Improving Graph-based Approaches for Personalized Tag Recommendation

Maryam Ramezani
Karlsruhe Institute of Technology/ SAP Research, Karlsruhe, Germany
maryam.ramezani@sap.com

***Abstract*—Social tagging applications allow users to annotate online resources, resulting in a complex network of interrelated users, resources and tags often called a Folksonomy. A folksonomy is often represented as a hyper-graph in which each hyper-edge connects a user, resource and tag. This tripartite hyper-graph is often used by data mining applications to provide services for the user such as tag recommenders. This paper provides an overview on the state of the art of graph-based tag recommendation from a critical perspective. In addition, we suggest improving the existing graph-based tag recommendation techniques by introducing a new model of the folksonomy as a directed graph.**

I. INTRODUCTION

Collaborative tagging systems such as Delicious[1], lastFm[2], and Bibsonomy[3] have emerged as powerful applications for Internet users. Tagging systems support users with several benefits. First they allow users to organize their own data with a level of freedom not possible in traditional taxonomic filing systems. Secondly they provide users with the means to openly share this information among friends and colleagues. Thirdly they also allow anyone to utilize the collective knowledge of others for discovering new topics, resources or perhaps even new friends.

While social tagging systems have many benefits, they also present several challenges. Most tagging applications permit unsupervised tagging; users are free to use any tag they wish to describe a resource. This unsupervised tagging can result in tag redundancy – in which several tags have the same meaning – or tag ambiguity – in which a single tag has many different meanings. Such inconsistencies can confound users as they attempt to utilize the folksonomy. It can be difficult for users to traverse the sheer volume of data. Moreover, noise in the data can impede the users experience. Data mining applications such as tag recommenders make it easier for the user to navigate the system.

Tag recommendation, the suggestion of tags during the annotation process reduces the user effort. By reducing the effort users are encouraged to tag more frequently, apply more tags to an individual resource, reuse common tags, and perhaps use tags the user had not previously considered. Moreover, user error is reduced by eliminating redundant tags caused by capitalization inconsistencies, punctuation errors, misspellings and other discrepancies. The tag recommender can further promote a core tag vocabulary steering the user toward adopting certain tags while not imposing any strict rules. The tag recommender may even avoid ambiguous tags in favor of tags that offer greater information value. This may aid other users when navigating through the folksonomy to find interesting resources related to a tag which is more often used by other users. In order to develop a recommender applications in a social tagging system the first step is to create a model of the folksonomy that takes into account the information flow between users, resources and tags.

In this paper we present a critical view on the existing graph-based tag recommendation approaches specifically on one of the most popular techniques, called FolkRank. In addition, we propose a weighted directed graph which models the informational channels of a folksonomy. We then apply PageRank to this model for tag recommendation. Our claim for a directed graph to model the folksonomy is based on the observation that the user navigation from one object (user, resource, or tag) to another object is not symmetric and by considering different weights on the edges of each direction we can better model the navigating from one node to the other.

[1]http://delicious.com/
[2]http://www.last.fm/
[3]www.bibsonomy.org/

II. RELATED WORK

Tag recommenders assist in the tagging process by suggesting users a set of tags that they are likely to use for a resource. Personalized tag recommenders take the users' tagging behavior in the past into account when they recommend tags. Researchers have applied different data mining and machine learning techniques to the tag recommendation problem.

Association rules are explored in [10] to recommend tags and introduce an entropy-based metric o find how predictable a tag is. The title of a resource and the user vocabulary is used in [14] to generate recommendations. The results show that tags retrieved from the user's vocabulary outperform recommendations driven by resource information. Basile et al propose a classification algorithm for tag recommendation in [3] and Adrian et al. suggest a semantic tag recommendation system in the context of a semantic desktop in [2].

User-defined tags and co-occurrence are employed by [18] to recommend tags to users on Flickr. The assumption is that the user has already assigned a set of tags to a photo and the recommender uses those tags to recommend more tags. A similar study is conducted in [8] and a classification algorithm

doi:10.4304/jetwi.3.2.168-176

for tag recommendation is introduced. An adaption of K-Nearest Neighbor for tag recommendation is presented in [9] and it has been shown that incorporating user tagging habits into recommendation can improve recommendation. Moreover, [19] uses single aspect PLSA[4] for tag recommendation.

Hotho et al. suggest an approach for converting the folksonomy into an undirected tripartite graph and use an adaptation of PageRank (called Adapted PageRank or simply PageRank) to facilitate search and recommendation in folksonomies [12], [11]. However, their representation based on an undirected graph has a shortcoming in that weights that flow in one direction of an edge will go back along the same edge in the next iteration of PageRank. To overcome this shortcoming, FolkRank was proposed. FolkRank has been one of the most successful tag recommendation algorithms in folksonomies to date.

Increasing interest in improving the effectiveness of tag recommendation attracted many researchers from all over the world to the ECML/PKDD Discovery Challenge 2009 [7] which was mainly focused on tag recommendation. Different data mining approaches including content-based techniques, probabilistic models, and factor models have been applied for tag recommendation. The the most successful algorithm for graph-based tag recommendation was introduced by Rendle and Thieme [16] which uses pairwise interaction tensor factorization for tag recommendation. The algorithm explicitly models the pairwise interactions between users, items and tags. Figure 1 shows the results from authors' experiments on comparison of their proposed approach with other popular tag recommendation algorithms. We can see in this figure that PageRank shows the worst results while FolkRank and the suggested algorithm are comparable.

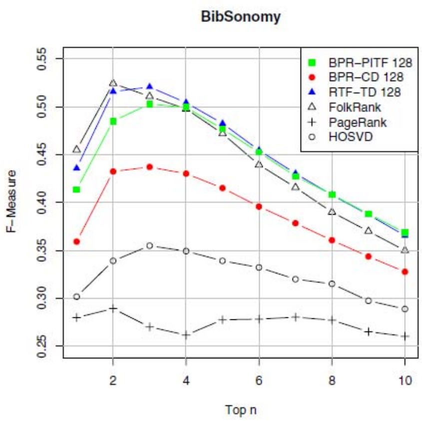

Fig. 1. Comparion of different state of the art tag recommendation techniques taken from [16]

Our contributions in this paper are two-fold. First, we demonstrate that with appropriate parameterizations, PageRank can outperform FolkRank in tag recommendation. Second,

[4]Probabilistic Latent Semantic Analysis

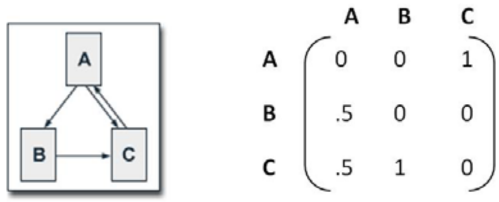

Fig. 2. A simplified example for presentation of PageRank algorithm

our extensive evaluation on three real world datasets reveal that PageRank generates better tag recommendations with a weighted directed graph model of the folksonomy than with an undirected graph model.

