personomies constitutes the *folksonomy*. The user can explore his personomy, as well as the whole folksonomy, in all dimensions: for a given user one can see all resources he has uploaded, together with the tags he has assigned to them; when clicking on a resource one sees which other users have uploaded this resource and how they tagged it; and when clicking on a tag one sees who assigned it to which resources. Based on the tags that are assigned to a resource, users are able to search and find her own or other users resources within such systems.

To support users in the tagging process and to expose different facets of a resource, most of the systems offered some kind of tag recommendations already at an early stage. Del.icio.us, for instance, had a tag recommender in June 2005 at the latest,[7] and also included resource recommendations.[8] However, no algorithmic details were published. We assume that these recommendations basically provide those tags which were most frequently assigned to the resource (called *most popular tags by resource* in the sequel).

[1] http://flickr.com [2] http://del.icio.us
[3] http://www.citeulike.org [4] http://www.connotea.org
[5] http://www.last.fm [6] http://www.bibsonomy.org
[7] http://www.socio-kybernetics.net/saurierduval/archive/2005_06_01_archive.html [8] http://blog.del.icio.us/blog/2005/08/people_who_like.html

As of today, nobody has empirically shown the benefits of recommenders in such systems. In this paper, we will evaluate a tag recommender based on Collaborative Filtering (introduced in Section 3.1), a graph based recommender using our ranking algorithm FolkRank (see Section 3.2), and several simpler approaches based on tag counts (Section 3.3). In Section 4, we discuss the computational costs of the different algorithms. The quality of the resulting recommendations is evaluated on three real world folksonomy datasets from del.icio.us, BibSonomy[9] and last.fm (Sections 5 and 6). In the next section we start with recalling the basics and discussing related work.

The results presented in this article built upon results presented at the 18th European Conference on Machine Learning (ECML) / 11th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD) 2007 [17].

2. Recommending Tags—Problem Definition and State of the Art

Most recommender systems are typically used to call users' attentions to new objects they do not know yet and have not rated already in the past. This is often due to the fact that there is no repeat-buying in domains like books, movies, music etc. in which these systems typically operate. In social bookmarking systems, on the contrary, re-occurring tags are an essential feature for structuring the knowledge of a user or a group of users, and have to be considered by a tag recommender.

This means that the fact that a tag already has been used to annotate a resource does not exclude the possibility of recommending the same tag for a different resource of the same user. Overall, recommending tags can serve various purposes, such as: increasing the chances of getting a resource annotated, reminding a user what a resource is about and consolidating the vocabulary across the users.

In this section we formalize the notion of folksonomies, formulate the tag recommendation problem, and briefly describe the state of the art on tag recommendations in folksonomies.

[9] We make the BibSonomy dataset publicly available for research purposes to stimulate research in the area of folksonomy systems (details in Section 5.1).

2.1. A Formal Model for Folksonomies

Formally, a *folksonomy* is a tuple $\mathbb{F} := (U, T, R, Y)$ where

- U, T, and R are finite sets, whose elements are called *users*, *tags* and *resources*, resp., and
- Y is a ternary relation between them, i. e., $Y \subseteq U \times T \times R$, whose elements are called tag assignments (*tas* for short).[10]

Users are typically described by their user ID, and tags may be arbitrary strings. What is considered a resource depends on the type of system. For instance, in del.icio.us, the resources are URLs, in BibSonomy URLs or publication references, and in last.fm, the resources are artists.

For convenience we also define, for all $u \in U$ and $r \in R$, $T(u, r) := \{t \in T \mid (u, t, r) \in Y\}$, i. e., $T(u, r)$ is the set of all tags that user u has assigned to resource r. The set of all *posts* of the folksonomy is then $P := \{(u, S, r) \mid u \in U, r \in R, S = T(u, r), S \neq \emptyset\}$. Thus, each *post* consists of a user, a resource and all tags that this user has assigned to that resource.

2.2. Problem Definition

Recommender systems (RS) in general recommend interesting or personalized information objects to users based on explicit or implicit ratings. Usually RS predict ratings of objects or suggest a list of new objects that the user hopefully will like the most. The task of a tag recommender system is to recommend, for a given user $u \in U$ and a given resource $r \in R$ with $T(u, r) = \emptyset$, a set $\tilde{T}(u, r) \subseteq T$ of tags. In many cases, $\tilde{T}(u, r)$ is computed by first generating a ranking on the set of tags according to some quality or relevance criterion, from which then the top n elements are selected.

Notice that the notion of tag relevance in folksonomies can assume different perspectives, i. e., a tag can be judged relevant to a given resource according to the society point of view, through the opinion of experts in the domain or based on the personal profile of an individual user. For all the evaluated algorithms, we concentrate here on measuring the individual notion of tag relevance, i. e., the degree of likeliness of a user for a certain set of tags, given a new or untagged resource.

[10] In the original definition [15], we introduced additionally a subtag/supertag relation, which we omit here. The version used here is known in Formal Concept Analysis [11] as a *triadic context* [19,28].

2.3. Related Work

General overviews on the rather young area of folksonomy systems and their strengths and weaknesses are given in [13,20,22]. In [23], Mika defines a model of semantic-social networks for extracting lightweight ontologies from del.icio.us. Recently, work on more specialized topics such as structure mining on folksonomies—e. g. to visualize trends [9] and patterns [27] in users' tagging behavior—as well as ranking of folksonomy contents [15], analyzing the semiotic dynamics of the tagging vocabulary [7], or the dynamics and semantics [12] have been presented.

The literature concerning the problem of tag recommendations in folksonomies is still sparse. The existent approaches usually lay in the collaborative filtering and information retrieval areas. In [24,6], algorithms for tag recommendations are devised based on content-based filtering techniques. Xu et al. [32] introduce a collaborative tag suggestion approach based on the HITS algorithm [18]. A goodness measure for tags, derived from collective user authorities, is iteratively adjusted by a reward-penalty algorithm. Benz et al. [3] introduce a collaborative approach for bookmark classification based on a combination of nearest-neighbor-classifiers. There, a keyword recommender plays the role of a collaborative tag recommender, but it is just a component of the overall algorithm, and therefore there is no information about its effectiveness alone. Basile et al. [1] suggests an architecture of an intelligent recommender tag system. In [10,31,29] the problem of tag-aware resource recommendations is investigated. The standard tag recommenders, in practice, are services that provide the most-popular tags used for a particular resource. This is usually done by means of tag clouds where the most frequent used tags are depicted in a larger font or otherwise emphasized.

The approaches described above address important aspects of the problem, but they still diverge on the notion of tag relevance and evaluation protocol used. In [32,1], e. g., no quantitative evaluation is presented, while in [24], the notion of tag relevance is not entirely defined by the users but partially by experts. Furthermore, most of them make use of some content information which is specific to the particular type of resource of the system. It is certainly interesting to exploit content information, but since folksonomies can support different types of resources, e.g. audio, image, text, or video, one would need to write specific recommenders suited for each distinct content type. In this paper we are particulary interested in generic algorithms that can be applied to folksonomies disregarding the domain and kind of resource supported.

Most recently, the ECML PKDD 2008 Discovery Challenge[11] has addressed the problem of tag recommendations in folksonomies.

3. Recommendation Algorithms

In this section we present three classes of recommendation algorithms we will evaluate in the following sections: a straight-forward adaptation of Collaborative Filtering [4,25] based on user-tag and user-resource projections, two adaptations of PageRank [5] for folksonomies, and various methods based on counting the most popular tags.

3.1. Collaborative Filtering

Due to its simplicity and promising results, Collaborative Filtering (CF) has been one of the most dominant methods used in recommender systems. In the next section we recall the basic principles and then present the details of the adaptation to folksonomies.

3.1.1. Basic Collaborative Filtering Principle

The idea is to suggest new objects or to predict the utility of a certain object based on the opinion of like-minded users [26]. In CF, for m users and n objects, the user profiles are represented in a user-object matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$. The matrix can be decomposed into row vectors:

$$\mathbf{X} := [\vec{x}_1, ..., \vec{x}_m]^T \text{ with } \vec{x}_u := [x_{u,1}, ..., x_{u,n}], \text{ for } u := 1, \ldots, m,$$

where $x_{u,o}$ indicates that user u rated object o by $x_{u,o} \in \mathbb{R}$. Each row vector $\vec{x}_u$ corresponds thus to a user profile representing the object ratings of a particular user. This decomposition leads to user-based CF—in contrast to item-based algorithms (see [8]).[12]

Now, one can compute, for a given user u, the recommendation as follows. First, based on the matrix $\mathbf{X}$ and for a given k, the set N_u^k of the k users that are most similar to user $u \in U$ are computed:

$$N_u^k := \underset{v \in U \setminus \{u\}}{\overset{k}{\operatorname{argmax}}} \operatorname{sim}(\vec{x}_u, \vec{x}_v)$$

[11] http://www.kde.cs.uni-kassel.de/ws/rsdc08/

[12] We also measured the performance of item-based CF. Since precision and recall of its recommendations were for all datasets worse than those of user-based CF, we decided to present the later type of CF as the baseline for CF algorithms.

