Scalable Faceted Ranking in Tagging Systems

José I. Orlicki[1,2], J. Ignacio Alvarez-Hamelin[2,3], and Pablo I. Fierens[2]

[1] Core Security Technologies, Humboldt 1967 1°, C1414CTU Buenos Aires, Argentina
[2] ITBA, Av. Madero 399, C1106ACD Buenos Aires, Argentina
[3] CONICET (Argentinian Council of Scientific and Technological Research), Argentina
jorlicki@coresecurity.com, ihameli@cnet.fi.uba.ar,
pfierens@itba.edu.ar

Abstract. Nowadays, web collaborative tagging systems which allow users to upload, comment on and recommend contents, are growing. Such systems can be represented as graphs where nodes correspond to users and tagged-links to recommendations. In this paper we analyze the problem of computing a ranking of users with respect to a facet described as a set of tags. A straightforward solution is to compute a PageRank-like algorithm on a facet-related graph, but it is not feasible for online computation. We propose an alternative: (i) a ranking for each tag is computed offline on the basis of tag-related subgraphs; (ii) a faceted order is generated online by merging rankings corresponding to all the tags in the facet. Based on the graph analysis of YouTube and Flickr, we show that step (i) is scalable. We also present efficient algorithms for step (ii), which are evaluated by comparing their results with two gold standards.

Keywords: Web intelligence, Tagging systems, Faceted ranking.

1 Introduction

In collaborative tagging systems, users assign keywords or *tags* to their uploaded content or bookmarks, in order to improve future navigation, filtering or searching, see, e.g., Marlow *et al.* [1]. These systems generate a categorization of content commonly known as a *folksonomy*.

Two well-known collaborative tagging systems for multimedia content are *YouTube* [2] and *Flickr* [3], which are analyzed in this paper. These systems can be represented in a tagged-graph as that shown in Figure 1. In this example, there are four users, *A*, *B*, *C* and *D*. **M** is the set of contents and associated tags. For example, user *B* has uploaded one multimedia content, *song2*, to which it has associated the tag-set $\{$*blues,jazz*$\}$. **V** is the set of recommendations; e.g., user *A* recommends *song2* of user *B*, which is represented in the graph as an arrow from *A* to *B* with tags *blues,jazz*.

Users can be ranked in relation to a set of tags, called a *facet* in this paper. Some applications of *faceted* (i.e., tag-associated) rankings are: (i) searching for content through navigation of the best users inside a facet; (ii) measuring reputation of users by listing their best rankings for different tags or tag-sets.

The order or ranking can be determined by a centrality measure, such as PageRank (see, e.g., Page *et al.* [4] and Langville and Meyer [5]), applied to a recommendation or subscription graph. Given a facet, a straightforward solution is to compute

J. Cordeiro and J. Filipe (Eds.): WEBIST 2009, LNBIP 45, pp. 283–296, 2010.

M = {*(A, song1,* {*blues*}*)*
(B, song2, {*blues,jazz*}*)*
(C, song3,{*blues*}*)*
(C, song4,{*jazz*}*)*
(D, song5,{*blues*}*)*
(D, song6,{*rock*}*)* }

V = {*(A, song2)*
(B, song4)
(B, song5)
(A, song3)
(A, song4)
(C, song6) }

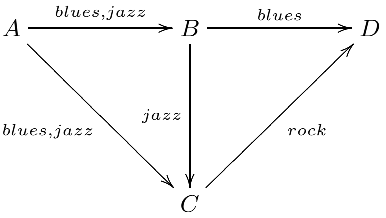

Fig. 1. Example of construction of a tagged graph from a set of contents **M** and a set of recommendations **V**

the centrality measure based on an appropriate facet-dependent subgraph of the recommendation graph. However, *online* computation of the centrality measure is unfeasible because its high time complexity, even for small facets with two or three tags. Moreover, the offline computation of the centrality measure for each facet is also unfeasible because the number of possible facets is exponential in the number of tags. Therefore, alternative solutions must be looked for. A simple solution is to use a general ranking computed offline, which is then filtered online for each facet-related query. The use of a single ranking of web pages or users within folksonomies has the disadvantage that the best ranked ones are those having the highest centrality in a global ranking which is facet-independent. In the information retrieval case, this implies that the returned results are ordered in a way which does not take into account the focus on the searched topic. Richardson and Domingos [6] call this problem *topic drift*.

In this paper we propose a solution to the problem of topic drift in faceted rankings which is based on PageRank as centrality measure. Our approach follows a two-step procedure: (i) a ranking for each tag is computed offline on the basis of a tag-related subgraph; (ii) a faceted order is generated online by merging rankings corresponding to all the tags in the facet. The fundamental assumption is that step (i) in this procedure can be computed with an acceptable overhead which depends on the size of the dataset. This hypothesis is validated by two empirical observations. On one hand, in the studied tagged-graphs of *Flickr* and *YouTube*, most of the tags are associated to very small subgraphs, while only a small number of tags have large associated subgraphs (see Sect. 3). On the other hand, the mean number of tags per edge is finite and small as explained in Sect. 4.1.