The rest of this paper is organized as follows. In section III, we provide a background on the PageRank algorithm and then we present the Adapted PageRank and FolkRank for folksonomies. Next, in section V-B we describe how to apply these algorithms for tag recommendation. Section VI presents our proposed algorithm for creating a weighted directed graph from a folksonomy. Our experimental methodology, datasets and results are offered in section VII. Finally, we conclude the paper with directions for future work.

III. Background on PageRank Algorithm

PageRank is a link analysis algorithm, introduced by Page and Brin [6] and it is the basic algorithm of the Google search engine. The underlying idea of PageRank is that pages that have many in-links from important pages are important themselves. PageRank uses probability distribution to represent the likelihood that a person randomly clicking on links will arrive at any particular page. The probability that the random surfer clicks on one link is given by the number of links on that page. For example, in figure 2 the probability of surfing from page B to page C is 1 while the probability of surfing from page A to page B is .5 since A has two out-links. The probability for the random surfer reaching one page is the sum of probabilities for the random surfer following links to this page. In our example, the probability of surfing to C is the sum of probability of surfing to B and probability of surfing to A. The PageRank algorithm considers that even an imaginary surfer who is randomly clicking on links might not click on an infinite number of links, but might get bored sometimes and jump to another page at random. At any step, the probability that the surfer does not follow the hyperlinks and instead jumps to a different page is γ. More formally, in PageRank, the authority of a page p, x_p, expresses the popularity of the page p and is defined based on the number of incoming links and on the authority of every page $q \in Q_p$ that connects to p with a forward link [5] and can be calculated as follows:

$$x_p = (1-\gamma) \sum_{q \in Q_p} \frac{x_q}{|H_q|} + \gamma \qquad (1)$$

Where H_q is the set of all pages linked from q, and $|H_q|$ is the *outdegree* of q. The equation can be reformulated as the

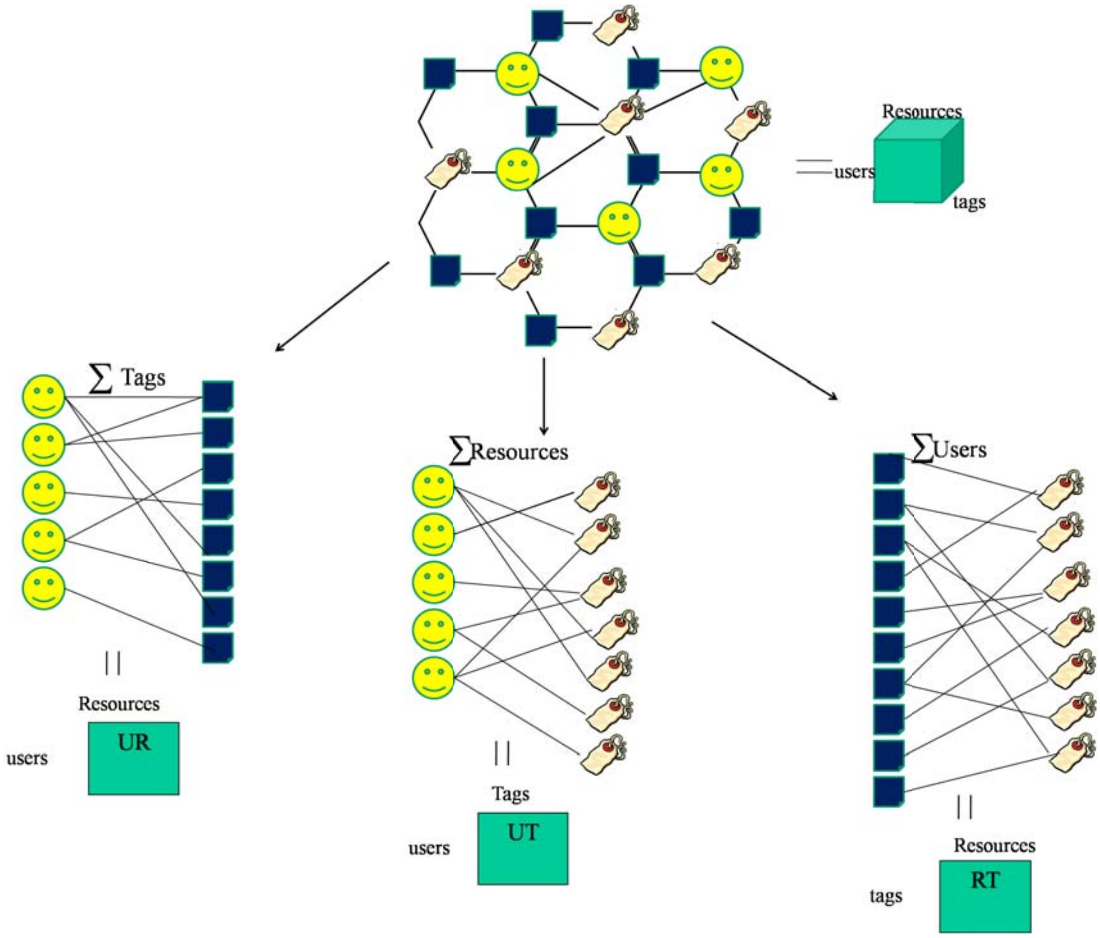

Fig. 3. We can reduce the hypergraph into three bipartite graphs with regular edges. The graphs model aggregate associations between users and resources (UR), users and tags (UT), and tags and resources (TR)

following linear system:

$$x = (1-\gamma)Wx + \gamma v \tag{2}$$

Here, the damping factor γ determines the influence of v, which is typically defined as $v = [1, \cdots, 1]^T$ but may be personalized with user preferences. The *transition matrix*, $W = \{w_{i,j}\}$ is defined as $w_{i,j} = 1/\left|H_j\right|$ if there exists a link from j to i and $w_{i,j} = 0$ otherwise. Every column must sum to either 1 or 0, making the matrix W column stochastic. The transition matrix for our simple example is shown in figure 2.