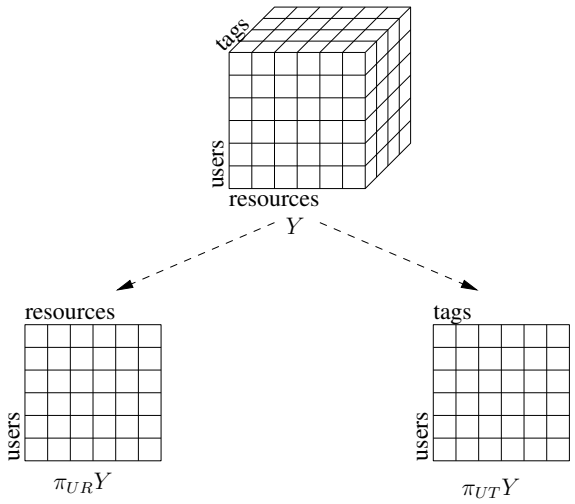

Fig. 1. Projections of Y into the user's resource and user's tag spaces.

where the superscript in the argmax function indicates the number k of neighbors to be returned, and sim is regarded (in our setting) as the cosine similarity measure, i. e., $\mathrm{sim}(\vec{x}_u, \vec{x}_v) := \frac{\langle \vec{x}_u, \vec{x}_v \rangle}{\|\vec{x}_u\| \|\vec{x}_v\|}$.

Then, for a given $n \in \mathbb{N}$, the top n recommendations consist of a list of objects ranked by decreasing frequency of occurrence in the ratings of the neighbors (see Eq. 1 below for the folksonomy case).

3.1.2. Collaborative Filtering for Recommending Tags in Folksonomies

Because of the ternary relational nature of folksonomies, traditional CF cannot be applied directly, unless we reduce the ternary relation Y to a lower dimensional space [21]. To this end we consider as matrix $\mathbf{X}$ alternatively the two 2-dimensional projections $\pi_{UR}Y \in \{0,1\}^{|U|\times|R|}$ with $(\pi_{UR}Y)_{u,r} := 1$ if there exists $t \in T$ s. t. $(u,t,r) \in Y$ and 0 else, and $\pi_{UT}Y \in \{0,1\}^{|U|\times|T|}$ with $(\pi_{UT}Y)_{u,t} := 1$ if there exists $r \in R$ s. t. $(u,t,r) \in Y$ and 0 else (Figure 1).

The projections preserve the user information, and lead to recommender systems based on occurrence or non-occurrence of resources or tags, resp., with the users. This approach is similar to recommenders that are based on web log data. Notice that here we have two possible setups in which the k-neighborhood N_u^k of a user u can be formed, by considering either the resources or the tags as objects.

Having defined matrix $\mathbf{X}$, and having decided whether to use $\pi_{UR}Y$ or $\pi_{UT}Y$ for computing user neighborhoods, we have the required setup to apply Collaborative Filtering. For determining, for a given user u, a given resource r, and some $n \in \mathbb{N}$, the set $\tilde{T}(u,r)$ of n recommended tags, we compute first N_u^k as described above, followed by:

$$\tilde{T}(u,r) := \operatorname*{argmax}_{t \in T}^{n} \sum_{v \in N_u^k} \mathrm{sim}(\vec{x}_u, \vec{x}_v)\delta(v,t,r) \quad (1)$$

where $\delta(v,t,r) := 1$ if $(v,t,r) \in Y$ and 0 else.

3.2. A Graph-Based Approach

The web search algorithm PageRank [5] reflects the idea that a web page is important if there are many pages linking to it, and if those pages are important themselves.[13] In [15], we employed the same underlying principle for Google-like search and ranking in folksonomies. The key idea of our FolkRank algorithm is that a resource which is tagged with important tags by important users becomes important itself. The same holds, symmetrically, for tags and users. We have thus a graph of vertices which are mutually reinforcing each other by spreading their weights. In this section we briefly recall the principles of the FolkRank algorithm, and explain how we use it for generating tag recommendations.

Because of the different nature of folksonomies compared to the web graph (undirected triadic hyperedges instead of directed binary edges), PageRank cannot be applied directly on folksonomies. In order to employ a weight-spreading ranking scheme on folksonomies, we overcome this problem in two steps. First, we transform the hypergraph into an undirected graph. Then we apply a differential ranking approach that deals with the skewed structure of the network and the undirectedness of folksonomies, and which allows for topic-specific rankings.

3.2.1. Folksonomy-Adapted PageRank

First we convert the folksonomy $\mathbb{F} = (U,T,R,Y)$ into an *un*directed tri-partite graph $G_{\mathbb{F}} = (V,E)$. The set V of nodes of the graph consists of the disjoint union of the sets of tags, users and resources (i. e., $V = U \dot{\cup} T \dot{\cup} R$). All co-occurrences of tags and users, users and resources, tags and resources become edges between the respective nodes. I. e., each triple (u,t,r) in Y gives rise to the three undirected edges $\{u,t\}$, $\{u,r\}$, and $\{t,r\}$ in E.

Like PageRank, we employ the random surfer model, that is based on the idea that an idealized random web surfer normally follows links (e. g., from a resource

[13] This idea was extended in a similar fashion to bipartite subgraphs of the web in HITS [18] and to n-ary directed graphs in [30].

page to a tag or a user page), but from time to time jumps to a new node without following a link. This results in the following definition.

The rank of the vertices of the graph is computed with the weight spreading computation

$$\vec{w}_{t+1} \leftarrow dA^T \vec{w}_t + (1-d)\vec{p} \ , \qquad (2)$$

where $\vec{w}$ is a weight vector with one entry for each node in V, A is the row-stochastic version of the adjacency matrix[14] of the graph $G_{\mathbb{F}}$ defined above, $\vec{p}$ is the random surfer vector—which we use as preference vector in our setting, and $d \in [0,1]$ is determining the strength of the influence of $\vec{p}$. By normalization of the vector $\vec{p}$, we enforce the equality $||\vec{w}||_1 = ||\vec{p}||_1$. This[15] ensures that the weight in the system will remain constant. The rank of each node is its value in the limit $\vec{w} := \lim_{t\to\infty} \vec{w}_t$ of the iteration process.

For a global ranking, one will choose $\vec{p} = \mathbf{1}$, i. e., the vector composed by 1's. In order to generate recommendations, however, $\vec{p}$ can be tuned by giving a higher weight to the user node and to the resource node for which one currently wants to generate a recommendation. The recommendation $\tilde{T}(u,r)$ is then the set of the top n nodes in the ranking, restricted to tags.

As the graph $G_{\mathbb{F}}$ is undirected, most of the weight that went through an edge at moment t will flow back at $t+1$. The results are thus rather similar (but not identical, due to the random surfer) to a ranking that is simply based on edge degrees. In the experiments presented below, we will see that this version performs reasonable, but not exceptional. This is in line with our observation in [15] which showed that the topic-specific rankings are biased by the global graph structure. As a consequence, we developed in [15] the following differential approach.

3.2.2. FolkRank—Topic-Specific Ranking

The undirectedness of graph $G_{\mathbb{F}}$ makes it very difficult for other nodes than those with high edge degree to become highly ranked, no matter what the preference vector is.

This problem is solved by the *differential* approach in FolkRank, which computes a topic-specific ranking of the elements in a folksonomy. In our case, the topic is determined by the user/resource pair (u,r) for which we intend to compute the tag recommendation.

1. Let $\vec{w}^{(0)}$ be the fixed point from Equation (2) with $\vec{p} = \mathbf{1}$.
2. Let $\vec{w}^{(1)}$ be the fixed point from Equation (2) with $\vec{p} = \mathbf{1}$, but $\vec{p}[u] = 1+|U|$ and $\vec{p}[r] = 1+|R|$.
3. $\vec{w} := \vec{w}^{(1)} - \vec{w}^{(0)}$ is the final weight vector.

Thus, we compute the winners and losers of the mutual reinforcement of nodes when a user/resource pair is given, compared to the baseline without a preference vector. We call the resulting weight $\vec{w}[x]$ of an element x of the folksonomy the *FolkRank* of x.[16]

For generating a tag recommendation for a given user/resource pair (u,r), we compute the ranking as described and then restrict the result set $\tilde{T}(u,r)$ to the top n tag nodes.

3.3. Most Popular Tags

In this section we introduce methods based on tag counts. In the sequel we will see that these methods are particulary cheap to compute and therefore might be good candidates for online computation of recommendations.

For convenience, we define, for a user $u \in U$, the set of all his tag assignments $Y_u := Y \cap (\{u\} \times T \times R)$. The sets Y_r (for any resource $r \in R$) and Y_t (for any tag $t \in T$) are defined accordingly. Similiarly, we define, for $t \in T$ and $r \in R$, $Y_{t,u} := Y \cap (\{u\} \times \{t\} \times R)$; and define $Y_{t,r}$ accordingly. Finally, we define, for a user $u \in U$, the set of all his tags $T_u := \{t \in T \mid \exists r \in R\colon (u,t,r) \in Y\}$. The set T_r (for any resource $r \in R$) is defined accordingly.

3.3.1. Variants of "Most Popular Tags"

1. Recommending the *most popular tags* of the folksonomy is the most simplistic approach. It recommends, for any user $u \in U$ and any resource $r \in R$, the same set:

$$\tilde{T}(u,r) := \operatorname*{argmax}^{n}_{t\in T}(|Y_t|).$$

   This approach suffers only minimally from cold-start problems.
2. Tags that globally are most specific to the resource will be recommended when using the *most popular tags by resource*:

$$\tilde{T}(u,r) := \operatorname*{argmax}^{n}_{t\in T}(|Y_{t,r}|) \ .$$

[14] $a_{ij} := \frac{1}{\deg(i)}$ if $\{i,j\} \in E$ and 0 else [15] ...together with the condition that there are no rank sinks—which holds trivially in the undirected graph $G_{\mathbb{F}}$.

[16] In [15] we showed that $\vec{w}$ provides indeed valuable results on a large-scale real-world dataset while $\vec{w}^{(1)}$ provides an unstructured mix of topic-relevant elements with elements having high edge degree. In [16], we applied this approach for detecting trends over time in folksonomies.