The problem then becomes to find a *good* and *efficient* algorithm to merge several rankings in step (ii), and we present several alternatives in Sect. 4. The *"goodness"* of a merging algorithm is measured by comparing its results to those produced by the naive approach of applying the PageRank algorithm on facet-dependent graphs (see Sect. 5). The *efficiency* of an algorithm is evaluated by means of its time complexity.

We concentrate our effort on facets that correspond to the *logical conjunction* of tags (*match-all-tags*-queries) because this is the most used logical combination in information retrieval (see Manning *et al.* [7]).

The remaining of the paper is organized as follows. We discuss prior works and their limitations in Sect. 2. In Sect. 3 we explore two real examples of tagged graphs. In particular, we analyze several important characteristics of these graphs, such as the scale-free behavior of the node indegree and assortativeness of the embedded recommendation graph (see Sect. 3.1). The proposed algorithms are introduced in Sect. 4, including an analysis of related scalability issues in Sect. 4.1. We discuss experimental results in Sect. 5 and we conclude with some final remarks and possible directions of future work in Sect. 6.

2 Related Work

There is abundant literature on faceted search. For example, basic topic-sensitive PageRank analysis was attempted biasing the general PageRank equation to special subsets of web pages by Al-Saffar and Heileman in [8], and using a predefined set of categories by Haveliwala in [9]. Although encouraging results were obtained in both works, they suffer from the limitation of a fixed number of topics biasing the rankings. The approach of DeLong *et al.* [10] involves the construction of a larger multigraph using the hyperlink graph with each node corresponding to a pair webpage-concept and each edge to a hyperlink associated with a concept. Although DeLong *et al.* obtain good ranking results for single-keyword facets, they do not support multi-keyword queries.

Query-dependent PageRank calculation was introduced by Richardson and Domingos in [6] to extract a weighted probability per keyword for each webpage. These probabilities are summed up to generate a query-dependent result. They also show that this faceted ranking has, for thousands of keywords, computation and storage requirements that are only approximately 100-200 times greater than that of a single query-independent PageRank. As we show in Sect. 4.1, the *offline* phase of our facet-dependent ranking algorithms has similar time complexity. Scalability issues were also tackled by Jeh and Widom [11] criticizing offline computation of multiple PageRank vectors for each possible query and preferring another more efficient dynamic programming algorithm for online calculation of the faceted rankings based on offline computation of basis vectors. They found that their algorithm scales well with the size of the biasing page set.

In this paper, we propose a different alternative to the problem of faceted ranking. Instead of computing offline the rankings corresponding to all possible facets, our solution requires only the offline computation of a ranking per tag. A faceted ranking is generated by adequately merging the rankings of the corresponding tags. Sect. 4 deals with different approaches to the merging step.

Some search engines incorporate tags from collaborative tagging systems. For example, Technorati [12] improves its search engine using tags entered by bloggers and Web page designers, providing also citation count ranking information (called "Authority") (see Weinman [13]). Tags have also been used to leverage topic-dependent ranking of egos. For example, John and Seligmann [14] explore the application of tagging systems to rank expertise of taggers in enterprise environments. Basically, they use a variation of PageRank biased on the basis of the facet-related tagging activity of each user. Yeung *et al.* [15] also rank the expertise of taggers. They use a variation of HITS algorithm (see Kleinberg [16]) that takes into account the time when a user applied tags to a given document, giving a higher ranking to those users which "discovered" the content first. Neither of these approaches is scalable because they require a new costly computation (PageRank in the case of [14] and HITS for [15]) for each facet.

Hotho *et al.* [17] modified PageRank (called FolkRank) to work on a tripartite graph corresponding to a folksonomy. The set of nodes of the graph consists of the union of users, tags and resources (contents). All co-occurrences of tags and users, users and resources, tags and resources become edges. Furthermore, edges are weighted with the frequency of appearance of the represented relation. One problem with FolkRank is that it works with symmetric adjacency matrices (corresponding to tags and users, users and resources, tags and resources) and, hence, weights "bounce back" during each iteration of the algorithm[1]. Shepitsen and Tomuro [18] solved this problem by slightly modifying the adjacency matrices, e.g., they multiplied the weight of the edge connecting a tag t to a content c by the fraction of users who used t for c over all user s who used tag t. In order to allow topic-dependent rankings, Hotho and colleagues introduced a form of topic-biasing on the modified PageRank. The problem with this approach is that the generation of a faceted ranking requires a new computation of the adapted algorithm on the graph for each new facet.

Social search (see, e.g., Long *et al.* [19]) consists in accounting for the social interactions or contributions of other users when answering a query from a given user. In this sense, there has been some work on accounting for the tagging activity of other users. For example, Zanardi and Capria [20] study the problem of ranking tagged documents for query answer presentation in this context. After expanding the query with related tags (based on a cosine similarity metric), they produce a ranking of content based on the relevance of tags associated to each document the similarity of the document taggers to the user who made the query. The similarity between users is based on past tagging activity. Schenkel *et al.* [21] also take into account the strength of similarity of users tagging activity and the relatedness of tags (although with different metrics), but focus on an efficient algorithm to compute the top-k results of a query. A more complete approach is presented by Symeonidis *et al.* [22], where users, tags, and resources are modelled in a 3-order tensor and content is recommended to users by performing Latent Semantic Analysis and dimensionality reduction using the Higher Order Singular Value Decomposition technique. In this work, we do not address the problem of user-personalization of queries or recommendations and we focus on producing user independent rankings.

