Collaboration-based Social Tag Prediction in the Graph of Annotated Web Pages

Hossein Rahmani[1], Behrooz Nobakht[1], and Hendrik Blockeel[1,2]

[1] Leiden Institute of Advanced Computer Science, Universiteit Leiden,
Niels Bohrweg 1, 2333 CA Leiden, The Netherlands
hrahmani@liacs.nl,bnobakht@liacs.nl,blockeel@liacs.nl
[2] Department of Computer Science, Katholieke Universiteit Leuven,
Celestijnenlaan 200A, 3001 Leuven, Belgium

Abstract. *Different approaches based on content or tag information have been proposed to address the problem of tag recommendation for a web page. In this paper, we analyze two approaches in a graph of web pages. Each node is a web page and edges represent hyperlinks. The first approach uses the content while the second one uses tag information in the graph. The second approach makes two assumptions about the tag set of two interacting nodes. The Tag Similarity Assumption claims that two interacting nodes discuss about rather similar topics; therefore, the chance of having more similar tag set is higher. The Tag Collaboration Assumption says that two interacting nodes complement each others topics. We apply algorithms such as Self Organizing Map (SOM), Reinforcement Learning (RL) and K-means clustering to compare methods on several datasets. We conclude that tag-based tag predictors outperform their content-based peers by more than ten percent with respect to the cosine similarity between predicted and actual tag sets.*

Key words: Social Tag Prediction, Tag Similarity Assumption, Tag Collaboration Assumption

1 Introduction

Social tagging systems have become increasingly popular in recent years. Various domain applications such as search engines, recommendation systems, spam detection and many others have improved their performance by considering these types of information.

Although using user's meta data has been shown useful for different purposes, these new types of describing resources have their own problems. The uncontrolled vocabulary nature of these data has inherent ambiguity and may make it hard to get a coherent vision of different users.

One possible way to solve the problems is to help users choose better informative tags. The system could suggest some "good tags" to users and users may or may not use those tags. In this paper, we proposed two different approaches for the task of social tag prediction in a graph of web pages. For each approach there are different implementations. The first approach uses only the content of the web pages for the task of tag prediction. The second approach considers two assumptions about the tag set of two interacting web pages in a graph. The first assumption says that two interacting web pages have similar tag sets (Tag Similarity Assumption) while the second one assumes that two interacting web pages have "collaborative" tag sets (Tag Collaboration Assumption), collaborative meaning in this case that the tag sets somehow complement each other.

The rest of the paper is organized as follows: Section 2 briefly reviews motivations and related works. Section 3 discusses our approach for social tag prediction. In Section 4, first, we tune the parameters of our methods and then, we present experimental results on five web page graphs. We present our conclusions in Section 5.

2 Related Works

Various works have explored social annotations for different purposes. A lot of methods designed to improve web search results by considering data from social bookmarking systems ([1, 2]). Social annotation could be seen as a new way of organizing information and categorizing resources ([3, 4]). Users of social tagging systems could be connected to each other based on their areas of interests. The term Folksonomy, a combination of folk and taxonomy, was first proposed by [5]. A general introduction of folksonomy could be found in [6]. Ranking and recommender systems for folksonomies are proposed in [7, 8]. While the above mentioned approaches might look similar to our work but they are actually orthogonal, since the authors do not directly address the problem of predicting tags for resources.

The approaches in [9, 10] predicts tags based on the content and tag cooccurance respectively, while our methods consider also the neighborhood context of each web page for the tag prediction task. [11] proposes neighborhood-based tag prediction by using content similarity. They apply a straightforward scoring model to select the candidate tags, however, we use machine learning methods for tag prediction. [12] predicts tags from the set of 100 most frequent tags found in del.icio.us by training the binary classifier for each tag. We do not restrict ourselves to a predefined set of tags which are predicted.

Another important difference between our method and [11, 12] is that our method predicts the set of tags without considering any number of known tags for a given web page, whereas the methods in [11] and [12] are more "Social Tag Expansion" methods; they start with some known tags for a web page and then try to expand that set. In other words, we do not assume any knowledge about the document's own tags, in contrast to [11] and [12].

We introduce a new assumption "Tag Collaboration", which to our knowledge is not discussed in any previous work. We also define the "Topic Locality" characteristic of a web page graph, which is shown to affect the optimal parameter settings and the performance of the approach. These novelties sets aside our methods from the related works.