3. Since users might have specific interests for which they already tagged several resources, using the *most popular tags by user* is another option:
$$\tilde{T}(u,r) := \operatorname*{argmax}^{n}_{t \in T}(|Y_{t,u}|) \ .$$

As we will later see (cf. Sec. 6), none of the aforementioned methods alone will in general provide the best recommendations. Nevertheless, the simplicity and cost efficiency of algorithms based on tag counts make them a favored approach for use in existing folksonomy systems. Therefore, we experimented with a *mix* of the recommendations generated by variants 2 and 3 which we call *most popular tags mix* in the following sections.

3.3.2. Mix of "Most Popular Tags" Recommenders

The main idea of this approach is to recommend a mix of the most popular tags of the user with the most popular tags of the resource. The simplest way to mix the tags is to add their counts and then sort them by their count:

$$\tilde{T}(u,r) := \operatorname*{argmax}^{n}_{t \in T}(|Y_{t,r}| + |Y_{t,u}|) \ .$$

This way of mixing will be called *most popular tags mix 1:1*, since we just add the counts as they are. For instance, if the resource has been tagged four times with *web* by other users and the user has used the tag *web* six times on other resources, the tag *web* would get a count of ten.

Although this method already yields good results (as we will show in the results section 6), the influence of the user-based recommendation will be very small compared to the resource-based recommendation if many people have tagged this resource. Vice versa, if a user has tagged many resources, his most popular tags might have counts that are much higher than the counts provided by the resources. Hence, we introduced another mix variant, where the tag counts of the two participating sets are *normalized* and *weighted* before they are added. We define as normalization function, for each tag $t \in T_r$:

$$\mathrm{norm}_r(t) := \frac{|Y_{t,r}| - \min_{t' \in T} |Y_{t',r}|}{\max_{t' \in T} |Y_{t',r}| - \min_{t' \in T} |Y_{t',r}|} \ . \tag{3}$$

For $t \in T_u$, the normalisation $\mathrm{norm}_u(t)$ is defined in an analogue fashion. After normalization the weights of all tags in T_r and T_u lie between zero and one—with the most popular tag(s) having weight 1 and the least important tag(s) having weight 0. A pre-defined factor $\rho \in [0,1]$ allows us to balance the influence of the user and the resource:

$$\tilde{T}(u,r) := \operatorname*{argmax}^{n}_{t \in T}(\rho\,\mathrm{norm}_r(t) + (1-\rho)\,\mathrm{norm}_u(t)) \ .$$

We call this method the *most popular tags ρ–mix*.

Note that the *most popular tags* 0*–mix* is just the *most popular tags by user* strategy, since the normalization does not change the order of the tags. Similarly, the *most popular tags* 1*–mix* is just the *most popular tags by resource* strategy. Note, however, that due to normalization the *most popular tags* 0.5*–mix* is not identical to the *most popular tags mix 1:1*!

In Section 6 we will analyze how well different values of ρ perform and find the best value for the examined datasets.

4. Computational Costs

In an online scenario, where tag recommendations should be given to the user while he tags a resource, one must consider the computational costs of the used algorithm. Hence, in this section we want to discuss briefly the costs of the algorithms proposed so far. We will see that the methods described in the preceding section are especially cheap to compute and therefore might be good candidates for real-time computation of recommendations, if they can provide useful recommendations. Here we want to estimate the complexity of recommending n tags for a given user-resource tuple (u,r) using the proposed solutions.

4.1. Collaborative Filtering

The computational complexity of the CF algorithm depends on three steps:

1. Computation of projections: In order to compose the projections, we need to determine only the resources' and tags' co-occurrences with the set of users $V \subseteq U$ that have tagged the active resource $r \in R$. For that, we need to do a linear scan in Y resulting in a complexity of $\mathcal{O}(|Y|)$. However, with appropriate index structures, which allow to access the tag assignments of u (or r) efficiently, this reduces to $\mathcal{O}(\log(|R|) + |Y_u||V|\log(|U|))$.

2. Neighborhood formation: In traditional user-based CF algorithms, the computation of the neighborhood N_u is usually linear on the number of users as one needs to compute the similarity of the active user with all the other users in the database. However, in CF-based tag recommendations we are only interested in the subset V of users that tagged the active resource. Thus, the upper bound on the complexity of this step would be $\mathcal{O}(|V|Z)$, as we need to compute $|V|$ similarities each requiring Z operations. In the worst case $|V| = |U|$ but this rarely occurs in practice. In addition, we need to sort the similarities to compute the N_u nearest users. Therefore the complexity of this step is $\mathcal{O}(|V|(Z+\log(|N_u|)))$.
3. Recommendations: In order to compute the top-n recommendations for a given (u, r) pair, we need to: (i) count the tag occurrences of nearest users N_u similarities (see Eq. 1), and (ii) sort the tags based on their weight, which results in a complexity of $\mathcal{O}(|Y_u||N_u|\log(n))$.

Hence, the whole complexity given the three steps above is $\mathcal{O}(\log(|R|)+|Y_u||V|+|V|(Z+\log(|N_u|))+|Y_u||N_u|\log(n))$ and can be simplified to $\mathcal{O}(|V|(2|Y_u|+\log(|V|)+|Y_u|\log(|n|)) \subseteq \mathcal{O}(|V||Y_u|)$ since $|N_u| \leq |V|$ and $Z \leq |Y_u|$.

4.2. The Graph-Based Approach

One iteration of the adapted PageRank requires the computation of $dA^T\vec{w} + (d-1)\vec{p}$, with $A \in \mathbb{R}^{s\times s}$ where $s := |U| + |T| + |R|$. If t marks the number of iterations, the complexity would therefore be $(s^2+s)t \in \mathcal{O}(s^2t)$. However, since A is sparse, it is more efficient to go linearly over all *tag assignments* in Y to compute the product $A^T\vec{w}$. Together with the costs of adding the preference vector $\vec{p} \in \mathbb{R}^s$ this results in a complexity of $\mathcal{O}((|Y|+s)t)$. After rank computation we have to sort the weights of the tags to collect the top n tags, thus the final complexity of the adapted PageRank for top-n tag recommendation is $\mathcal{O}((|Y|+s)t + |T|\log(n))$.

For FolkRank, one has to compute the baseline $\vec{w}^{(0)}$ once (and update it on a regular basis)—hence, these costs do not really add up to the costs for computing one recommendation. However, the baseline $\vec{w}^{(0)}$ has to be subtracted from $\vec{w}^{(1)}$, which costs at most $|T|$ iterations (since we are only interested in the weights of the tags). Thus, the costs of FolkRank are $\mathcal{O}((|Y| + s)t + |T|\log(n) + |T|)$, which can be simplified to $\mathcal{O}((|Y|+s)t)$, since $|T|$ is small compared to $|Y|$.

4.3. Most Popular Tags

If we want to compute, for a given pair (u, r), the most popular tags of the user u (or the resource r), we need to linearly scan Y to calculate the occurrence counts for u's tags (or r's tags) and afterwards sort the tags we gathered by their count. This would result in a complexity of $\mathcal{O}(|Y| + |T_u|\log(n))$ (or $\mathcal{O}(|Y| + |T_r|\log(n))$). Nevertheless (as for CF), with efficient index structures to access T_u (or T_r) this reduces to $\mathcal{O}(\log(|U|) + |Y_u| + |T_u|\log(n))$ (or $\mathcal{O}(\log(|R|) + |Y_r| + |T_r|\log(n))$).

For the *most popular tags mixes* we have to consider both of the costs and additionally add the costs to normalize the tags, which includes finding the tags with the highest and lowest counts. This results in a complexity of $\mathcal{O}(\log(|U|)+|Y_u|+\log(|R|)+|Y_r|+|T_u|+|T_r|+(|T_u|+|T_r|)\log(n))$. With $|T_u| \leq |Y_u|$ the costs are at most $\mathcal{O}(4|Y_u| + 2|Y_u|\log(n)) \subseteq \mathcal{O}(|Y_u|)$

4.4. Comparison

Since Y_u is only a small part of Y, *CF* and the most popular methods are much cheaper to compute than *FolkRank*, which in each iteration has to scan Y. Additionally, both methods don't need any iteration. Comparing *CF* and the *most popular mixes* requires to estimate the size of the set V of users, which have tagged a particular resource. This certainly depends on the resource at hand, but on average the factor $|V|$ of the *CF* costs will be larger than the constant factors of $|Y_u|$ in the *most popular mix* costs. In general, both methods have similiar costs with some advantage on the side of the mixes.

5. Evaluation Procedure

In order to evaluate the quality of the recommendations of the different algorithms, we have run experiments on three real-world datasets. In this section we first describe the datasets we used, how we prepared the data, the methodology deployed to measure the performance, and which algorithms we used, together with their specific settings. The results will be discussed in Section 6.

5.1. Datasets

To evaluate the proposed recommendation techniques we have chosen datasets from three different folksonomy systems: *del.icio.us, BibSonomy* and *last.fm*. They have different sizes, different resources, and are probably used by different people. Therefore we assume that our observations will also be significant for other social bookmarking systems. Table 1 gives an overview on the datasets. For all datasets we disregarded if the tags had lower or upper case, since this is the behaviour of most systems when querying them for posts tagged with a certain tag (although often they store the tags as entered by the user).

Del.icio.us. One of the first and most popular folksonomy systems is del.icio.us[17] which exists since the end of 2003. It allows users to tag bookmarks (URLs) and had according to its blog around 1.5 Mio. users in February 2007. We used a dataset from del.icio.us we obtained from July 27 to 30, 2005 [15].