In order to compute personalized PageRank, v can be used to express user preferences by giving a higher weight to the components which represent the users preferred web pages. We call such vector, v, a "**preference vector**" and will use that for tag recommendation. In this case, the value of $\gamma \in [0,1]$ controls the influence of the preference vector. The preference vector allows the algorithm to be trained on a specific region of the graph.

IV. Folksonomy Graph

In order to model networks of folksonomies at an abstract level, we use the formalization introduced by Mika [15], representing the system as a tripartite graph with hyperedges. The set of vertices is partitioned into the three disjoint sets $U = u_1, u_2, ..., u_k$, corresponding the set of users, $T = \{t_1, t_2, ..., t_m\}$ the set of tags, $R = r_1, r_2, ...r_l$ the set of resources or objects annotated. In a social tagging system, users associate objects with tags, creating ternary associations between the user, the tag and the resource. Thus we can define A, a set of annotations, represented as user-resource-tag triples.

$$A \subseteq \{\langle u, r, t \rangle : u \in U, r \in R, t \in T\} \tag{3}$$

Thus, the folksonomy can be described as a four-tuple $D = \langle U, R, T, A \rangle$. To simplify analysis, we can reduce the hypergraph into three bipartite graphs with regular edges. The graphs model aggregate associations between users and resources (UR), users and tags (UT), and tags and resources (TR) [15], [12]. Aggregate projections of the data can reduce the dimensionality by sacrificing information [17]. The relation between resources and tags can be formulated as a two-dimensional projection, RT, such that each entry, $RT(r,t)$, is the weight associated with the resource, r, and the tag, t. This weight may be binary, merely showing that one or more users have applied that tag to the resource, or it may be finer grained using the number of users that have applied that tag to the resource:

$$RT(r,t) = |\{a = \langle u,r,t \rangle \in A : u \in U\}| \tag{4}$$

Such a measure is equivalent to *term frequency* or *tf* common in Information Retrieval. Similar two-dimensional projections can be constructed for *UT* in which the weights correspond to users and tags, and *UR* in which the weights correspond to users and resource where

$$UT(u,t) = |\{a = \langle u,r,t \rangle \in A : r \in R\}| \tag{5}$$

$$UR(u,r) = |\{a = \langle u,r,t \rangle \in A : t \in T\}| \tag{6}$$

In words, the bipartite graph UT links the users to the tags that they have associated at least to one resource. Each link is weighted by the number of times the person has used that tag. UR links the users to the resources that they have tagged . Each link is weighted by the number of tags the person has used for that resource. Figure 3 shows the steps of the described process.

V. FOLKSONOMY-ADAPTED PAGERANK

Hotho et.al [12], [11] introduced an adaption to PageRank for folksonomies with the same basic notion that a resource which is tagged with important tags by important users becomes important itself. The same holds, symmetrically, for tags and users. Unlike the original PageRank that only deals with pages and links between them, the folksonomy Adapted PageRank considers three dimensions of users, resources, and tags. The folksonomy can be represented by a tripartite hypergraph in which the vertices are mutually reinforcing each other by spreading their weights.

Hotho et. al use the projections introduced in section IV and reduce the tripartite hypergraph to three bipartite graphs with regular edges, RT, UR, and UT, which reflect the relations between resources and tags, users and resources, and users and tags respectively. These three projections taken together produce a weighted undirected tripartite graph with set of nodes: $V = U \cup R \cup T$ and weighted edges $E = \{\{u,t\},\{r,t\},\{u,r\}|(u,t,r) \in A\}$, with each edge $\{u,t\}$ being weighted with $UT(u,t)$, each edge $\{t,r\}$ with $RT(r,t)$, and each edge $\{u,r\}$ being weighted with $UR(u,r)$. The adjacency matrix of this graph is shown in figure 4. The folksonomy transition matrix is built by normalizing each column of the matrix to 1. Such column-stochastic matrix is applied as W in equation 2 to find the rank of each element in folksonomy.

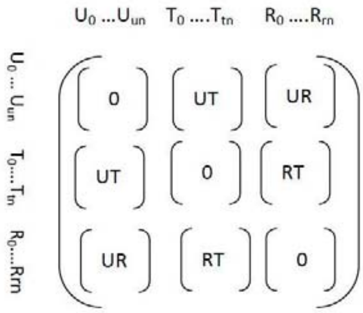

Fig. 4. The folksonomy adjacency matrix

A. FolkRank

The problem with adapted PageRank is that it is based on an undirected graph while the origin of PageRank comes from the in-link and out-link concepts. Thus, in the Adapted PageRank algorithm, weights that flow in one direction of an edge will basically swash back along the same edge in the next iteration. Therefore the result is very similar to a ranking based on counting edge degrees [12]. Hotho et al. [12], [11] suggest FolkRank to solve this problem.

The FolkRank vector is taken as a difference between two computations of PageRank: one with a preference vector and one without the preference vector. The preference vector (v in equation 2) is designed in a way that it gives higher weight to specific elements based the goal of ranking. For example, if the goal is to search for resources given a specific tag, the preference vector would give higher weight to that tag. More formally, if we consider x_0 to be the fixed point from equation 2 without preference vector and x_1 as the fixed point with preference vector p, then the FolkRank vector is defines as

$$x = x_1 - x_0 \tag{7}$$

Because a generic but popular item will receive a similar PageRank score in both models, its FolkRank will be reduced to 0 (or less, if its new score is less than the original). The inventors of FolkRank [12], [13] claim that x resulted from FolkRank provides valuable results on a large-scale real-world dataset while x_1, provides an unstructured mix of topic-relevant elements with elements having high edge degree.

However, our experimental results and analysis in the next sections show that with correct parameterization of the damping factor, adapted PageRank produces better results than FolkRank.