3 Two Tagging Systems: YouTube and Flickr

In this section, we present two examples of collaborative tagging systems,*YouTube* and *Flickr*, where content is tagged and recommendations are made. Although these systems actually rank content, to our knowledge, no use of graph-based faceted ranking is made.

[1] PageRank works with a directed graph and, hence, a node "passes on" weight only on outgoing links.

The taxonomy of tagging systems by Marlow *et al.* in [1] allows us to classify *YouTube* and *Flickr* in the following ways: regarding the tagging rights, both are *self-tagging* systems; with respect to the aggregation model, they are *set* systems; concerning the object-type, they are called *non-textual* systems; in what respects to the source of material, they are classified as *user-contributed*; finally, regarding tagging support, while *YouTube* can be classified as a *suggested* tagging system, *Flickr* must be considered a *blind* tagging system.

In our first example the content is multimedia in the form of favorite videos recommended by users. The information was collected from the service *YouTube* using the public API crawling 185,414 edges and 50,949 nodes in Breadth-First Search (BFS) order starting from the popular user *jcl5m* that had videos included in the top twenty top rated videos during April 2008. We only considered nodes with indegree greater than one, because they are the relevant nodes to PageRank. ¿From this information, we constructed a full tagged graph G. We have also constructed subgraphs by preserving only those edges that contain a given tag (e.g., $G(music)$ and $G(funny)$ corresponding to the tags $music$ and $funny$, respectively), any tag in a set (e.g., $G(music \vee funny)$) or all tags in a set (e.g., $G(music \wedge funny)$). Table 1 presents the number of nodes and edges of each of these graphs. We must note that mandatory categorical tags such as $Entertainment$, $Sports$ or $Music$, always capitalized, were removed in order to include only tags inserted by users.

Table 1. Sizes of *YouTube*, *Flickr* graphs and some of their subgraphs

YouTube	nodes	edges	*Flickr*	nodes	edges
G	50,949	185,414	G	30,974	225,650
$G(music \vee funny)$	4,990	13,662	$G(blue \vee flower)$	5,440	14,273
$G(music)$	2,650	5,046	$G(blue)$	3,716	6,816
$G(funny)$	2,803	6,289	$G(flower)$	2,771	6,370
$G(music \wedge funny)$	169	188	$G(blue \wedge flower)$	280	327

In our second example the content are photos and the recommendations are in the form of favorite photos[2]. The information was collected from the service *Flickr* using the public API crawling 225,650 edges and 30,974 nodes in BFS order starting from the popular user *junku-newcleus*, where only nodes with indegree grater than one have been considered. The full tagged graph G and the sample subgraphs $G(blue \vee flower)$, $G(blue)$, $G(flower)$ and $G(blue \wedge flower)$ were constructed. The number of nodes and edges of these graphs are shown in Table 1.

3.1 Analysis of the Recommendation Graphs

In this section, we present a summary of the graph analysis of both collaborative tagging systems, *YouTube* and *Flickr*. The objective is to disentangle the properties of these systems in order to take advantage by new ranking algorithms.

[2] Only the first fifty favorites photos of each user were retrieved.

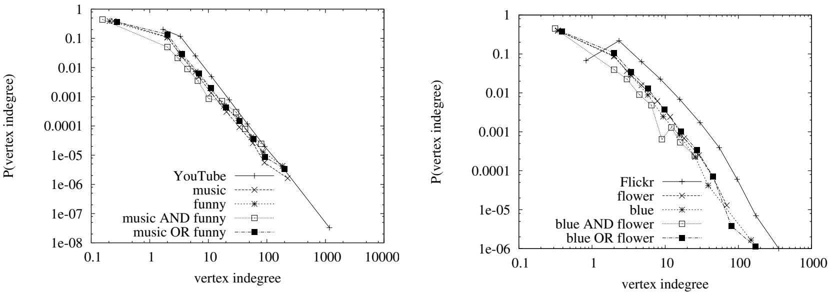

Fig. 2. Binned indegree distribution

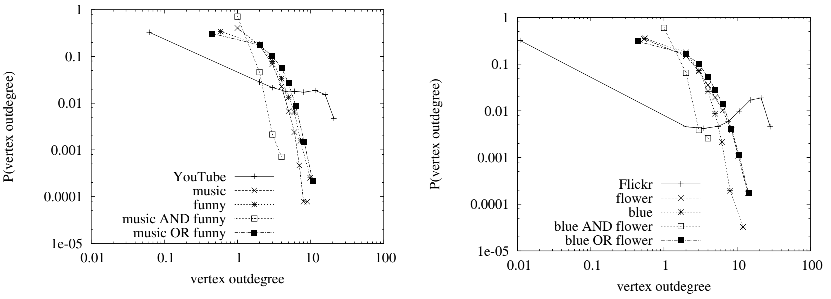

Fig. 3. Binned outdegree distribution

Graph analysis was made using the tool Network Workbench [23], except for the calculation of PageRank (a C++ stand-alone software). We computed node's indegree and outdegree distributions, the correlation of indegree of in-neighbors with indegree of nodes, the PageRank distribution, and the number of tags per edge. Figures 2-6 show the parameters for the whole *YouTube* and *Flickr* graphs and also for some specific subgraphs. All graph-analytical parameters, except those for small subgraphs like $G(music \wedge funny)$ were binned and plotted in log-log curves[3].