BibSonomy. This system allows users to manage and annotate bookmarks and publication references simultanously. Since three of the authors have participated in the development of BibSonomy,[18] we were able to create a complete snapshot of all users, resources (both publication references and bookmarks) and tags publicly available at April 30, 2007, 23:59:59 CEST.[19] From the snapshot we excluded the posts from the DBLP computer science bibliography[20] since they are automatically inserted and all owned by one user and all tagged with the same tag (*dblp*). Therefore they do not provide meaningful information for the analysis.

Last.fm. Audioscrobbler[21] is a "database that tracks listening habits". The user profiles are built through the use of the company's flagship product, last.fm,[22] a system that provides personalized radio stations for its users and updates their profiles using the music they listen to. Audioscrobbler exposes large portions of data through their web services API. The data were gathered during July 2006, partly through the web services API (collecting user nicknames), partly crawling the last.fm site. Here the resources are artist names, whose spellings are already normalized by the system.

[17] http://del.icio.us [18] http://www.bibsonomy.org
[19] On request to bibsonomy@cs.uni-kassel.de a snapshot of BibSonomy is available for research purposes. [20] http://www.informatik.uni-trier.de/~ley/db/
[21] http://www.audioscrobbler.net [22] http://www.last.fm

5.2. Core Computation

Many recommendation algorithms suffer from sparse data and will thus produce bad recommendations on the "long tail" of items which were used by only few users. We follow the conventional approach and restrict the evaluation to the "dense" part of the folksonomy. To this end, we adapt the notion of a p-core [2] to tripartite hypergraphs. The p-core of level k is a subset of the folksonomy with the property, that *each user, tag and resource has/occurs in at least k posts*. For the del.icio.us dataset we will later see that using the core will (except for the adapted PageRank) not change the relative performance differences of the algorithms.

To construct the p-core, recall that a folksonomy (U, T, R, Y) can be formalized equivalently as undirected tri-partite hypergraph $G = (V, E)$ with $V = U \dot\cup T \dot\cup R$ and $E = \{\{u, t, r\} \mid (u, t, r) \in Y\}$. First we define, for a subset V' of V (with $V' = U' \dot\cup T' \dot\cup R'$ and $U' \subseteq U, T' \subseteq T, R' \subseteq R$), the function

$$\mathrm{posts}(v, V') = \begin{cases} \{(v, S, r) \mid r \in R', S = T_{V'}(v, r)\} & \text{if} \quad v \in U' \\ \{(u, v, r) \mid u \in U', r \in R'\} & \text{if} \quad v \in T' \\ \{(u, S, v) \mid u \in U', S = T_{V'}(u, v)\} & \text{if} \quad v \in R' \end{cases} \tag{4}$$

which assigns to each $v \in V'$ the set of all posts in which v occurs. Here, $T_{V'}(u, r)$ is defined as in Section 2.1, but restricted to the subgraph (V', E'), with E' containing all edges from E whose nodes are contained in V'. Let $p(v, V') := |\mathrm{posts}(v, V')|$. The p-core at level $k \in \mathbb{N}$ is then the subgraph of (V, E) induced by V', where V' is a maximal subset of V such that, for all $v \in V'$, $p(v, V') \geq k$ holds.

Since $p(v, V')$ is, for all v, a monotone function in V, the p-core at any level k is unique [2], and we can use the algorithm presented in [2] for its computation.

An overview on the p-cores we used for our datasets is given in Table 2. For BibSonomy, we used $k = 5$ instead of 10 because of its smaller size. The largest k for which a p-core exists is listed, for each dataset, in the last column of Table 1.

Although the p-core as defined above breaks the symmetry of the hypergraph structure (contrary to tags, for users and resources the p-degree is not the same as the natural degree in the graph) it is the natural definition for our recommender scenario. We have also performed the evaluation on the symmetric variant of p (with lines 1 and 3 in Equation 4 modified similar to line 2), with rather similar results.

Table 1
Characteristics of the used datasets.

dataset	$\|U\|$	$\|T\|$	$\|R\|$	$\|Y\|$	$\|P\|$	date	$k_{\max}$
del.icio.us	75,245	456,697	3,158,435	17,780,260	7,698,653	2005-07-30	77
BibSonomy	1,037	28,648	86,563	341,183	96,972	2007-04-30	7
last.fm	3,746	10,848	5,197	299,520	100,101	2006-07-01	20

Table 2
Characteristics of the p-cores at level k.

dataset	k	$\|U\|$	$\|T\|$	$\|R\|$	$\|Y\|$	$\|P\|$
del.icio.us	10	37,399	22,170	74,874	7,487,319	3,055,436
BibSonomy	5	116	412	361	10,148	2,522
last.fm	10	2,917	2,045	1,853	219,702	75,565

5.3. Evaluation Methodology

5.3.1. Evaluation Measures

To evaluate the recommenders we used a variant of the leave-one-out hold-out estimation [14] which we call *LeavePostOut*. In all datasets, we picked randomly, for each user u, one resource r_u, which he had posted before. The task of the recommenders was then to predict the tags the user assigned to r_u, based on the folksonomy (U, T, R, Y') with $Y' := Y \setminus (\{u\} \times T(u, r_u) \times \{r_u\})$.

As performance measures we use precision and recall which are standard in such scenarios [14]. For (U, T, R, Y'), u, and r_u as defined above, precision and recall of a recommendation $\tilde{T}(u, r_u)$ are defined as follows

$$\text{recall}(\tilde{T}(u, r_u)) = \frac{|T(u, r_u) \cap \tilde{T}(u, r_u)|}{|T(u, r_u)|} \quad (5)$$

$$\text{precision}(\tilde{T}(u, r_u)) = \frac{|T(u, r_u) \cap \tilde{T}(u, r_u)|}{|\tilde{T}(u, r_u)|} \ . \quad (6)$$

For each dataset, we averaged these values over all its users:

$$\text{recall} = \frac{1}{|U|} \sum_{u \in U} \text{recall}(\tilde{T}(u, r_u)) \quad (7)$$

$$\text{precision} = \frac{1}{|U|} \sum_{u \in U} \text{precision}(\tilde{T}(u, r_u)) \ . \quad (8)$$

This process was repeated ten times for each dataset, each time with another resource per user, to further minimize the variance. In the sequel, the listed recall and precision values are thus always the averages over all ten runs.

5.3.2. Settings of the Algorithms

For each of the algorithms of our evaluation, we will now describe briefly the specific settings used to run it.

Collaborative Filtering UT. For this Collaborative Filtering variant the neighborhood is computed based on the user-tag matrix $\pi_{UT}Y$. The only parameter to be tuned in the CF based algorithms is the number k of nearest neighbors. For that, multiple runs were performed where k was successively incremented in steps of 10 until a point where no more improvements in the results were observed. The best values for k were 80 for del.icio.us, 20 for BibSonomy and 60 for the last.fm dataset.

Collaborative Filtering UR. Here the neighborhood is computed based on the user-resource matrix $\pi_{UR}Y$. For this approach the best values for k were 100 for del.icio.us, 30 for BibSonomy and 100 for the last.fm dataset.

Adapted PageRank. With the parameter $d = 0.7$ we stopped computation after 10 iterations. In $\vec{p}$, we gave higher weights to the user u and the resource r_u at hand: While each user, tag and resource got a preference weight of 1, u and r_u got a preference weight of $1 + |U|$ and $1 + |R|$, resp.

FolkRank. The same parameters and preference weights were used as in the adapted PageRank.

Most Popular Tags / Most Popular Tags by Resource / Most Popular Tags by User. These three approaches have no parameters. They were applied as described in Section 3.3.

Most Popular Tags ρ–Mix. We computed these recommendations for all $\rho \in \{0, 0.1, \ldots, 0.9, 1\}$. We will show in Section 6.1.1 that $\rho = 0.6$ is the most suit-

able of these values (at least on del.icio.us and BibSonomy), so that the comparison with the other algorithms will be done with this setting only.

It is important to notice that not all algorithms necessarily have maximal coverage, i. e., can always recommend n tags. Since *FolkRank* and *most popular tags* are the only algorithms with maximal coverage, the evaluation can be perturbated if the other algorithms cannot fill the list up to the given n. In this sense, whenever the recommendation list of an algorithm is not filled up to n, we complete the remaining entries with tags taken from the *most popular tags* that are not already in the list.

6. Results

In this section we present and describe the results of the evaluation. We will see that all three datasets show the same overall behavior: *most popular tags* is outperformed by all other approaches; the *CF-UT* algorithm performs slightly better than and the *CF-UR* approach approximately as good as the *most popular tags by resource*, and *FolkRank* uniformly provides significantly better results. The results for *most popular tags by user* and the *most popular tags 0.6–mix* are different among the datasets, however. We will further elaborate on this later.

There are two types of diagrams. The first type of diagram (e. g., Figure 3) shows in a straightforward manner how the recall depends on the number of recommended tags. The other diagrams are usual precision-recall plots. Here a datapoint on a curve stands for the number of tags recommended (starting with the highest ranked tag on the left of the curve and ending with ten tags on the right). Hence, the steady decrease of all curves in those plots means that the more tags of the recommendation are regarded, the better the recall and the worse the precision will be.

Since we averaged for each dataset over ten runs, we added error bars showing the standard deviation to the plots. However, except for the BibSonomy dataset, the standard deviation is so small that the error bars are mostly hidden by the datapoint symbols.

6.1. Del.icio.us

Due to the fact that the dataset from del.icio.us is by far the largest of the three we considered, we will discuss the results in more detail.