3 Proposed Methods

The graph of web pages is represented by $G(P, E)$ in which P is a set of web pages and E is a set of interactions between web pages. Each $e_{pq} \in E$ shows a hyperlink between two web pages $p \in P$ and $q \in P$. Let T be the set of all the tags that occur in one dataset. Each classified web page $p \in P$ is annotated with a $|T|$-dimensional vector TS_p that indicates the tag set of this web page: $TS_p(t_i)$ is 1 if $t_i \in T$ is assigned to the web page p, and 0 otherwise. TS_p can also be seen as the set of all tags t_i for which $TS_p(t_i) = 1$. Similarly, the $|T|$-dimensional vector NB_p describes how often each tag occurs in the neighborhood (all the web pages that are reachable from p with a path of length at most 1) of web page p. $NB_p(t_i) = n$ means that among all the web pages that interact with p, n are annotated with tag t_i.

In this section, we discuss different approaches for solving the problem of social tag prediction in a graph of web pages. We analyze two main approaches for this purpose. The first approach uses the content while the second one uses tag information in the graph.

3.1 Content-Based Tag Predictor

In this approach, we use only the content of a web page for predicting its tags. We use the standard bag-of-words approach: after stemming and stop word removal, a vector is constructed in which each component is the frequency with which a particular word occurs on the page.

Most similar: Our first implementation for this approach is a standard nearest neighbor approach, which we refer to as "Most similar". In this implementation, first, we compare the content of the unannotated web page with the content of all the annotated web pages in the graph, using cosine similarity (Formula 1). After finding the most similar annotated web page, we select the top tags of the annotated web page and assign them to the unannotated one.

$$\textit{Cosine similarity}\ (p, q) = \frac{p.q}{|p| * |q|} \tag{1}$$

K-means: In this approach, we first cluster the web pages in the graph based on their content. Then, we find the most frequently occurring tags in each cluster. The popular (most frequent) tags of each cluster will be assigned to all the unannotated web pages in that cluster.

We use K-means [13] for clustering the graph of web pages. We start clustering the network with k random centers and iteratively assign the web pages to these k clusters based on the content similarity between web pages and the cluster centers. In each iteration, cluster centers move towards more balanced points in the cluster. As similarity metrics we use both Jaccard similarity (Formula 2) and Cosine similarity (Formula 1) of the tag sets.

$$Jaccard\ similarity\ (p, q) = \frac{|p \cap q|}{|p \cup q|} \tag{2}$$

3.2 Tag-Based Tag Predictors

Contrary to the previous approach, here, we use only tag information available in the graph for the task of tag prediction. We consider two assumptions behind the web page interactions. In the first assumption, two interacting web pages discuss about same topics and as a result their tag set will be similar (Tag Similarity assumption). Second assumption claims that interacting web pages complement the topics of each other and they do not necessarily have the same set of tags (Tag Collaboration assumption). We propose "Majority Rule Tag Predictor" as one of the possible implementations for *tag similarity* assumption. Two methods based on Reinforcement Learning (RL) and Self Organizing Map (SOM) are proposed for implementing *tag collaboration* assumption. We describe each proposed implementation in detail in the following:

Majority Rule Tag Predictor: This method simply implements tag similarity assumption by finding the most common tag(s) among the neighbors of the unannotated web page. Typically, a fixed number of tags is predicted, for instance the five most frequently occurring tags in the neighborhood are predicted for the unannotated web page.

The hypothesis behind *tag similarity* based approaches is that web pages with similar tags are always topologically close in the graph which all web pages in actual graphs not necessarily corroborate this hypothesis.

Reinforcement Based Tag Predictor: In this method, we try to quantify how strongly two tags t_i and t_j collaborate, in the following way. Let $TagColVal(t_i, t_j)$ denote the strength of collaboration between t_i and t_j.