B. Graph-Based Tag Recommendation in Folksonomies

In traditional recommendation algorithms the input is often a user, u, and the output is a set of items, I. The user experience is improved if this set of items is relevant to the user's needs. Tag recommendation in folksonomies however differs in that the input is both a user, u, and a resource, r. The output remains a set of items, in this case a recommended set of tags, T_r. An algorithm for tag recommendation in folksonomies therefore requires a means to include both user and resource information in the process so that the recommendation set includes tags that are relevant to the resource and also represent the user's tagging practice.

In order to generate tag recommendations the preference vector is biased towards the query user and resource [13].

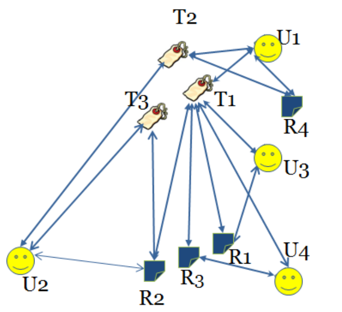

Fig. 5. An arbitrary example of folksonomy

These elements are given a substantial weight in the preference vector where all other elements have uniformly small weights. Similar to [13] each user, tag and resource gets a preference weight of 1 while the target user and resource get a preference weight of $1+|U|$ and $1+|R|$, resp.

The value of $\gamma \in [0,1)$ in equation 2 controls the impact of the preference vector. The higher it gets, the more the algorithm is biased toward the target user and resource. However, in previous tag recommendation experiments, authors [13], [16] have only tested the algorithms by fixing the value of $\gamma = .3$ without any specific reasoning, which indeed provides the best results for FolkRank but not for the PageRank algorithm. In this work, we will show that with increasing the value of γ, FolkRank will not do as well, and in fact PageRank can provide much better results.

VI. A Weighted Directed Graph Model for Folksonomies

The problem with the weighting schema of Adapted PageRank is that the weight of the undirected edges does not accurately reflect the flow of information across the folksonomy. We extend this graph model to consider the expectation of hypothetical user navigating from one node to another. Consider r representing a non-popular resource and t representing a popular tag associated to r in a folksonomy. For instance, in figure 5, an arbitrary example is given where T_1 is a popular tag while R_3 is a non-popular resource. In the current weighting schema of Adapted PageRank, the weight between r and t is defined by the number of users who associate them together represented with $RT(r,t) = |u \in U : (u,t,r) \in A|$.

Our weighting schema suggests that the weight from r to t should be higher than the weight from t to r since t is a popular tag and a user is more likely to navigate from a non-popular resource to a popular tag than navigating from a popular tag to a non-popular resource.

Our proposed weighting approach is inspired by the weighted PageRank algorithm [1], where the importance of pages are considered for weighting the graph. The weighted PageRank algorithm in [1] assigns larger rank values to more important (popular) pages instead of dividing the rank value of a page evenly among its out-link pages. Each out-link page gets a value proportional to its popularity (its number of in-links and out-links).

We take a similar approach and our goal is to assign different weights to each direction of the link based on the popularity of the in-link element. Since we face a folksonomy graph, we have three different types of vertices (U,R,T) and each edge connects two types of the vertices. The popularity of the elements is defined differently for each type of edge and is based on the number of connected elements of the third type of vertex. To clarify, consider an edge between resource r and tag t. Popularity of t is defined as the sum of all users who have used the tag t (sum of the in-links to tag t from the user nodes). The weight from r to t is defined as multiplicative factor of popularity and $RT(r,t)$. Similarly, the popularity of r will be the total of number of users who have annotated resource r in their profile. In our arbitrary example, the number of users who have tag T_1 in their profile is 3 while the number of users who have resource R_3 is 1. So the weight of the link from R_3 to T_1 is three times higher than the weight from T_1 to R_3. Equations 8 and 9 show the formal definitions of edge weights from r to t and from t to r respectively.

$$weight(r \to t) = RT(r,t) \times \sum_{r' \in R} RT(r',t) \tag{8}$$

$$weight(t \to r) = RT(r,t) \times \sum_{t' \in T} RT(r,t') \tag{9}$$

The weight of an edge between tag t and user u is defined as follows. Popularity of tag t is the number of all resources associated with t and the popularity of user u is the sum of all resources tagged by the user u. The following equations show the formal definitions for the edge weights.

$$weight(u \to t) = UT(u,t) \times \sum_{u' \in U} UT(u',t) \tag{10}$$

$$weight(t \to u) = UT(u,t) \times \sum_{t' \in T} UT(u,t') \tag{11}$$

Weights of UR edges are defined similarly. Popularity of user u is defined as the number of all tags the user has applied and popularity of r will be the sum of number of tags assigned to r. Thus, the weights are defined as follows:

$$weight(u \to r) = UR(u,r) \times \sum_{u' \in U} UR(u',r) \tag{12}$$

$$weight(r \to u) = UR(u,r) \times \sum_{r' \in R} UR(u,r') \tag{13}$$

With the proposed weighting schema, we have a weighted directed graph and we can apply the PageRank algorithm. In the following section we use the suggested graph model for tag recommendation and compare the results with FolkRank and PageRank with undirected graph.

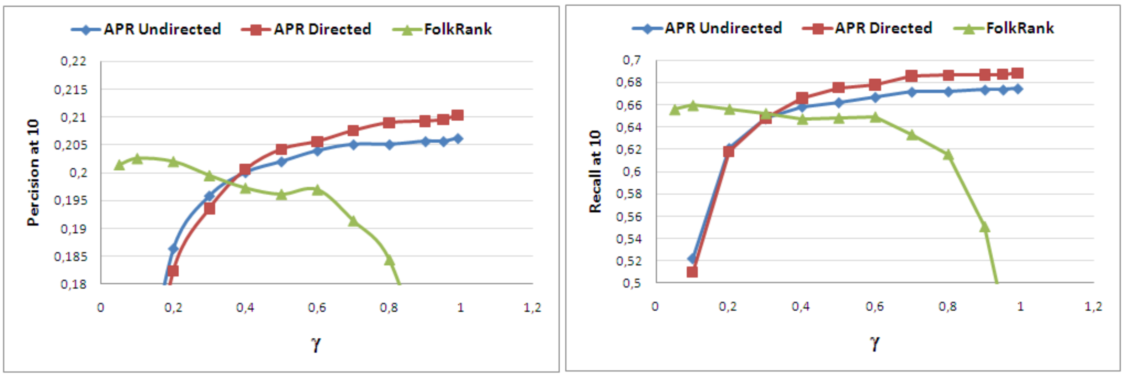

Fig. 6. The effect of changing γ on precision and recall for a recommendation set of 10 tags in Bibsonomy data set

VII. Experimental Evaluation

Our tag recommendation algorithm uses the PageRank algorithm with the directed graph described above. We also experimentally analyze the effect of changing the parameter γ in equation 2 for the three approaches: PageRank with undirected graph, FolkRank, and PageRank with directed graph. As described before, for all experiments, we bias the preference vector by setting the weight of $1+|U|$ and $1+|R|$ for the target user and resource respectively, while other users, tags and resources get equal weight of 1.