6.1.1. Determining ρ for Most Popular ρ–Mix

Before comparing the different algorithms described in the previous sections, we focus on finding an appropriate ρ for the *most popular ρ–mix* recommender on the del.icio.us p-core at level 10. Therefore, we varied ρ in 0.1-steps from 0 to 1 and plotted the resulting precision and recall; for comparison purposes we also added the plot of the *most popular mix 1:1* recommender.

As can be seen in Figure 2, the *most popular tags by user* ($\rho = 0$) recommender performs worse than the *most popular tags by resource* ($\rho = 1$) recommender for all numbers of recommended tags. All mixed versions perform better than *most popular tags by user* and all mixed versions with $\rho \geq 0.5$ perform better than *most popular tags by resource*. The best performance is obtained for $\rho = 0.6$ for the top three recommendations and $\rho = 0.7$ for more than three recommendations. We conclude, that the tags which other users used for that resource are better recommendations than the most popular tags of the user. Nevertheless, adding a small amount of popular tags of the user to the tags from the resource increases both precision and recall.

We observed a similar precision/recall behaviour for the different values of ρ on the non-pruned del.icio.us data as well as on the BibSonomy dataset. (The results are not shown here because of space restrictions.) For the following evaluations we decided therefore to include the results of the *most popular tags* 0.6*–mix* recommendations only, since for the top recommendations they have the best recall and precision and for more tags are still very close to the best results.

6.1.2. Comparison of Algorithms on p-core at Level 10

Figure 3 shows how the recall increases, when more tags of the recommendation are used. All algorithms perform significantly better than the baseline *most popular tags* and the *most popular tags by user* strategy—whereas it is much harder to beat the *most popular tags by resource*. The most apparent result is that the graph based *FolkRank* recommendations have superior recall—independent of the number of regarded tags. The top 10 tags given by *FolkRank* contained on average 80 % of the tags the users decided to attach to the selected resource. The second best results come from the *most popular tags* 0.6*–mix*, followed by the *Collaborative Filtering* approach based on user's tag similiarities.

The idea to suggest the *most popular tags by resource* results in a recall which is very similiar to

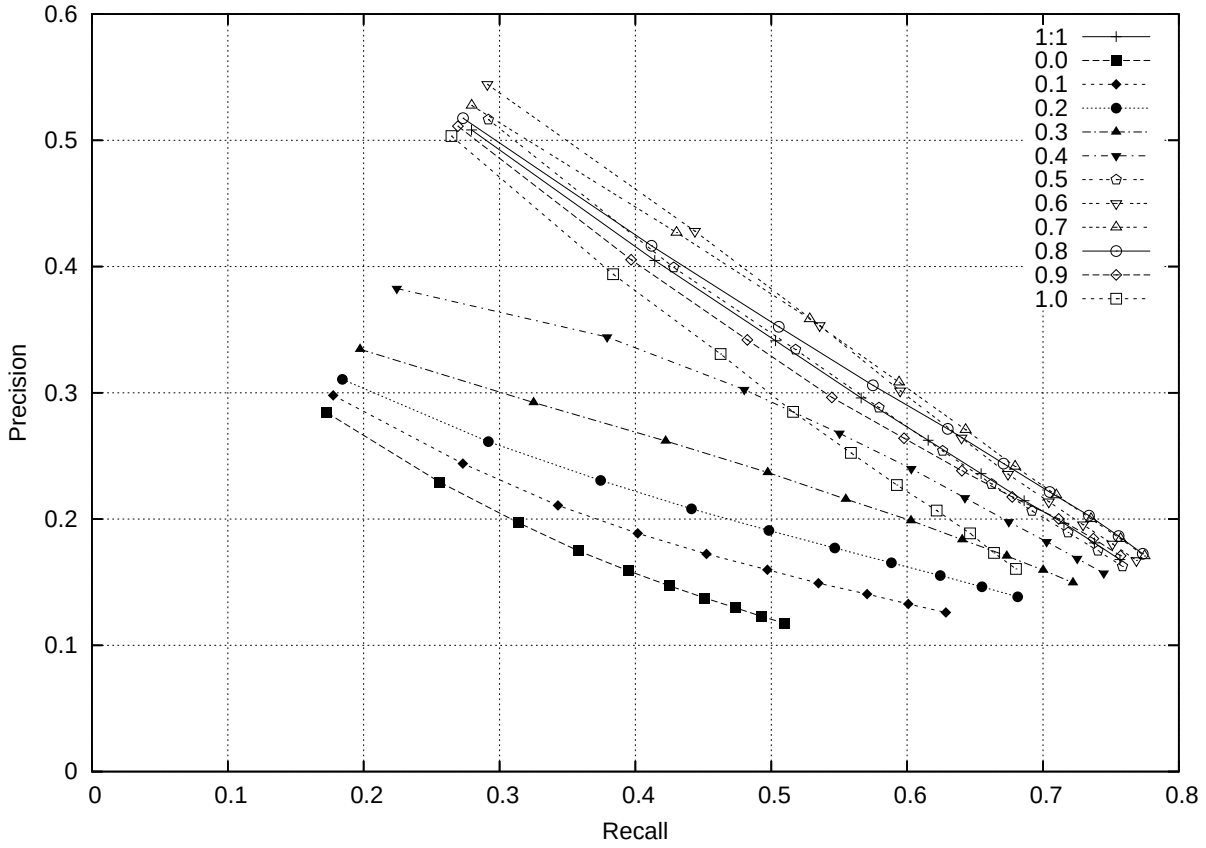

Fig. 2. Precision and recall of *most popular tags mix* $1:1$ and *most popular tags* ρ*–mix* for $\rho \in \{0, 0.1, \ldots, 0.9, 1\}$ on the del.icio.us p-core at level 10.

using the *CF* recommender based on user's resource similiarities—both perform worse than the aforementioned approaches. Between *most popular tags by resource* and *most popular tags* are the *adapted PageRank* which is biased towards the high degree nodes, as discussed in Section 3.2.1, and the *most popular tags by user* recommendations, which again perform not so well.

The precision-recall plot in Figure 4 extends Figure 3 with the precision measure. It again reveals clearly the quality of the recommendations given by *FolkRank* compared to the other approaches. Its precision values are systematically above those of the other approaches. For its top recommendations, FolkRank reaches precisions of 58.7 %.

A post in del.icio.us contains only 2.45 tags on average. A precision of 100 % can therefore not be reached when recommending ten tags. This justifies the poor precision of less than 20 % for all approaches when recommending ten tags. However, from a subjective point of view, the additional 'wrong' tags may even be considered as highly relevant, as the following example shows, where the user *tnash* has tagged the page http://www.ariadne.ac.uk/issue43/chudnov/ with the tags *semantic, web,* and *webdesign*. Since that page discusses the interaction of publication reference management systems in the web by the OpenURL standard, the tags recommended by *FolkRank* (*openurl, web, webdesign, libraries, search, semantic, metadata, social-software, sfx, seo*) are adequate and capture not only the user's point of view that this is a webdesign related issue in the semantic web, but also provide him with more specific tags like *libraries* or *metadata*. The *CF* based on user's tag similiarities recommends very similiar tags (*openurl, libraries, social-software, sfx, metadata, me/toread, software, myndsi, work, 2read*). The additional tags may thus animate users to use more tags and/or tags from a different viewpoint for describing resources, and thus lead to converging vocabularies.

The essential point in this example is, however, that *FolkRank* is able to predict—additionally to globally

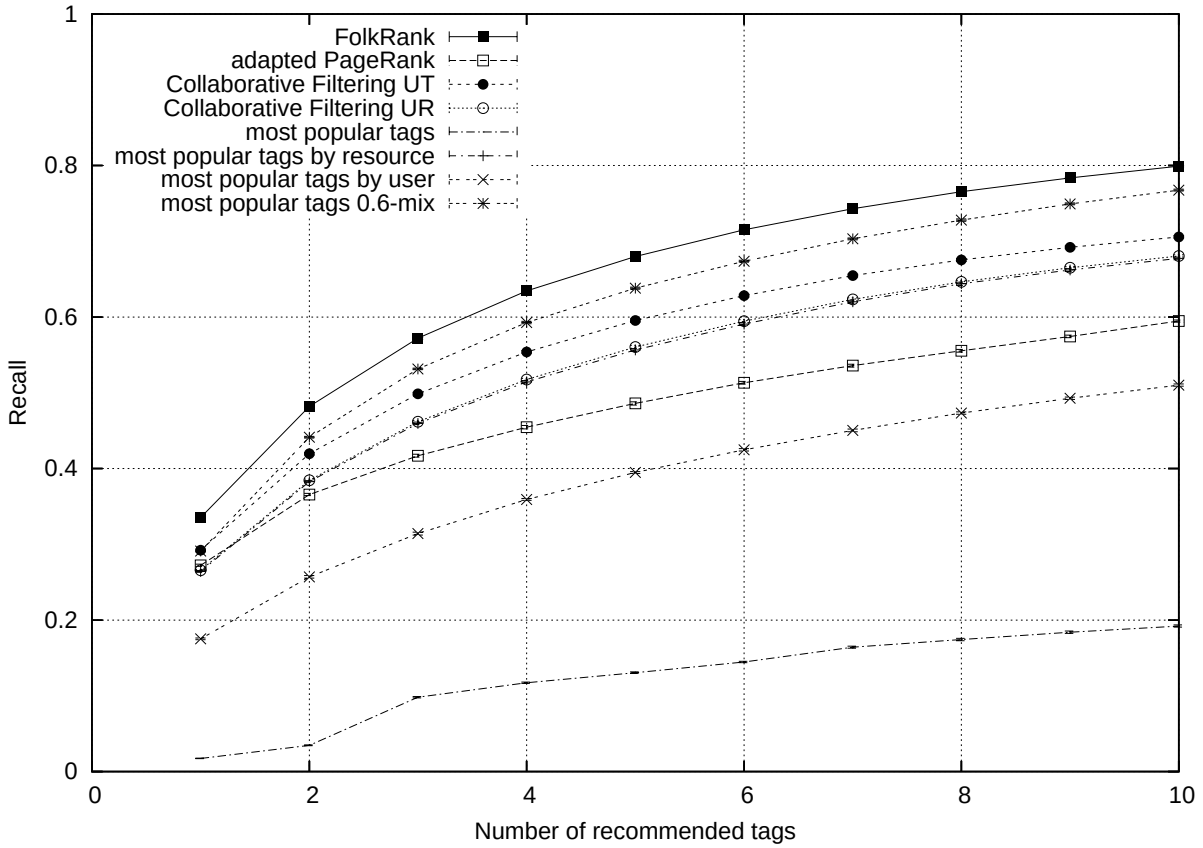

Fig. 3. Recall for del.icio.us p-core at level 10.