Node indegree, in both video and photo graphs, can be bounded by a power-law distribution: $P(k) \approx k^{-\gamma}$, where $2 < \gamma < 3$ (see Figure 2). Random variables modelled by this type of heavy-tailed distributions have a finite mean, but infinite second and higher non-central moments. Furthermore, there is a non-vanishing probability of finding a node with an arbitrary high indegree. Clearly, in any real-world graph, the total number of nodes is a natural upper-bound to the greatest possible indegree. However, experience with Internet related graphs shows that the power-law distribution of the indegree does not change significantly as the graph grows and, hence, the probability of finding a node with an arbitrary degree eventually becomes non-zero (for more details see, e.g., Pastor-Satorras and Vespignani [24]).

[3] This is the reason why some degree points appear below one (x-axis), because there exist nodes with either indegree or outdegree equal to zero.

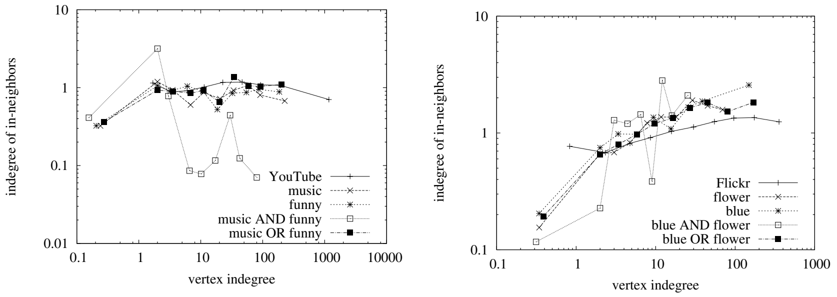

Fig. 4. Binned correlation of indegree of in-neighbors with indegree

Since recommendation lists are made by individual users, node outdegree does not show the same kind of scale-free behavior than node indegree. On the contrary, each user recommends only 20 to 30 other users on average (see Figure 3). Moreover, since node outdegree is mostly controlled by individual *human* users, we do not expect its average to change significantly as the graph grows. Another way of looking at the human origin of tag annotations is by observing Fig. 6, which shows that only few edges have many tags and the number of edges per tag falls off quickly after approximately ten.

The correlation of indegree of in-neighbors with node indegree (see Figure 4) indicates the existence of assortative (positive slope) or disassortative behavior (negative slope). Assortativeness is commonly observed in social graphs, where people with many connections relates to people which is also well-connected. Disassortativeness is more common in other kinds of graphs, such as information, technological and biological graphs (see, e.g., Newman [25]). In *YouTube*'s graph there is no clear correlation (small or no slope), but in *Flickr*'s graph there is a slight assortativeness indicating a biased preference of nodes with high indegree for nodes with high indegree (see Figure 4).

We also computed the PageRank of the sample graphs, removing dangling nodes with indegree 1 and out degree 0, because most of them correspond to nodes which have not been visited by the crawler (BFS), having the lowest PageRank (a similar approach is taken by Page *et al.* in [4]). Figure 5 shows that PageRank distributions are also scale-free, i.e., they can be bounded by power law distributions. Note that the power law exponents are very similar for the complete tagged graph and subgraphs, on each graph, showing the same behavior. This fact suggests that it is possible to get a good faceted ranking by combining the ranking of each tag in the facet.

4 Algorithms for Faceted Ranking

Given a set **M** of tagged content, a set **V** of favorite recommendations and a tag-set or facet **F**, the *faceted ranking problem* consists on computing a ranking of users according to facet **F**. The naive solution is to find a graph associated to the facet and apply the PageRank algorithm to it. This approach leads to two algorithms, called **edge-intersection** and **node-intersection**, which turn out to be too costly for online queries. Indeed, their computation requires the extraction of a subgraph which might be very