First, we describe the methods used to gather and pre-process data for the experiments and provide details of our datasets. Next, we discuss our testing methodology and briefly explain the common metrics recall and precision and then detail the results of our experiments.

A. Datasets

We have chosen three datasets for our experiments: Delicious, Bibsonomy and Citeulike. In order to reduce noise and focus on the denser portion of the dataset a *P*-core was taken such that each user, resource and tag appear in at least p posts as in [4], [13]. A post is defined as as a user, resource and all tags applied by that user to that resource.

Folksonomy	Delicious	Citeulike	Bibsonomy
Users	7,665	2,051	402
Resources	15,612	5,376	2,014
Tags	5,746	3,343	1,755
Posts	720,788	42,278	15,760
Annotations	2,762,235	105,873	53,554

TABLE I
DATASETS

Delicious is a popular Web site in which users annotate URLs. On 10/19/2008, 198 of the most popular tags were taken from the user interface. For each of these tags the 2,000 most recent annotations including the contributors of the annotations were collected. This resulted in 99,864 distinct usernames. For each user the social network was explored recursively resulting in a total of 524,790 usernames. From 10/20/2008 to 12/15/2008 the complete profiles of all users were collected. Each user profile consisted of a collection of posts including the resource, tags and date of the original bookmark. The top 100 most prolific users were visually inspected; twelve were removed from the data because their post count was many orders of magnitude larger than other users and were suspected to be Web-bots. Due to memory and time constraints, 10% of the user profiles was randomly selected. A p-core of 20 was taken from this dataset for experiments.

The Bibsonomy dataset was gathered on 1/1/2009 encompassing the entire system. This data set has been made available by the system administrators. A 5-core was taken to reduce noise and increase density.

Citeulike is a popular online tool used by researchers to manage and discover scholarly references. They make their dataset freely available to download. On 2/17/2009 the most recent snapshot was downloaded. The data contains anonymous user ids and posts for each user including resources, the date and time of the posting and the tags applied to the resource. A *P*-core of 5 was calculated.

Table I summarizes the base statistics for each dataset. From these fundamental numbers we derive a few simple statistics which help to characterize the "density" of the post-processed datasets. The average number of tags in a Delicious post is 3.82, and the average number of posts per user is 94.04. For Citeulike, the averages are 2.50 tags per post and 20.61 posts per user; for Bibsonomy, 3.39 and 39.2. We therefore note that Delicious has by far the most tagging data per user, followed by Bibsonomy, and Citeulike last. In terms of posts per resource, we again see that Delicious has far and away the highest average with 46.1; Citeulike and Bibsonomy are nearly identical in this respect (7.86 vs. 7.82, respectively).

An important distinction between the datasets is their focus. Users in Delicious are able to tag any URL available on the Web. As such an individual's interests are often varied encompassing many topics. In Citeulike, however, researchers tag scholarly publications and their tagging is often focused in their area of expertise. Due to its dual-nature, we expect Bibsonomy users to display a somewhat mixed approach.

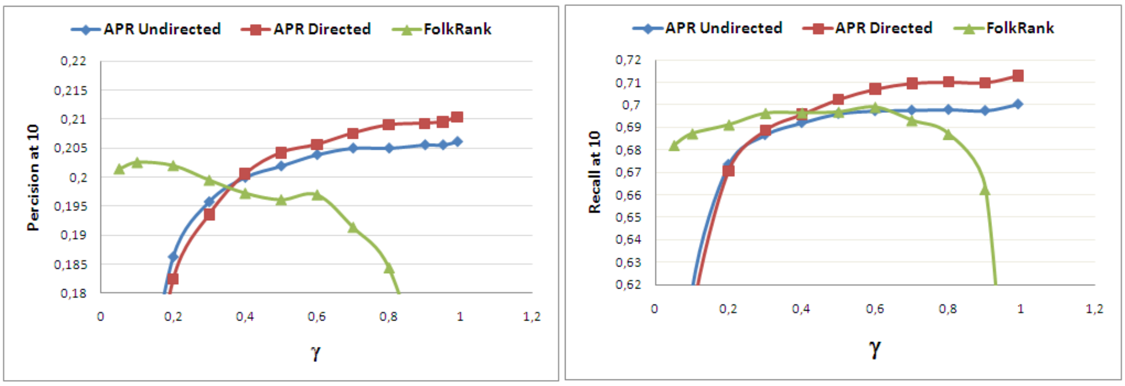

Fig. 7. The effect of changing γ on precision and recall for a recommendation set of 10 tags in Citeulike data set

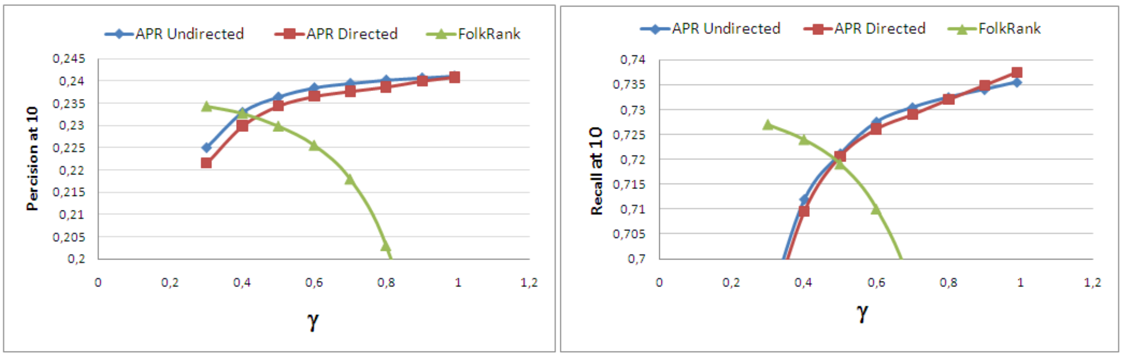

Fig. 8. The effect of changing γ on precision and recall for a recommendation set of 10 tags in Delicious data set

We have adopted the test methodology as described in [13]. In this approach, called *LeavePostOut*, a single post is randomly removed from each user's profile. The training set is then comprised of all of the remaining data, while the test set contains one test case per user. Each test case consists of a user, u, a resource, r, and all the tags the user has applied to that resource. These tags, T_h, are analogous to the holdout set commonly used in Information Retrieval. The tag recommendation algorithms accept the user-resource pair and return an ordered set of recommended tags, T_r. From the holdout set and recommendation set utility metrics were calculated. For each evaluation metric the average value is calculated across all test cases.