relevant tags—the exact tags of the user which *CF* could not. This is due to the fact that *FolkRank* considers, via the hypergraph structure, also the vocabulary of the user himself, which *CF* does not do. It was this observation that motivated the creation of the *most popular tags* ρ*–mix*-recommender, where we—in contrast to *CF*—include also the user's tags in the recommendations. As the diagrams show, we succeeded and could gain results better than those of *CF* and only slightly worse than those of *FolkRank*.

The standard deviation for the ten runs of all algorithms on this dataset is for both precision and recall below 3 ‰.

6.1.3. Comparison of Algorithms on the Unpruned Dataset

We conclude the evaluation on del.icio.us with results on the unpruned del.icio.us dataset, see Figure 5. Due to the long tail of users and resources which occur in only one post, we regarded only resources and users with at least two posts. Otherwise, most of the algorithms would not be able to produce recommendations. Apart from the *adapted PageRank*, the results are similiar to the results on the the p-core at level 10, with an overall decrease of both precision and recall. The only algorithm which seems to profit from the remaining long tail is the *adapted PageRank*. This is likely due to the fact that the many tags in the long tail together are able to outbalance to a certain degree the strong influence of the nodes with high edge degree. Nevertheless, it can not reach the performance of *FolkRank* or the *most popular tags* 0.6*–mix*.

The standard deviation for the ten runs of all algorithms on this dataset is for both precision and recall below 2 %.

6.2. BibSonomy

For this dataset, the results have a much larger standard deviation, as can be seen by the error bars in Figure 6. This is due to the fact that every run is averaging over 116 users only (cf. Table 2) and thus the performance of the ten runs differs more. Nevertheless,

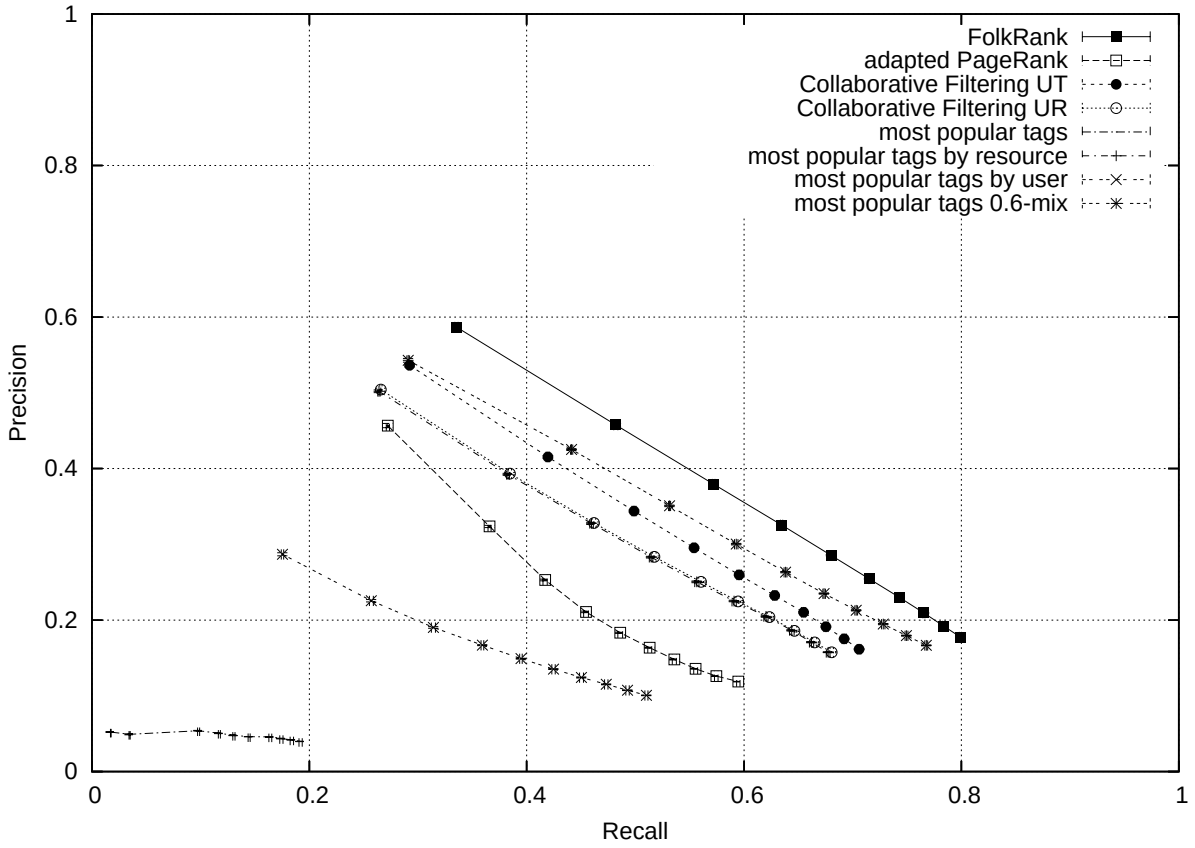

Fig. 4. Recall and Precision for del.icio.us p-core at level 10.

the tendency of the performance of the different methods is similar to the performance on the other datasets. *FolkRank* provides on average best precision and recall, followed by the *most popular tags* 0.6–*mix* recommender. Both *Collaborative Filtering* algorithms and *most popular tags by resource* show similar results for higher numbers of tags.

6.3. Last.fm

On this dataset, FolkRank again outperforms the other approaches. Here, its recall is considerably higher than on the other datasets, see Figure 7. Even when just two tags are recommended, the recall is close to 60 % and goes up to 92 % for 10 tags. The standard deviation for the ten runs of all algorithms on this dataset is for both precision and recall below 7 ‰.

The most surprising observation is, though, that here *most popular tags by user* is considerably better than *most popular tags by resource* and even *Collaborative Filtering*, such that it is the second best algorithm after *FolkRank*. An explanation could be the average number of tags a user has in this dataset (cf. Table 3). Compared to the del.icio.us and BibSonomy datasets, here the average is much lower with around twelve tags. Additionally, the average number of tags per resource in the last.fm dataset is much higher than in the other two datasets and in particular higher than the average number of tags per user (in contrast to the other two datasets, where it is the other way around). Hence, if a user has on average only twelve tags, proposing tags he used earlier instead of tags other users attached to the resource provides a better chance to suggest the tags the user finally chose. Needless to say that it would be interesting to know, why the averages on the last.fm dataset are so different from the other datasets. It could depend on the rather limited domain of the resources which can be tagged in last.fm, but might also result from the crawling strategy which was deployed to gather this dataset.

Due to the different performance of the *most popular tags by user/resource* recommendations, the perfor-