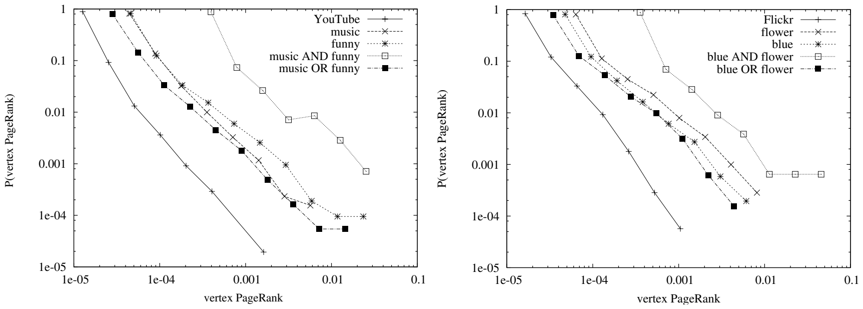

Fig. 5. Binned Node PageRank distribution

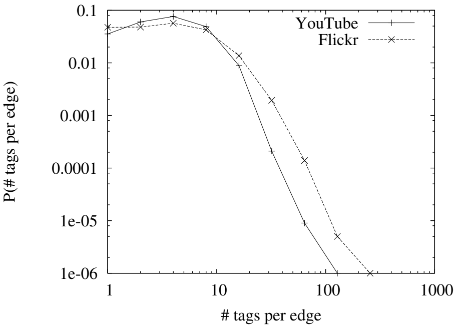

Fig. 6. Binned distribution of number of tags per edge

large in a large graph[4] and the calculation of the corresponding PageRank vector. However, they serve as a basis of comparison (i.e., "gold standards") for more efficient algorithms.

We define the following algorithms for *match-all-tags*-queries.

Edge-intersection. Given a set of tags, a ranking is calculated by computing the centrality measure of the subgraph corresponding to the recommendation edges which include *all* the tags of a certain facet.

Node-intersection. Consider the example given in Fig. 1 under the query $blues \wedge rock$. According to the **edge-intersection** algorithm, there is no node in the graph satisfying the query. However, it is reasonable to return node D as a response to such search. In order to take into account this case, we devised another algorithm called **node-intersection**. In this case, the union of all edge recommendations per tag is used when computing the PageRank, but only those *nodes* involved in recommendations for *all tags* are kept (hence, *node-intersection*). This is another possible way to obtain a subgraph having only a specific tag-related information.

[4] We have observed that as the graph grows the relative frequency of tags usage converges. Similar behavior was observed for particular resources by Golder and Huberman in [26].

Single-ranking. A simple online faceted ranking consists of a monolithic ranking of the full graph, without any consideration of tags, which is then filtered to exclude those nodes that are not related to all tags in the facet.

Winners-intersection. In this case, as well as in the next two algorithms, the *offline phase* consists of running PageRank on each tag-related subgraph (such as $G(music)$) and storing only the best-w ranked users. Given a conjunction-of-tags facet, a new graph is constructed by considering only the w "winners" corresponding to each tag and the edges connecting them. A facet-related ranking is then calculated by means of the PageRank algorithm on this reduced graph. The choice of an adequate number w is application-dependent. For this paper, we have arbitrarily chosen $w = 128$. We shall show that reasonably good results are obtained even for this small value of w.

Probability-product. Let us recall that PageRank is based on the idea of a random websurfer and nodes are ranked according to the estimated stationary probability of such a surfer being at each node at any given time. This basic concept together with the product rule for the joint probability of independent events motivated the **probability-product** algorithm. This algorithm pre-computes a PageRank for each tag-related subgraph. A ranking associated with a conjunction-of-tags facet is then calculated *online* by multiplying, on a node-by-node basis, the probabilities corresponding to each tag in the facet.

Rank-sum. Consider a recommendation graph G larger than that in Fig. 1 and the query $blues \wedge jazz$. Assume that the PageRank of the top three nodes in the rankings corresponding the subgraphs $G(blues)$ and $G(jazz)$ are as given in Table 2. Ignoring other nodes, the ranking given by the **probability-product** rule is a, b and c. However, it may be argued that node b shows a better equilibrium of PageRank values than node a. Intuitively, one may feel inclined to rank b over a given the values in the table. In order to follow this intuition, we devised the **rank-sum** algorithm which is also intended to avoid *topic drift* within a queried facet, that is, any tag prevailing over the others. Given a conjunction-of-tags facet, the **rank-sum** algorithm adds-up the ranking position of nodes in each tag-related subgraph. The corresponding facet-related ranking is calculated by ordering the resulting sums (see Table 2).

Table 2. Probability-product vs. **rank-sum** in an example

Node	$G(blues)$	$G(jazz)$	Prob.-pr.	Rank-sum
a	0.75	0.04	0.03	4
b	0.1	0.1	0.01	3
c	0.01	0.03	0.003	6
d	0.01	0.05	0.005	7

The first two columns show the probability of each node according to PageRank on the corresponding tag-related subgraph.

4.1 Complexity Analysis of the Algorithms

As noticed by Langville and Meyer in [5], the number of iterations of PageRank is fixed when both the tolerated error and other parameters are fixed. As each iteration consists of a sparse adjacency matrix multiplication, the time complexity of PageRank is linear on the number of edges of the graph. Since **probability-product**, **rank-sum** and **winners-intersection** algorithms require the offline computation of PageRank for each tag-related subgraph, it is clear that, if the average number of tags per edge is constant or grows very slowly as the graph grows, then the *offline phase* of these algorithms is scalable, i.e., linear on the number of edges of the complete tagged graph. Fig. 6 shows that the distribution of tags per edge falls quickly, having a mean of ~ 9 tags per edge for the *YouTube* tagged graph and ~ 13 for the *Flickr* tagged graph. These are *not* heavy-tailed distributions and, since tags are manually added to each uploaded content, we do not expect the average number of tags per recommendation to increase significantly as the graph grows. In other words, Fig. 6 validates the hypothesis on which the scalability of the *offline phase* the **probability-product**, **rank-sum** and **winners-intersection** algorithms.