For the evaluation we use the common recall and precision measures as is common in Information Retrieval. Recall is a metric for completeness of the recommendation results and measures the percentage of items in the recommendation set that appear in the test set. Precision is a metric for exactness of the recommendation results and measures the percentage of items in the test set that appear in the recommendation set. In order to be able to compare the performance of the algorithms, we use F-measure defined as follows:

$$F\text{-}Measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{14}$$

B. Experimental Results

Here we present our experimental results with tuning of value of γ and changing the number of recommended tags. We have recorded the values of precision and recall with changing the number of recommended tags from 1 to 10 and also changing the value of γ between .05 to 1. The Adapted PageRank results are shown with abbreviation of "APR" in the graphs. Figures 6 through 8 show the effect of changing the value of γ on precision and recall for the three data sets. The charts show the precision and recall values for recommendation set of 10 tags (Recall at 10 and Precision at 10) when changing the value of γ.

Increasing the value of γ indicates a greater emphasis on the preference vector and our results show that increasing γ has a consistently positive effect on tag recommendation using the Adapted PageRank algorithm. Even though the results for $\gamma = .99$ still show an improvement, setting the value of $\gamma = 1$ will drastically reduce the recommendation precision to nearly zero. This is expected, as $\gamma = 1$ in equation 2 means that there is no learning involved and the final resulting vector is equal to the preference vector.

Our results show that for FolkRank, on the other hand, by increasing the value of γ, the performance decreases. To explain this we should note that FolkRank is basically the difference of the resulting weights of the adapted PageRank

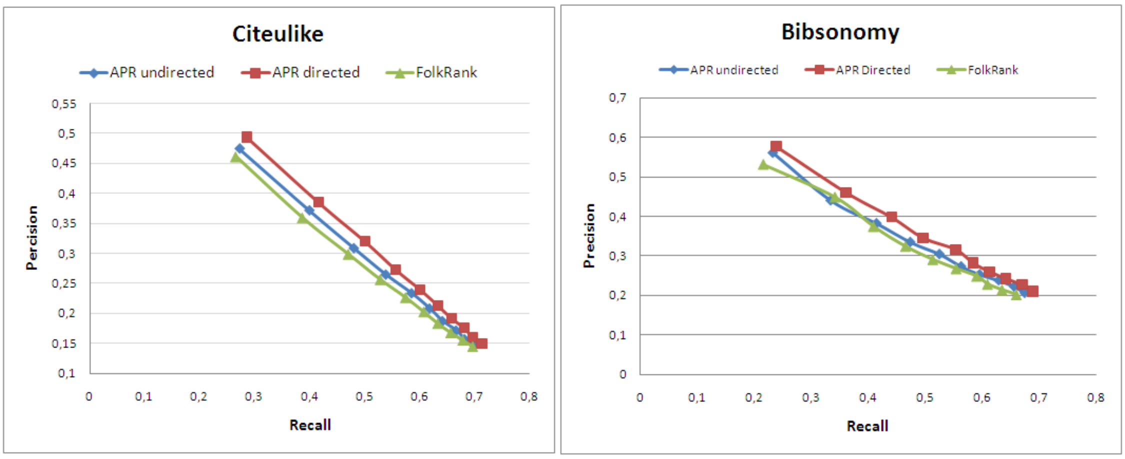

Fig. 9. Comparison of precision and recall for undirected APR,directed APR for $\gamma = .99$ and FolkRank for $\gamma = .3$

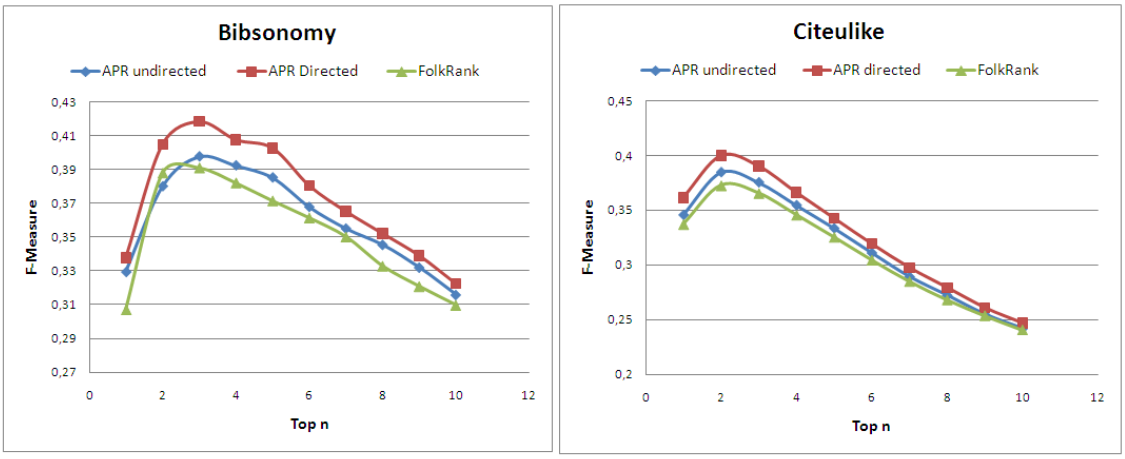

Fig. 10. Comparison of F-measure for undirected APR, directed APR for $\gamma = .99$ and FolkRank for $\gamma = .3$

with and without preference vector. As the value of γ increases the result is more and more focused on the preference vector which results in higher weights for the nodes that are connected to the target nodes. The differential approach in FolkRank is supposed to compute a topic-specific ranking of the elements in a folksonomy. However, the fact that the weights from the PageRank without preference (x_0) are subtracted from the one with preference vector (x_1) creates negative values in the resulting vector for any node which has a higher value in x_1. This is useful for small values of γ to remove the high weighted popular nodes, however, as the value of γ increases x_1 is more and more representative of the actual ranks considering the specific resource and tag. By subtracting x_0 from x_1 we end up with an ad-hoc unrealistic weight vector, since we might omit many high weighted nodes in x_1 only because they have occurred with a higher weight in x_0.