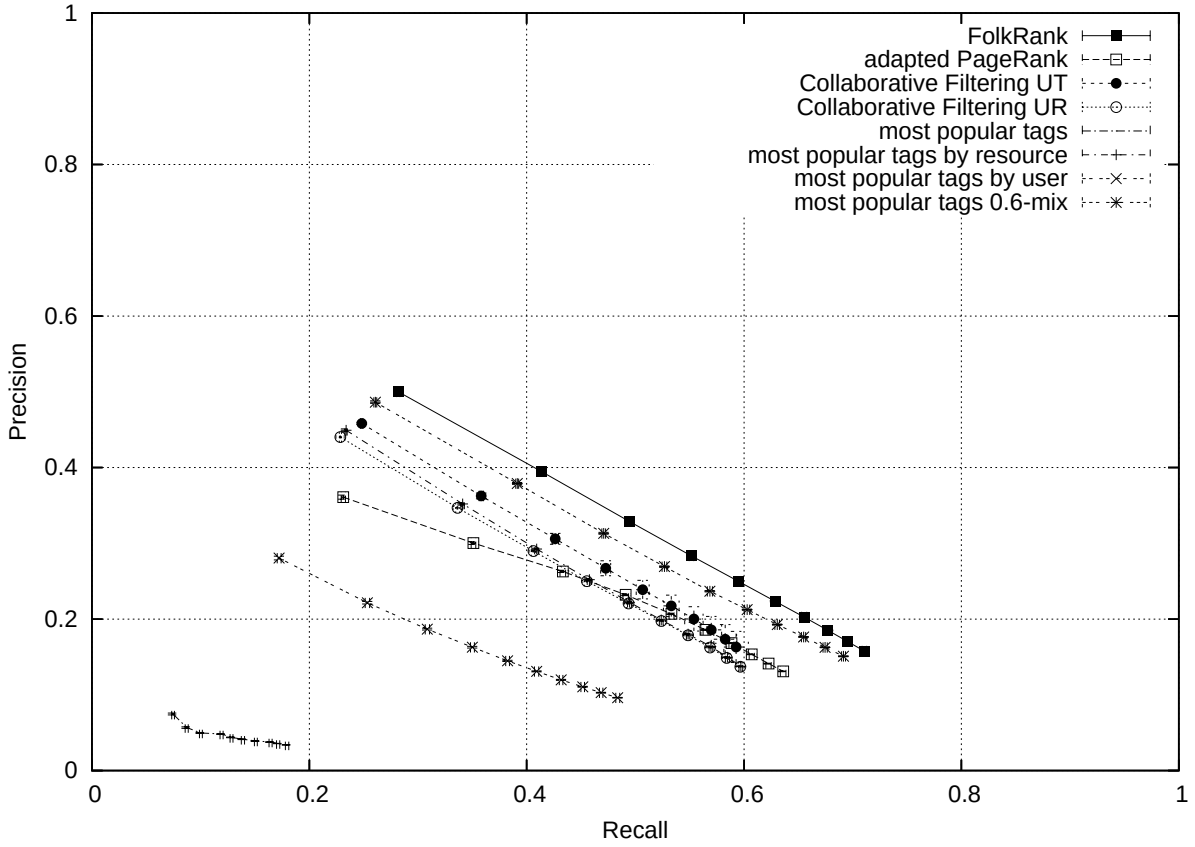

Fig. 5. Recall and Precision for del.icio.us.

Table 3
Average number of tags per user and tags per resource.

dataset	$\frac{1}{\lvert U\rvert}\sum_{u\in U}\lvert T_u\rvert$	$\frac{1}{\lvert R\rvert}\sum_{r\in R}\lvert T_r\rvert$
del.icio.us p-core at level 10	59.18	25.87
BibSonomy p-core at level 5	31.85	14.14
last.fm p-core at level 10	11.84	44.19

mance of the *most popular tags ρ–mix*, of course, differs significantly from the results on the other datasets. A comparison (not shown here) of different values for ρ showed, that the *most popular tags ρ–mix* on this dataset mostly performed worse than *most popular tags by user* (although always better than *most popular tags by resource*).

7. Conclusion

The presented results show that the graph-based approach of FolkRank is able to provide tag recommendations which are significantly better than those of approaches based on tag counts and even better than those of state-of-the-art recommender systems like Collaborative Filtering. The tradeoff is, though—as discussed in Section 4— that computation of FolkRank recommendations is cost-intensive so that one might prefer less expensive methods to recommend tags in a social bookmarking system.

The *most popular tags ρ–mix* approach proposed in this work has proven to be considered as a solution for this problem. It provides results which can almost reach the grade of FolkRank but which are extremely cheap to generate. Especially the possibility to use index structures (which databases of social bookmarking services typically provide anyway) makes this approach a good choice for online recommendations.

Finally, despite its simplicity and non-personalized aspect, the *most popular tags* achieved reasonable precision and recall on the small datasets (last.fm and BibSonomy) what indicates its adequacy for the cold start problem.

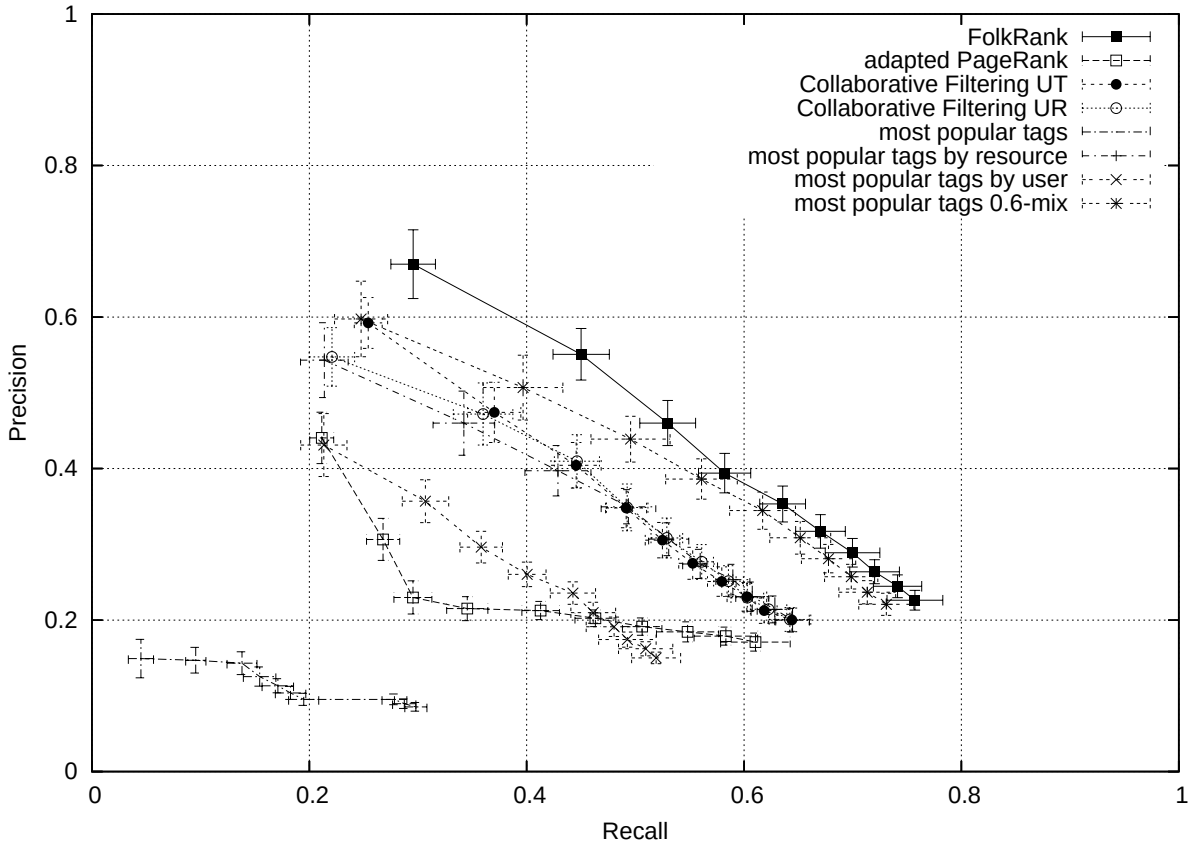

Fig. 6. Recall and Precision for BibSonomy p-core at level 5.

8. Summary and Future Work

In this paper we presented three classes of algorithms for tag recommendations in folksonomies: straightforward Collaborative Filtering adaptations based on projections, adaptations of the well-known PageRank algorithm, and simpler methods based on tag counts. We conducted experiments on three real-life datasets, showed that FolkRank outperforms the other methods, and that the *most popular tags ρ–mix* provides a good tradeoff between recommendation performance and computational costs. The main conclusions of our experiments were that the exploitation of the hypergraph structure in FolkRank yields a significant advantage and that simple methods based on tag counts can provide recommendations nearly as good as the best results with only minimal computational costs.

Currently, our approach for FolkRank always returns a fixed number of tags, often yielding low precision. Future work will include a method to determine a good cut-off point automatically. This, of course, is a problem which is worth to be looked at for other methods, too.

Particularly appealing would be the inclusion of the *most popular tags ρ–mix* method into some implemented social bookmarking system. Since three of the authors are involved in the development of BibSonomy, chances are good that soon this recommendation method will assist the users during tagging. It will be interesting to analyse its user acceptance.

Future work further includes improving the *most popular tags ρ–mix* method. One idea is to study different normalization aspects like the introduction of a frequency dependent normalization. This would allow to incorporate the differences in the tag frequency distributions of users and resources.