The time complexity of **edge-intersection** algorithm can be estimated by decomposing it into three phases. The first step is to find the graph associated to a facet with k tags, which takes $\mathcal{O}(k \cdot E_{tag})$, where E_{tag} is the number of edges of largest tag-related subgraph. The second step is to compute the PageRank of the resulting graph, taking $\mathcal{O}(E_{facet})$ time, where E_{facet} is the number of edges of the graph and, clearly, $E_{facet} \leq k \cdot E_{tag}$. Finally, the list of N_{facet} nodes of the graph must be ordered according to the PageRank, taking $\mathcal{O}(N_{facet} \log(N_{facet}))$. We have found that N_{facet} is, in general, much smaller than E_{facet} (e.g., see Table 1).

For the **node-intersection** algorithm, the time complexity is the same that in the former case, but E_{facet} and N_{facet} are usually larger because the graph includes more edges.

In the case of **single-ranking**, the *online* computation takes $\mathcal{O}(k \cdot N_{tag})$ time, where N_{tag} is the maximum number nodes of a tag-related subgraph. Indeed, assuming that there is an ordered list of nodes related to each tag, its (ordered) intersection can be computed in time that grows linearly with the sum of the lengths of the lists.

The **winners-intersection**, **probability-product** and **rank-sum** algorithms have the same time complexity, $\mathcal{O}(k)$, because they only consider the top-w ranked users of each tag in the facet and, hence, their complexity depends only on the number of tags (i.e, the number of operations for each tag is fixed by the constant w). This remark is supported because top-w is widely used in the most popular search engines.

5 Experimental Results and Discussion

In this section, we compare the behavior of the algorithms presented in Sect. 4. As a basis of comparison we use two algorithms whose online computation is unfeasible, but which are reasonably good standards: **edge-intersection** and **node-intersection**.

In order to quantify the "distance" between the results given by two different algorithms, we use two ranking similarity measures, *OSim* (Haveliwala [9]) and *KSim* (Kendall [27] and Haveliwala [9]). The first measure, $OSim(R_1, R_2)$ indicates the degree of overlap between the top n elements of rankings R_1 and R_2, i.e., the number of common nodes. The second measure, $KSim(R_1, R_2)$ is a variant of Kendall's distance that considers the relative orderings, i.e., counts how many inversions are in a fixed top set. In both cases, values closer to 0 mean that the results are not similar and closer to 1 mean the opposite.

We have run our algorithms on all facets of tag pairs extracted from the 100 most used tags[5] in each of the graphs, *YouTube* and *Flickr*. For each tag pair the proposed ranking algorithms (**single-ranking**, **probability-product**, **rank-sum** and **winners-intersection**) were compared to the gold-standards (**edge-intersection** and **node-intersection**) using $OSim$ and $KSim$ to measure the rankings' similarity.

Tables 3-4 present a summary of the comparisons, where we display averaged similarities for different sizes of top-sets of ranked users. Figures 7 and 8 show a more detailed summary of results for the $OSim$ metric for the proposed algorithms. We do not include gray-scale figures corresponding to the *YouTube* graph because they are very similar. The x-axis in the figures corresponds to the number of nodes resulting from the basis of comparison algorithm (**edge-intersection** or **node-intersection**) and the y-axis to the top number n of nodes used to compute the similarities. The similarity results of $OSim$ were averaged for log-log ranges. Observe that darker tones correspond to values closer to 1, i.e., more similar results. White spaces correspond to cases for which there are no data, e.g., whenever the y coordinate is greater than intersection size.

Table 3. *YouTube*: Comparison of ranking algorithms (average similarity: OSim|KSim)

Algorithm	**edge-intersection**			**node-intersection**		
	top 8	top 16	top 32	top 8	top 16	top 32
Single-ranking	0.08|0.48	0.10|0.50	0.13|0.51	0.31|0.53	0.34|0.55	0.39|0.56
Winners-inters.	0.06|0.48	0.04|0.49	0.04|0.50	0.10|0.49	0.08|0.50	0.08|0.51
Prob.-product	**0.72**|**0.71**	**0.80**|**0.78**	**0.86**|**0.83**	**0.42**|**0.59**	**0.52**|**0.66**	**0.67**|**0.74**
Rank-sum	**0.73**|**0.72**	**0.81**|**0.79**	**0.86**|**0.84**	**0.41**|**0.58**	**0.50**|**0.64**	**0.67**|**0.72**

Table 4. *Flickr*: Comparison of ranking algorithms (average similarity: OSim|KSim)

Algorithm	**edge-intersection**			**node-intersection**		
	top 8	top 16	top 32	top 8	top 16	top 32
Single-ranking	0.07|0.48	0.09|0.49	0.11|0.50	0.17|0.50	0.21|0.51	0.27|0.53
Winners-inters.	0.30|0.53	0.23|0.53	0.11|0.51	0.19|0.50	0.19|0.52	0.18|0.53
Prob.-product	**0.57**|**0.63**	**0.64**|**0.66**	0.72|**0.71**	**0.32**|**0.55**	**0.42**|**0.59**	**0.56**|**0.67**
Rank-sum	**0.57**|**0.63**	**0.64**|**0.67**	**0.72**|**0.72**	**0.31**|**0.53**	**0.41**|**0.58**	**0.56**|**0.66**

As can be appreciated from Tables 3-4 and Figures 7-8, the **single-ranking** algorithm gave the worst results in most cases.