The above charts show the results when the number of recommended tags is fixed to 10. We want to examine if this results are valid when we change the number of recommended tags. Figure 9 shows the results for different values of precision and recall when recommending from 1 to 10 tags. In this experiment, we keep the value of γ constant and set it to the value that results in best performance for each technique. From figure 7 we can observe that this value is .3 for FolkRank and .99 for Adapted PageRank. In this figure we can observe that the adapted PageRank using the directed graph outperforms the undirected version and the FolkRank for all values of precision and recall. Figure 10 shows similar results comparing the F-measure for different approaches. This figure confirms that the performance of the adapted pagerank is better than FolkRank, and that the directed graph model outperforms the undirected one for recommendation of any number of tags.[5] Note that the rightmost point of this graph is the one which is presented in figure 7. Although in this figure it seems that all points converge to the same value, our significance tests show that the differences are significant even for that point.

[5]The difference between the F-measure values in figure 10 and 1 is because of the difference in the data set used in our experiments. Our experiments were performed before the release of the new data set by Bibsonomy.

C. Discussion

The results from different data sets show that increasing the value of γ results in increasing the recommendation accuracy of the PageRank algorithm and it has the opposite effect for the FolkRank. We can see that performance of FolkRank drops when the value of γ increases. The comparison of directed and undirected graphs show that directed graph produces better or similar results across different datasets. In Bibsonomy and Citeulike the results of the directed graph are considerably better for certain values of γ.

We performed significance test to determine if the differences between observed values from each approach in different data sets are significant. We used pair sample t-test and compared the mean of precision and recall resulting from the constant γ which produces the best results for each technique. The results from the significance test show that the differences between the values from the directed graph is significantly better than the undirected one for Bibsonomy and Citeulike datasets with $p < .001$. However, in Delicious dataset the differences are not significant which means that we can not reject the null hypothesis that the two approaches (directed and undirected graph models) produce similar results. This difference is likely because of existence of much broader concepts in Delicious and deserves further investigation. The significance tests show that in all datasets the Adapted PageRank significantly outperforms FolkRank.

VIII. Conclusion and Future Work

In this paper we suggested to model the folksonomy as a weighted directed graph which can capture the informational channels of a folksonomy. We then applied PageRank to this model for tag recommendation. Our extensive evaluation on three real world datasets revealed that with appropriate parameterization, Adapted PageRank can outperform FolkRank in tag recommendation. In addition, we have shown that with modeling the folksonomy as a directed weighted graph, we can get further improvement in accuracy of the recommendations.

Future work can improve the weighting schemas to better model the probability of navigation in the folksonomy and apply the graph models for search and resource recommendation. Moreover, efficiency and scalability are major drawbacks of this algorithm which deserve more attention. In the current setting, since the preference vector is dependent on the target resource and user, all the iterative calculation must be done in online recommendation phase which depending on the size of the folksonomy can be quite time consuming. Developing new algorithms that can help calculate the personalized PageRank with an incremental approach -so that the complete iteration calculations are not necessary at the run time- can be a valuable future work which can improve personalized search and recommendation not only in the social tagging environment but also in other Web applications.

IX. Acknowledgments

A preliminary version of this paper has been presented in the 2nd ACM RecSys10 Workshop on Recommender Systems and the Social Web. Thanks to Jonathan Gemmell, Thomas Schimoler, and Bamshad Mobasher for their considerable help and contributions on this work.

References

[1] Weighted pagerank algorithm. In *CNSR '04: Proceedings of the Second Annual Conference on Communication Networks and Services Research*, pages 305–314. IEEE Computer Society, 2004.

[2] Benjamin Adrian, Leo Sauermann, and Thomas Roth-Berghofer. Contag: A semantic tag recommendation system. In Tassilo Pellegrini and Sebastian Schaffert, editors, *Proceedings of I-Semantics' 07*, pages pp. 297–304. JUCS, 2007.

[3] 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.

[4] V. Batagelj and M. Zaveršnik. Generalized cores. *Arxiv preprint cs/0202039*, 2002.

[5] Monica Bianchini, Marco Gori, and Franco Scarselli. Inside pagerank. *ACM Trans. Interet Technol.*, 5(1):92–128, 2005.

[6] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *Seventh International World-Wide Web Conference (WWW 1998)*, 1998.

[7] Folke Eisterlehner, Andreas Hotho, and Robert Jschke, editors. *ECML PKDD Discovery Challenge 2009 (DC09)*, volume 497 of *CEUR-WS.org*, September 2009.

[8] Nikhil Garg and Ingmar Weber. Personalized, interactive tag recommendation for flickr. In *RecSys '08: Proceedings of the 2008 ACM conference on Recommender systems*, pages 67–74, New York, NY, USA, 2008. ACM.

[9] Jonathan Gemmell, Thomas Schimoler, Maryam Ramezani, and Bamshad Mobasher. Adapting k-nearest neighbor for tag recommendation in folksonomies. *Intelligent Techniques for Web Personalization & Recommender Systems*, 2009.

[10] Paul Heymann, Daniel Ramage, and Hector Garcia-Molina. Social tag prediction. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 531–538, New York, NY, USA, 2008. ACM.

[11] Andreas Hotho, Robert Jäschke, Christoph Schmitz, and Gerd Stumme. Folkrank: A ranking algorithm for folksonomies. In *Proc. FGIR 2006*, 2006.

[12] Andreas Hotho, Robert Jäschke, Christoph Schmitz, and Gerd Stumme. Information retrieval in folksonomies: Search and ranking. In *The Semantic Web: Research and Applications*, pages 411–426, 2006.