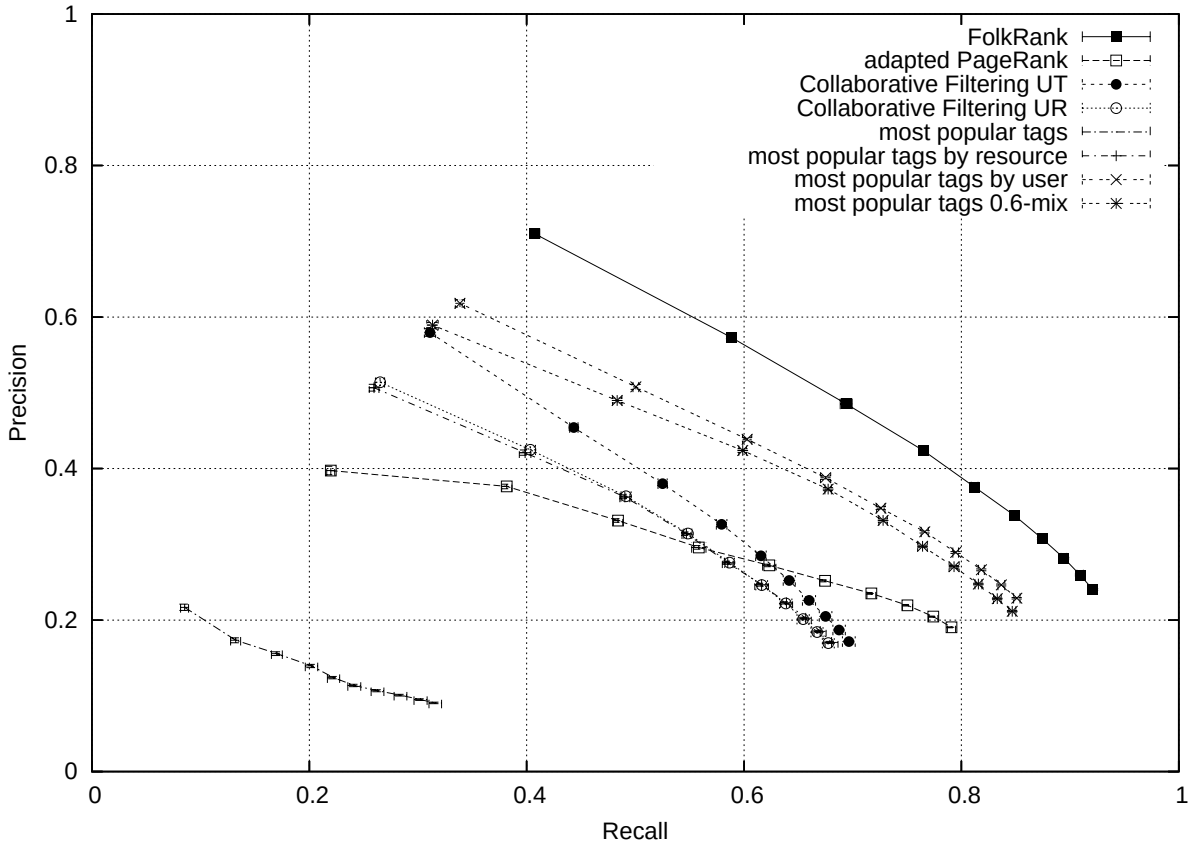

Fig. 7. Recall and Precision for Last.fm p-core at level 10

Acknowledgements

Part of this research was funded by the EU in the Nepomuk[23] (FP6-027705), Tagora[24] (FP6-2005-34721), and the X-Media[25] (IST-FP6-026978) projects.

References

[1] Pierpaolo Basile, Domenico Gendarmi, Filippo Lanubile, and Giovanni Semeraro. Recommending smart tags in a social bookmarking system. In *Bridging the Gep between Semantic Web and Web 2.0 (SemNet 2007)*, pages 22–29, 2007.

[2] V. Batagelj and M. Zaversnik. Generalized cores, 2002. cs.DS/0202039, http://arxiv.org/abs/cs/0202039.

[3] D. Benz, K. Tso, and L. Schmidt-Thieme. Automatic bookmark classification: A collaborative approach. In *Proceedings of the Second Workshop on Innovations in Web Infrastructure (IWI 2006)*, Edinburgh, Scotland, 2006.

[4] John S. Breese, David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*, pages 43–52, 1998.

[5] Sergey Brin and Lawrence Page. The Anatomy of a Large-Scale Hypertextual Web Search Engine. *Computer Networks and ISDN Systems*, 30(1-7):107–117, April 1998.

[6] Andrew Byde, Hui Wan, and Steve Cayzer. Personalized tag recommendations via tagging and content-based similarity metrics. In *Proceedings of the International Conference on Weblogs and Social Media*, March 2007.

[7] Ciro Cattuto, Vittorio Loreto, and Luciano Pietronero. Collaborative tagging and semiotic dynamics, May 2006. http://arxiv.org/abs/cs/0605015.

[8] Mukund Deshpande and George Karypis. Item-based top-n recommendation algorithms. *ACM Trans. Inf. Syst.*, 22(1):143–177, 2004.

[9] M. Dubinko, R. Kumar, J. Magnani, J. Novak, P. Raghavan, and A. Tomkins. Visualizing tags over time. In *Proc. of the 15th International WWW Conference*, Edinburgh, Scotland, 2006.

[10] Claudiu-S Firan, Wolfgang Nejdl, and Raluca Paiu. The benefit of using tag-based profiles. In *5th Latin American Web Congress, October 31 - November 2 2007, Santiago de Chile*, 2007.

[23] http://nepomuk.semanticdesktop.org
[24] http://www.tagora-project.eu [25] http://www.x-media-project.org

[11] Bernhard Ganter and Rudolf Wille. *Formal Concept Analysis: Mathematical Foundations*. Springer, Berlin – Heidelberg, 1999.

[12] H. Halpin, V. Robu, and H. Shepard. The dynamics and semantics of collaborative tagging. In *Proceedings of the 1st Semantic Authoring and Annotation Workshop (SAAW'06)*, 2006.

[13] Tony Hammond, Timo Hannay, Ben Lund, and Joanna Scott. Social Bookmarking Tools (I): A General Review. *D-Lib Magazine*, 11(4), April 2005.

[14] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, 22(1):5–53, 2004.

[15] Andreas Hotho, Robert Jäschke, Christoph Schmitz, and Gerd Stumme. Information retrieval in folksonomies: Search and ranking. In York Sure and John Domingue, editors, *The Semantic Web: Research and Applications*, volume 4011 of *Lecture Notes in Computer Science*, pages 411–426, Heidelberg, June 2006. Springer.

[16] Andreas Hotho, Robert Jäschke, Christoph Schmitz, and Gerd Stumme. Trend detection in folksonomies. In Yannis S. Avrithis, Yiannis Kompatsiaris, Steffen Staab, and Noel E. O'Connor, editors, *Proc. First International Conference on Semantics And Digital Media Technology (SAMT)*, volume 4306 of *LNCS*, pages 56–70, Heidelberg, Dec 2006. Springer.

[17] Robert Jäschke, Leandro Balby Marinho, Andreas Hotho, Lars Schmidt-Thieme, and Gerd Stumme. Tag recommendations in folksonomies. In Joost N. Kok, Jacek Koronacki, Ramon López de Mántaras, Stan Matwin, Dunja Mladenic, and Andrzej Skowron, editors, *Knowledge Discovery in Databases: PKDD 2007, 11th European Conference on Principles and Practice of Knowledge Discovery in Databases*, volume 4702 of *Lecture Notes in Computer Science*, pages 506–514, Berlin, Heidelberg, 2007. Springer.

[18] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.

[19] F. Lehmann and R. Wille. A triadic approach to formal concept analysis. In G. Ellis, R. Levinson, W. Rich, and J. F. Sowa, editors, *Conceptual Structures: Applications, Implementation and Theory*, volume 954 of *Lecture Notes in Computer Science*. Springer, 1995.

[20] Ben Lund, Tony Hammond, Martin Flack, and Timo Hannay. Social Bookmarking Tools (II): A Case Study - Connotea. *D-Lib Magazine*, 11(4), April 2005.

[21] Leandro Balby Marinho and Lars Schmidt-Thieme. Collaborative tag recommendations. In *Proceedings of 31st Annual Conference of the Gesellschaft für Klassifikation (GfKl), Freiburg*. Springer, 2007.

[22] Adam Mathes. Folksonomies – Cooperative Classification and Communication Through Shared Metadata, December 2004. http://www.adammathes.com/academic/computer-mediated-communication/folksonomies.html.

[23] Peter Mika. Ontologies Are Us: A Unified Model of Social Networks and Semantics. In Yolanda Gil, Enrico Motta, V. Richard Benjamins, and Mark A. Musen, editors, *ISWC 2005*, volume 3729 of *LNCS*, pages 522–536, Berlin Heidelberg, November 2005. Springer-Verlag.

[24] Gilad Mishne. Autotag: a collaborative approach to automated tag assignment for weblog posts. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 953–954, New York, NY, USA, 2006. ACM Press.

[25] P. Resnick, N. Iacovou, M. Suchak, P. Bergstorm, and J. Riedl. GroupLens: An Open Architecture for Collaborative Filtering of Netnews. In *Proceedings of ACM 1994 Conference on Computer Supported Cooperative Work*, pages 175–186, Chapel Hill, North Carolina, 1994. ACM.

[26] Badrul M. Sarwar, George Karypis, Joseph A. Konstan, and John Reidl. Item-based collaborative filtering recommendation algorithms. In *World Wide Web*, pages 285–295, 2001.

[27] Christoph Schmitz, Andreas Hotho, Robert Jäschke, and Gerd Stumme. Mining association rules in folksonomies. In V. Batagelj, H.-H. Bock, A. Ferligoj, and A. Žiberna, editors, *Data Science and Classification: Proc. of the 10th IFCS Conf.*, Studies in Classification, Data Analysis, and Knowledge Organization, pages 261–270, Berlin, Heidelberg, 2006. Springer.

[28] Gerd Stumme. A finite state model for on-line analytical processing in triadic contexts. In Bernhard Ganter and Robert Godin, editors, *ICFCA*, volume 3403 of *Lecture Notes in Computer Science*, pages 315–328. Springer, 2005.

[29] Karen Tso-Sutter, Leandro Balby Marinho, and Lars Schmidt-Thieme. Tag-aware recommender systems by fusion of collaborative filtering algorithms. In *Proceedings of 23rd Annual ACM Symposium on Applied Computing (SAC'08)*, Edinburgh, Scotland, 2007.

[30] W. Xi, B. Zhang, Y. Lu, Z. Chen, S. Yan, H. Zeng, W. Ma, and E. Fox. Link fusion: A unified link analysis framework for multi-type interrelated data objects. In *Proc. 13th International World Wide Web Conference*, New York, 2004.

[31] Yanfei Xu, Liang Zhang, and Wei Liu. Cubic analysis of social bookmarking for personalized recommendation. *Frontiers of WWW Research and Development - APWeb 2006*, pages 733–738, 2006.

[32] Zhichen Xu, Yun Fu, Jianchang Mao, and Difu Su. Towards the semantic web: Collaborative tag suggestions. In *Proceedings of the Collaborative Web Tagging Workshop at the WWW 2006*, Edinburgh, Scotland, 2006.