The **winners-intersection** algorithm, which is based on retaining only the 128 top-ranked users for each tag, gives worse results than **probability-product** and **rank-sum**, even for smaller intersections. This fact is explained by the relevance of a large number of recommendations of low-ranked users when computing the PageRank in both the **edge-intersection** and the **node-intersection** cases. Also note that the **winners-intersection** approach gave better results for *Flickr* than for *YouTube*. A possible cause

[5] Some tags like *you*, *video* or *youtube* which give no information were removed from the experiment.

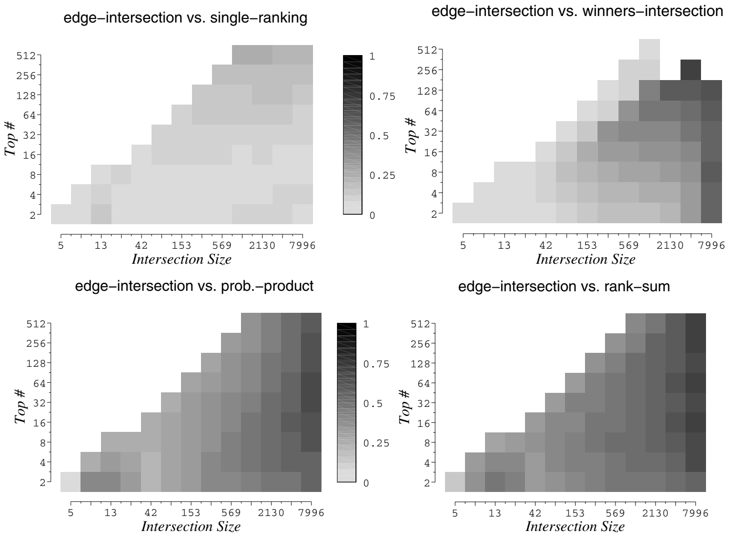

Fig. 7. Flickr graph: Average similarity to **Edge-intersection**

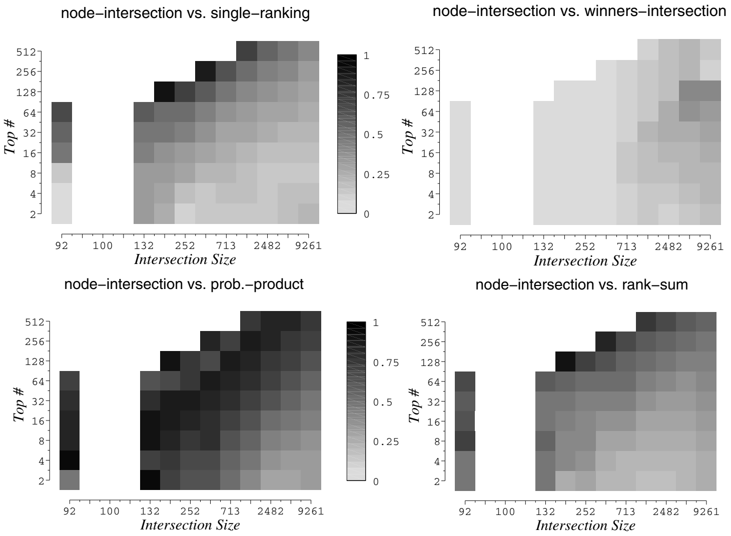

Fig. 8. Flickr graph: Average similarity to **Node-intersection**

is the assortativeness of *Flickr*'s graph (see Sect. 3.1). Indeed, since assortativeness implies that users with many recommendations are preferentially recommended by users with also many recommendations, the relevance of low-ranked users in the computation of the centrality measure is lower.

The **probability-product** and **rank-sum** algorithms exhibit a similar behavior and clearly outperform other ranking algorithms when considering the similarity to the **edge-intersection** and the **node-intersection** standards on both graphs.

It is also remarkable that the similarity increases more rapidly (as the size of the ranking increases) in the cases of **probability-product** and **rank-sum** algorithms than in the of single ranking (see Tables 3-4 and Figures 7-8). On the contrary, in the case of **winners-intersection** algorithm, the Osim metric decreases. This is due to the limitation on the number of winners w considered.

6 Conclusions and Future Work

We have proposed different algorithms for merging tag-related rankings into complete faceted-rankings of users in collaborative tagging systems. In particular, two of our algorithms, **probability-product** and **rank-sum** are feasible for online computation and give results comparable to those of two reasonable, though computationally costly, standards.

A prototypic application called *Egg-O-Matic* is available online [28]. It includes the online method **rank-sum** to approximate the **edge-intersection** offline ranking (in a mode called *"all tags, same content"*), and the **probability-product** method to approximate the **node-intersection** ranking (in a mode called *"all tags, any content"*).