We consider each annotated web page $p \in P$ in turn. If tag t_j occurs in the neighborhood of web page p (i.e., $NB_p(t_j) > 0$) then we increase the collaboration value between tag t_j and all the tags in TS_p:

$$\forall t_i \in TS_p : TagColVal(t_i, t_j) {+}{=} \frac{NB_p(t_j) * R}{support(t_j)}$$

If tag t_j does not occur in the neighborhood of p ($NB_p(t_j) = 0$), we decrease the collaboration value between tag t_j and all the tags belonging to TS_p:

$$\forall t_i \in TS_p : TagColVal(t_i, t_j) {-}{=} \frac{P}{support(t_j)}$$

$support(t_j)$ is the total number of times that tag t_j appears on a side of an edge e_{pq} in the graph. R and P are "Reward" and "Punish" coefficients determined by the user.

Next, we determine the candidate tags for an unannotated web page p and rank them based on how well they collaborate with the neighborhood of web page p. As an example of a candidate tag strategy, consider Majority Rule: this method nominates all tags that appear in the direct neighborhood of the unannotated web page (and among these, will select the most frequently occurring ones). Here, we consider a number of extensions to this candidate tag strategy:

- First Tag Level Strategy (First-TL): This strategy selects the tags that appear in the direct neighborhood of the web page as candidate tags. This strategy nominates tags similar to the Majority Rule method.
- Second, Third, Fourth Tag Level Strategy (Second-TL, Third-TL, Fourth-TL): Define the n-neighborhood of a web page p as all the web pages that are reachable from p with a path of length at most n (thus, the 2-neighborhood includes all neighbors of neighbors of p, etc.). In Second-TL, Third-TL, Fourth-TL, all the tags occurring in the 2-, 3- or 4-neighborhood of p, respectively, are considered as candidate tags.
- All Tag Strategy (All-TL): All the tags are taken into account in this strategy.

After selecting candidate tags, we rank them based on how well they collaborate with the neighborhood of unannotated web page p. Formula (3) assigns a collaboration score to each candidate tag t_c:

$$Score(t_c) = \sum_{\forall t_j \in T} NB_p(t_j) * TagColVal(t_j, t_c) \tag{3}$$

High score candidate tag(s) collaborate(s) better with the neighborhood of p and are predicted as its tags.

We call the above method the "Reinforcement based tag predictor", as it is based on reinforcing collaboration values between tags as they are observed.

SOM Based Tag Predictor: Our second method makes use of a Self Organizing Map (SOM) for the task of tag prediction in a graph of web pages. We map the web page graph to a SOM as follows:

- Input Layer: The number of input neurons equals the number of tags in the web page graph. So, if $inputNeurons$ is the set of all neurons in the input layer then $|inputNeurons| = |T|$. The values we put in the input layer are extracted from the neighborhood tag vector of the web page: if $inputNeuron(i)$ is the i'th neuron in the input layer then $inputNeuron(i) = NB_p(t_i)$.
- Output Layer: The number of output neurons equals to the number of tags in the web page graph ($|outputNeurons| = |T|$). The values we put in the output layer are extracted from the tag vector of the web page: if $outputNeuron(i)$ is the i'th neuron in the output layer then $outputNeuron(i) = TS_p(t_i)$.

– Network Initialization: Weights of the neurons can be initialized to small random values; in our implementation we initialized all the weights to zero.
– Adaption: Weights of winner neurons and neurons close to them in the SOM lattice should be adjusted towards the input vector. The magnitude of the change decreases with time and with the distance from the winner neuron. Here, we take some new parameters into consideration which are $LearningRate(LR)$, $DecreasingLearningRate(DecLR)$ and $TerminateCriteria(TC)$ parameters. LR is the change rate of the weights toward the input vector and $DecLR$ determines the change rate of LR in different iterations. TC is the criteria in which the learning phase of SOM will terminate. Here, we think of TC as the minimum amount of change required in one iteration: when there is less change, the training procedure stops. We use Formula (4) for updating weights of output neurons.

$$W_{ij,New} = W_{i,j,Current} + LR * (NB_p(j) - W_{i,j,Current}) \tag{4}$$

– Testing: For each web page p in the graph that we did not use in the training phase, we find the Euclidean distance between NB_p and the weight vectors. We select the output neurons which have the shortest Euclidean distance to NB_p and predict them as the tag set of web page p. The number of predicted tags is fixed and determined by the user.