[13] R. Jaschke, L. Marinho, A. Hotho, L. Schmidt-Thieme, and G. Stumme. Tag Recommendations in Folksonomies. *Lecture Notes in Computer Science*, 4702:506, 2007.

[14] Marek Lipczak. Tag recommendation for folksonomies oriented towards individual users. In *Proceedings of the ECML/PKDD 2008 Discovery Challenge Workshop*, 2008.

[15] P. Mika. Ontologies are us: A unified model of social networks and semantics. *Web Semantics: Science, Services and Agents on the World Wide Web*, 5(1):5–15, 2007.

[16] Steffen Rendle and Lars Schmidt-Thieme. Pairwise interaction tensor factorization for personalized tag recommendation. In *WSDM '10: Proceedings of the third ACM international conference on Web search and data mining*, pages 81–90, New York, NY, USA, 2010. ACM.

[17] C. Schmitz, A. Hotho, R. Jaschke, and G. Stumme. Mining association rules in folksonomies. In *Proc. IFCS 2006 Conference*, pages 261–270. Springer, 2006.

[18] Börkur Sigurbjörnsson and Roelof van Zwol. Flickr tag recommendation based on collective knowledge. pages 327–336, 2008.

[19] Robert Wetzker, Winfried Umbrath, and Alan Said. A hybrid approach to item recommendation in folksonomies. In *ESAIR '09: Proceedings of the WSDM '09 Workshop on Exploiting Semantic Annotations in Information Retrieval*, pages 25–29, New York, NY, USA, 2009. ACM.

Call for Papers and Special Issues

Aims and Scope

Journal of Emerging Technologies in Web Intelligence (JETWI, ISSN 1798-0461) is a peer reviewed and indexed international journal, aims at gathering the latest advances of various topics in web intelligence and reporting how organizations can gain competitive advantages by applying the different emergent techniques in the real-world scenarios. Papers and studies which couple the intelligence techniques and theories with specific web technology problems are mainly targeted. Survey and tutorial articles that emphasize the research and application of web intelligence in a particular domain are also welcomed. These areas include, but are not limited to, the following:

- Web 3.0
- Enterprise Mashup
- Ambient Intelligence (AmI)
- Situational Applications
- Emerging Web-based Systems
- Ambient Awareness
- Ambient and Ubiquitous Learning
- Ambient Assisted Living
- Telepresence
- Lifelong Integrated Learning
- Smart Environments
- Web 2.0 and Social intelligence
- Context Aware Ubiquitous Computing
- Intelligent Brokers and Mediators
- Web Mining and Farming
- Wisdom Web
- Web Security
- Web Information Filtering and Access Control Models
- Web Services and Semantic Web
- Human-Web Interaction
- Web Technologies and Protocols
- Web Agents and Agent-based Systems
- Agent Self-organization, Learning, and Adaptation
- Agent-based Knowledge Discovery
- Agent-mediated Markets
- Knowledge Grid and Grid intelligence
- Knowledge Management, Networks, and Communities
- Agent Infrastructure and Architecture
- Agent-mediated Markets
- Cooperative Problem Solving
- Distributed Intelligence and Emergent Behavior
- Information Ecology
- Mediators and Middlewares
- Granular Computing for the Web
- Ontology Engineering
- Personalization Techniques
- Semantic Web
- Web based Support Systems
- Web based Information Retrieval Support Systems
- Web Services, Services Discovery & Composition
- Ubiquitous Imaging and Multimedia
- Wearable, Wireless and Mobile e-interfacing
- E-Applications
- Cloud Computing
- Web-Oriented Architectrues

Special Issue Guidelines

Special issues feature specifically aimed and targeted topics of interest contributed by authors responding to a particular Call for Papers or by invitation, edited by guest editor(s). We encourage you to submit proposals for creating special issues in areas that are of interest to the Journal. Preference will be given to proposals that cover some unique aspect of the technology and ones that include subjects that are timely and useful to the readers of the Journal. A Special Issue is typically made of 10 to 15 papers, with each paper 8 to 12 pages of length.

The following information should be included as part of the proposal:

- Proposed title for the Special Issue
- Description of the topic area to be focused upon and justification
- Review process for the selection and rejection of papers.
- Name, contact, position, affiliation, and biography of the Guest Editor(s)
- List of potential reviewers
- Potential authors to the issue
- Tentative time-table for the call for papers and reviews

If a proposal is accepted, the guest editor will be responsible for:

- Preparing the “Call for Papers” to be included on the Journal’s Web site.
- Distribution of the Call for Papers broadly to various mailing lists and sites.
- Getting submissions, arranging review process, making decisions, and carrying out all correspondence with the authors. Authors should be informed the Instructions for Authors.
- Providing us the completed and approved final versions of the papers formatted in the Journal’s style, together with all authors’ contact information.
- Writing a one- or two-page introductory editorial to be published in the Special Issue.

Special Issue for a Conference/Workshop

A special issue for a Conference/Workshop is usually released in association with the committee members of the Conference/Workshop like general chairs and/or program chairs who are appointed as the Guest Editors of the Special Issue. Special Issue for a Conference/Workshop is typically made of 10 to 15 papers, with each paper 8 to 12 pages of length.

Guest Editors are involved in the following steps in guest-editing a Special Issue based on a Conference/Workshop:

- Selecting a Title for the Special Issue, e.g. “Special Issue: Selected Best Papers of XYZ Conference”.
- Sending us a formal “Letter of Intent” for the Special Issue.
- Creating a “Call for Papers” for the Special Issue, posting it on the conference web site, and publicizing it to the conference attendees. Information about the Journal and Academy Publisher can be included in the Call for Papers.
- Establishing criteria for paper selection/rejections. The papers can be nominated based on multiple criteria, e.g. rank in review process plus the evaluation from the Session Chairs and the feedback from the Conference attendees.
- Selecting and inviting submissions, arranging review process, making decisions, and carrying out all correspondence with the authors. Authors should be informed the Author Instructions. Usually, the Proceedings manuscripts should be expanded and enhanced.
- Providing us the completed and approved final versions of the papers formatted in the Journal’s style, together with all authors’ contact information.
- Writing a one- or two-page introductory editorial to be published in the Special Issue.

More information is available on the web site at http://www.academypublisher.com/jetwi/.

(Contents Continued from Back Cover)