A matter of future research is the possibility of reducing the complexity of the offline phase of the proposed algorithms by first clustering the tags. In this case, facets would be represented by a set of clusters of related tags.

This work also opens the path for a more complex comparison of reputations, for example by integrating the best positions of a user even if the tags involved are not related (*disjunctive* queries) in order to summarize the relevance of a user generating content on the web.

It is also possible to extend the **rank-sum** algorithm in Sect. 4 to merge rankings generated from different systems or search engines, producing a ranking from a broader view (*cross-system ranking*).

References

1. Marlow, C., Naaman, M., Boyd, D., Davis, M.: HT06, tagging paper, taxonomy, Flickr, academic article, to read. In: HYPERTEXT 2006: Proc. of the seventeenth conference on Hypertext and hypermedia, pp. 31–40. ACM Press, New York (2006)
2. YouTube (2008), `http://www.youtube.com/`
3. Flickr (2008), `http://www.flickr.com/`
4. Page, L., Brin, S., Motwani, R., Winograd, T.: The PageRank Citation Ranking: Bringing Order to the Web. Technical report, Stanford Digital Library Technologies Project (1998)
5. Langville, A.N., Meyer, C.D.: Survey: Deeper Inside PageRank. Internet Mathematics 1(3) (2003)
6. Richardson, M., Domingos, P.: The Intelligent Surfer: Probabilistic Combination of Link and Content Information in PageRank. In: Advances in Neural Information Processing Systems 14. MIT Press, Cambridge (2002)

7. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press, Cambridge (2008)
8. Al-Saffar, S., Heileman, G.: Experimental Bounds on the Usefulness of Personalized and Topic-Sensitive PageRank. In: WI 2007: Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence, Washington, DC, USA, pp. 671–675. IEEE Computer Society, Los Alamitos (2007)
9. Haveliwala, T.H.: Topic-sensitive PageRank. In: Proc. of the Eleventh International World Wide Web Conference, Honolulu, Hawaii (May 2002)
10. DeLong, C., Mane, S., Srivastava, J.: Concept-Aware Ranking: Teaching an Old Graph New Moves. Icdmw, 80–88 (2006)
11. Jeh, G., Widom, J.: Scaling personalized web search. Technical report, Stanford University (2002)
12. Technorati, `http://technorati.com`
13. Weinman, J.: A new approach to search. Business Communications Review (October 2007)
14. John, A., Seligmann, D.: Collaborative tagging and expertise in the enterprise. In: 15th International Conference on the World Wide Web (2006)
15. Yeung, C.A., Noll, M.G., Gibbins, N., Meinel, C., Shadbolt, N.: On measuring expertise in collaborative tagging systems. In: Proceedings of the WebSci 2009: Society On-Line, Athens, Greece (March 2009)
16. Kleinberg, J.M.: Authoritative sources in a hyperlinked environment. Journal of the ACM 46(5), 604–632 (1999)
17. Hotho, A., Jäschke, R., Schmitz, C., Stumme, G.: Information Retrieval in Folksonomies: Search and Ranking. In: Sure, Y., Domingue, J. (eds.) ESWC 2006. LNCS, vol. 4011, pp. 411–426. Springer, Heidelberg (2006)
18. Shepitsen, A., Tomuro, N.: Search in social tagging systems using ontological user profiles. In: 3rd International AAAI Conference on Weblogs and Social Media (ICWSM 2009), San Jose, California, USA, Association for the Advancement of Artificial Intelligence (AAAI) (May 2009)
19. Longo, L., Barret, S., Dondio, P.: Toward social search - from explicit to implicit collaboration to predict users' interests. In: 5th International Conference on Web Information Systems and Technologies (WEBIST 2009), Lisboa, Portugal, March 2009, pp. 693–696 (2009)
20. Zanardi, V., Capra, L.: Social ranking: uncovering relevant content using tag-based recommender systems. In: RecSys 2008: Proceedings of the 2008 ACM conference on Recommender systems, pp. 51–58. ACM, New York (2008)
21. Schenkel, R., Crecelius, T., Kacimi, M., Michel, S., Neumann, T., Parreira, J.X., Weikum, G.: Efficient top-k querying over social-tagging networks. In: SIGIR 2008: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval, pp. 523–530. ACM, New York (2008)
22. Symeonidis, P., Ruxanda, M., Nanopoulos, A., Manolopoulos, Y.: Ternary semantic analysis of social tags for personalized music recommendation. In: Proceedings of 9th International Conference on Music Information Retrieval, Philadelphia, USA (2008)
23. NWB Team, Network Workbench Tool (2006)
24. Pastor-Satorras, R., Vespignani, A.: Evolution and structure of the Internet: A statistical physics approach. Cambridge University Press, Cambridge (2004)
25. Newman, M.E.J.: Assortative Mixing in Networks. Phys. Rev. Lett. 89(20), 208701 (2002)
26. Golder, S., Huberman, B.A.: Usage patterns of collaborative tagging systems. Journal of Information Science 32(2), 198–208 (2006)
27. Kendall, M.G.: A New Measure of Rank Correlation. Biometrika 30(1/2), 81–93 (1938)
28. Egg-O-Matic (2008), `http://eggomatic.itba.edu.ar/`