4 Empirical Results

4.1 Dataset

We were interested in graphs of web pages in which *a*) nodes are reasonably interconnected to each other at least as form of a tree; and *b*) nodes are tagged in a tagging system such as Delicious. Seemingly, there is no current data set available providing such information for web pages. Starting with ECML PKDD 09 Data Set [3] as the base, the first issue was to update the *weighted* tag assignments on web pages; we used URL's in the original data set to fetch their weighted tag assignments from Delicious [4]. The next consideration was the fact that there are many web pages in the data set that are *sparsely tagged* at Delicious; thus, we constrained the data set to the web pages annotated in Delicious with a minimum tag assignment weight. To build the desired graphs, starting from a web page in the data set, we included those neighboring URLs of the web page (and the links to them) that were tagged at Delicious. The same procedure was then applied to those neighboring URLs, and so on, up to a maximum crawling depth. Table 1 shows brief statistical information of datasets we used for evaluating our methods. Detailed information on data construction and preprocessing phase is available at `http://www.liacs.nl/~bnobakht/social-tag-prediction/`.

[3] `http://www.kde.cs.uni-kassel.de/ws/dc09/dataset/#files`
[4] `http://delicious.com/help/feeds`

Number of web page graph	27
Average page count of each web page graph	140.40
Average link count of each web page graph	311.03
Average number of tags for each page	6.92

Table 1. Statistical information of datasets.

4.2 Parameter Tuning

The methods we want to evaluate have some parameters for which good values need to be found. In order to tune the method's parameters, we select five different graphs and tune the parameters on those graphs and then use the tuned values for the other graphs.

While Majority Rule assigns only tags from the direct neighborhood to a web page, we are interested to find out whether using candidate tags from a wider neighborhood (including neighbors of neighbors) would be advantageous. We tested this by extending the Majority Rule so that it can consider not only direct neighbors, but also neighbors in the 2- or 3-neighborhood (as defined in Section 3.2). Figure (1) shows the effect of considering a wider neighborhood in Majority Rule in different datasets. There is no improvement over the base MR method by considering the second and third neighborhood level.

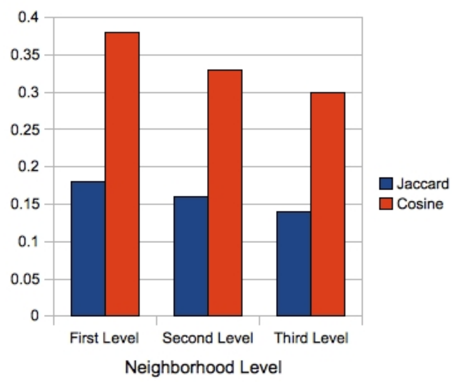


Fig. 1. Tuning "Neighborhood Tag Level" in the Majority Rule Tag Predictor method. There is no improvement over the base MR method by considering the second and third neighborhood level.

Figure 2 tunes the "Candidate Tag Strategy" parameter of RL method. The best value for this parameter is "First Level".

"Number of clusters (k)" is the parameter which should be tuned in the K-means algorithm. Before we tune this parameter, we introduce a *topic locality*

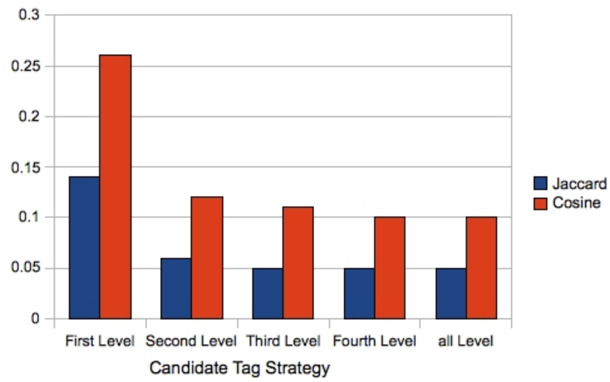


Fig. 2. Tuning "Candidate Tag Strategy" in the RL method. "First Level" tag strategy produces the best result.

feature in a graph of web pages. We define *topicLocality*(G) as the number of distinct cluster tags divided by the number of clusters in the graph G (Formula 5).

$$TopicLocality(G) = \frac{Number\ of\ distinct\ cluster\ tags}{Number\ of\ clusters} \tag{5}$$

For instance, if all the web pages in the graph talk about the same topic then independent of different values for k, applying the K-means tag predictor will always lead to the same result. In this case we have small topic locality. But, if different parts of the graph discuss different topics, then we expect that by increasing the number of clusters, we increase the topic locality and this results in better tag prediction. Figure 3 shows the value of *topicLocality* in three different graphs. We cluster each graph into four clusters. In graph $G1$, all four clusters are about topic "Music", so the value of *topicLocality* is $\frac{1}{4}$ in this graph. Two clusters of graph $G2$ are about "Music", while the topics of the other two clusters are "Sport" and "Food". There are 3 distinct topics and 4 clusters, therefore the *topicLocality* of graph $G2$ equals $\frac{3}{4}$. In $G3$, all four clusters are about distinct topics, so the *topicLocality* of $G3$ equals $\frac{4}{4}$. So, choosing the right value for k in K-means algorithm is completely dependent to the *topic locality* of that specific graph.

Figure 4-(a) shows the result of applying K-means tag predictor with different values of k to graph $G(142, 292)$ with high *topic locality*. In a case of $k \leq 5$, there are k big "topic divergent" clusters, so the average cosine similarity is small (around 2%). As we increase the number of clusters the average cosine similarity value improves which means *topic locality* value of this graph is high. The best setting for K-means tag predictor is $60 \leq K \leq 64$.

Figure 4-(b) shows the effect of choosing different values for k in graph $G(362, 934)$ with low *topic locality*. In this graph, the topic of the most of the

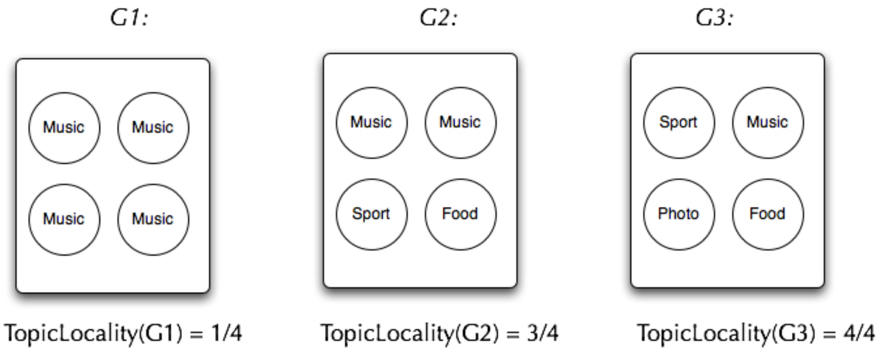


Fig. 3. *TopicLocality* character of different graphs.

web pages are similar to each other and changing the value of k does not effect the average cosine similarity a lot.

4.3 Comparison of Different Methods

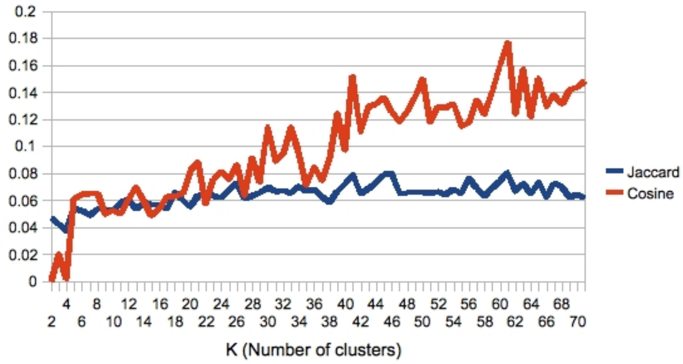
In this section, we compare content-based tag predictors (i.e., Most Similar and K-means) with tag-based tag predictors (i.e., Majority Rule, RL and SOM) on several datasets. We use average cosine similarity as the evaluation criterion. We predict 5 tags for each unannotated web page in all methods and then we compare the cosine similarity of different methods.

In the proposed implementations, we use the parameter values tuned in the previous section. Majority Rule (MR) selects the five most frequently occurring tags in the neighborhood of the web page in the network. As we discussed in the previous section, choosing the right value for k in K-means algorithm is completely dependent to the topic locality nature of that specific graph. In this section, we use fixed value for $k(=\frac{N}{5})$ for clustering all the graph. N equals the number of web pages in the graph.

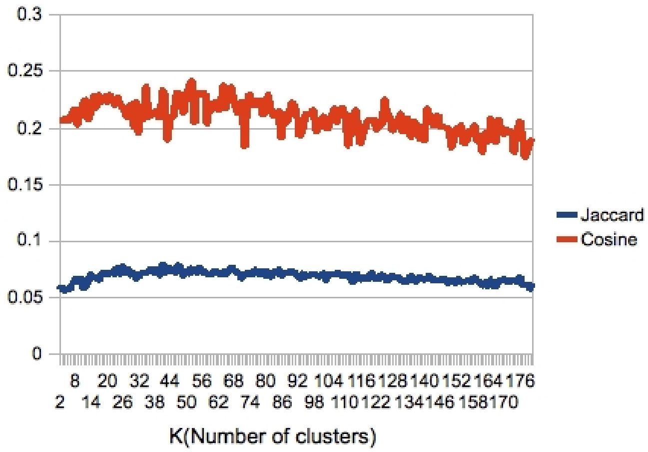
Figure 5 compares content-based tag predictors with tag-based tag predictors. Tag-based tag predictors outperform their content-based peers by more than 10 percent with respect to cosine similarity metric.

By assuming number of known tags for a given web page ("Social Tag Expansion" methods [11, 12]) or limit our methods predicting just small set of frequent tags (like [12]), we could expect a gain in precision; however this could drastically restrict the generality of our framework.

After analyzing each graph individually, we believe that there is no fixed approach (or parameters) which works best in all different types of web-page graphs. For example, considering "Topic locality" as a one out of many graph characteristics, in the graph with low Topic Locality (G1 in Figure 3) it might be better to use methods working based on the Tag Similarity assumption, while in high Topic Locality graphs (G3 in Figure 3), it is recommended to use Tag Collaboration based methods.



(a) Graph $G(142, 292)$ with high "Topic Locality" characteristics.



(b) Graph $G(362, 934)$ with low "Topic Locality" characteristics.

Fig. 4. Tuning "Number of Clusters" in the K-means algorithm.

5 Conclusion

To our knowledge, this is the first study that considers graph of annotated web pages for the task of tag prediction. We proposed two different approaches for the task of social tag prediction in a graph of web pages. For each approach, we recommend different implementations. The first approach uses only the content of the web pages for the task of tag prediction. For this approach we propose "Most Similar" and "K-means" methods. Contrary to the first approach, the second approach uses only tag information available in the graph for the task of tag prediction. It considers two assumptions about the tag set of two interacting web pages in a graph. The first assumption says that two interacting web pages have similar tag sets (Tag Similarity Assumption) while the second one assumes

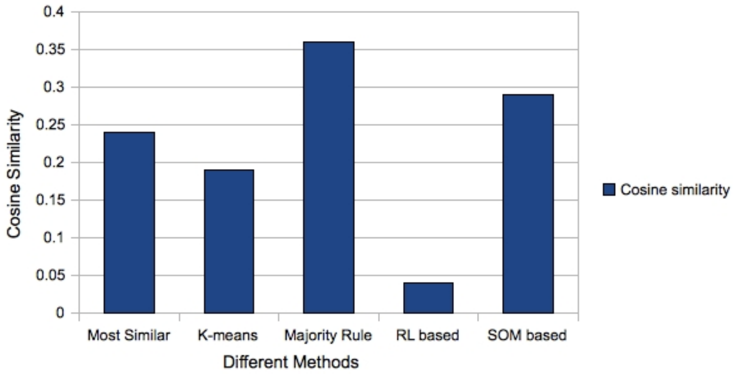


Fig. 5. Compare content-based tag predictors (i.e.,Most Similar and K-means) with tag-based tag predictors (i.e., Majority Rule, RL and SOM) on the different datasets. Tag-based tag predictors outperform their content-based peers more than ten percent with respect to cosine similarity.

that two interacting web pages have collaborative tag sets (Tag Collaboration Assumption). We proposed "Majority Rule" method as one of the possible implementations for that similarity assumption. This method simply predicts most frequently occurring tags in the neighborhood of unannotated web page. We used two machine learning methods Self Organizing Map (SOM) and Reinforcement Based Tag Predictor for implementing tag collaboration assumption. Both methods first learn the collaboration value between each pair of tags and then at prediction time, they rank candidate tags based on how well they collaborate with the neighborhood of unannotated web page.

We compared content-based tag predictors with tag-based tag predictors and we found out that Tag-based tag predictors outperform their content-based peers by more than 10 percent with respect to cosine similarity metric. Among the tag-based tag predictors, Majority Rule method predicts the best tags for unannotated web pages which means Tag Similarity Assumption dominates Tag Collaboration Assumption in the graph of web pages. So, in general, web pages in the our dataset tend to discuss more about similar topics rather than complementary topics.

We also analyzed each graph individually and we concluded that graph characteristics have direct impact on choosing the right method for social tag prediction. We observed that in low Topic Locality graphs, results of Tag Similarity methods outperform the results Tag Collaboration methods while in graphs with high topic locality, it is better to apply Tag Collaboration methods.

Funding

This research is funded by the Dutch Science Foundation (NWO) through VIDI grant 639.022.605.

References

1. Han, P., Wang, Z., Li, Z., Kramer, B., Yang, F.: Substitution or complement: An empirical analysis on the impact of collaborative tagging on web search. In: WI '06: Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence, Washington, DC, USA, IEEE Computer Society (2006) 757–760
2. Bao, S., Xue, G., Wu, X., Yu, Y., Fei, B., Su, Z.: Optimizing web search using social annotations. In: WWW '07: Proceedings of the 16th international conference on World Wide Web, New York, NY, USA, ACM (2007) 501–510
3. Aliakbary, S., Abolhassani, H., Rahmani, H., Nobakht, B.: Web page classification using social tags. In: CSE '09: Proceedings of the 2009 International Conference on Computational Science and Engineering, Washington, DC, USA, IEEE Computer Society (2009) 588–593
4. Zubiaga, A., Martínez, R., Fresno, V.: Getting the most out of social annotations for web page classification. In: DocEng '09: Proceedings of the 9th ACM symposium on Document engineering, New York, NY, USA, ACM (2009) 74–83
5. Vander Wal, T.: Folksonomy definition and wikipedia. (November 2005)
6. Quintarelli, E.: Folksonomies: power to the people. ISKO Italy-UniMIB Meeting (2005)
7. Hotho, A., Jäschke, R., Schmitz, C., Stumme, G.: Information retrieval in folksonomies: Search and ranking. In Sure, Y., Domingue, J., eds.: ESWC. Volume 4011 of Lecture Notes in Computer Science., Springer (2006) 411–426
8. Jschke, R., Marinho, L.B., Hotho, A., Schmidt-Thieme, L., Stumme, G.: Tag recommendations in folksonomies. In Kok, J.N., Koronacki, J., de Mntaras, R.L., Matwin, S., Mladenic, D., Skowron, A., eds.: Knowledge Discovery in Databases: PKDD 2007, 11th European Conference on Principles and Practice of Knowledge Discovery in Databases, Warsaw, Poland, September 17-21, 2007, Proceedings. Volume 4702 of Lecture Notes in Computer Science., Springer (2007) 506–514
9. Chirita, P.A., Costache, S., Nejdl, W., Handschuh, S.: P-tag: large scale automatic generation of personalized annotation tags for the web. In: WWW '07: Proceedings of the 16th international conference on World Wide Web, New York, NY, USA, ACM (2007) 845–854
10. Sigurbjörnsson, B., van Zwol, R.: Flickr tag recommendation based on collective knowledge. In: WWW '08: Proceeding of the 17th international conference on World Wide Web, New York, NY, USA, ACM (2008) 327–336
11. Budura, A., Michel, S., Cudr-Mauroux, P., Aberer, K.: Neighborhood-based tag prediction. In Aroyo, L., Traverso, P., Ciravegna, F., Cimiano, P., Heath, T., Hyvnen, E., Mizoguchi, R., Oren, E., Sabou, M., Simperl, E.P.B., eds.: ESWC. Volume 5554 of Lecture Notes in Computer Science., Springer (2009) 608–622
12. Heymann, P., Ramage, D., Garcia-Molina, H.: Social tag prediction. In Myaeng, S.H., Oard, D.W., Sebastiani, F., Chua, T.S., Leong, M.K., eds.: SIGIR, ACM (2008) 531–538
13. MacQueen, J.B.: Some methods for classification and analysis of multivariate observations. In Cam, L.M.L., Neyman, J., eds.: Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability. Volume 1., University of California Press (1967) 281–297