

A Collaborative Filtering Tag Recommendation System based on Graph

Yuan Zhang, Ning Zhang, and Jie Tang

Knowledge Engineering Group
Department of Computer Science and Technology, Tsinghua University, Beijing, China
fancyzy0526@gmail.com, zntsinghua1117@gmail.com and
jietang@tsinghua.edu.cn

Abstract. With the rapid development of web2.0 technologies, tagging become much more important today to organize information and help users search the information they need with social bookmarking tools. In order to finish the second task of ECML PKDD challenge 2009, we propose a graph-based collaborative filtering tag recommendation system. We also refer to an algorithm called FolkRank, which is an adaptation of the famous Page Rank. We evaluate and compare these two approaches and show that a combination of these two methods will perform better results for our task.

1 Introduction

Tagging is very useful for users to figure out other users with similar interests within a given category. Users with similar interests might post similar tags and similar resources might have similar tags posted to them. Collaborative filtering is widely used in automatic prediction system. The idea behind it is very simple: those who agreed in the past tend to agree again in the future. Traditional collaborative filtering systems have two steps. The first step is to look for users who share the same rating patterns with the active user whom the prediction is for. Then, the systems will use the ratings from those like-minded users found in the first step to calculate a prediction for the active user. Since all the tags, users and resources in the test data are also in the training file, we can make use of the history of users' tag, also called *personomy*[3] and tags previously posted to the resource to recommend tags for a active post. This paper presents our proposed tag recommendation system, which is a combination of two methods: one is an adaption of item-based collaborative filtering, the other is FolkRank according to [4,5].

As we mentioned above, collaborative filtering performs well for automatic prediction. However, current widely used collaborative filtering systems are for predicting the ratings of some products or recommend some products to users. For example, the famous websites, Amazon.com¹, Last.fm², eBay³ apply this method to

¹ <http://www.amazon.com>

their recommendation systems. Our first method considers the tags previously posted to the resource and users' similarities to recommend tags. The second method is an application of the FolkRank algorithm in [4, 5].

These two methods have some common features. They both use the history of the user and tags previously posted to resource for recommendation. They are both suitable to the case that test data are in the training data. Both of them do not need to establish models in advance. But they are different to some extents. The first method just considers tags in the candidate set while the FolkRank will consider all the tags in the training data. Moreover, the first method focuses more on collaborative information while the second focuses on the graph information.

This paper is organized as follows: Section 2 introduces recent trends in the area of social bookmark tag recommendation systems. Section 3 describes our proposed system and the combination method in details. In Section 4, we present and evaluate our experimental results on the test data of ECML PKDD challenge 2009 and make some conclusions in Section 5.

2 Related work

Some researchers have already used some approaches based on collaborative information for tag recommendation systems. For example, AutoTag[7] and TagAssist[6] make use of information retrieval skills to recommend tags for weblog posts. They recommend tags based on the tags posted to the similar weblogs. Our first method is similar to these two approaches.

FolkRank in[4, 5] is a topic-specific ranking in folksonomies. The key idea of FolkRank algorithm is that a resource which is tagged with important tags by important users becomes important itself. In [5], the author compared the performance of some baseline methods and his FolkRank algorithm, and found that FolkRank outperformed other methods. His experimental results relied on a dense core of the training file and considering that our training data is a post-core two dataset, we decide to refer to this algorithm in our proposed tag recommendation system.

In the RSDC '08 challenge, the participants [1, 2] who make use of resource's similarities and users' personomy outperformed other approaches. Consequently, we consider using the collaborative information of resource's similarities and users' personomy in our tag recommendation system.

² <http://www.last.fm>

³ <http://www.eBay.com>

3 Our Tag Recommendation

3.1 Notations

First, we define notations used in this paper. We group the data in bookmark by its url_hash and data in bibtex by its simhash1. If some posts in bookmark of bibtex file have the same url_hash or simhash1, they are mapped to one resource r . For each resource r_d , assuming a vector \mathbf{t}_d of T_d tags posted to this resource r_d by the user u_d . Then the training dataset can be represented as

$$D = \{(r_1, t_1, u_1), \dots, (r_D, t_D, u_D)\}$$

Table1 summarizes the notation.

Symbol	Description
T	the collection of tags posted in the training data
R	the collection of resources posted in the training data (grouped by the url_hash or simhash1)
U	the collection of users who posted tags in the training data
D	training data set containing tagged resources. $D = \{(r_j, \mathbf{t}_i, u_i)\}$, which represents a set of pairs of resources and users, with the assigned tags by the corresponding users.
D'	The test data set containing resources and users. $D' = \{(r_j, u_i)\}$. Note that: the user u_i , the resource r_j and the original tags posted by u_i to r_i appear in the training dataset.
N_d	number of word tokens in the $d \in D$
T_r	number of tags posted to resource r
\mathbf{t}_d	vector form of tags in $d \in D$
u_d	the user in $d \in D$
$C(u, r)$	the candidate set of tags to be recommended for a given user u and a given resource r
$\tilde{T}(u, r)$	the set of tags that will be recommended for a given user u and a given resource r
$n(t, r)$	the number of times that the tag t has been posted to the resource r in the training dataset

Table1: Notations

3.2 Collaborative Filtering method

Our proposed collaborative filtering method for tag recommendation has two steps. First of all, for a given resource r and a given user u in the test dataset, we make use of the tags previously posted to the resource r in the training dataset and define them as the candidate set:

$$C(u, r) = \{t_i \mid (r, t_i, u') \in D, u' \in U\}$$

The second step is to score all the tags in the candidate set and recommend the tags with the highest scores. In our proposed tag recommendation system, we score the tags in the candidate set using the following equation for all tags $t \in C(u, r)$:

$$P(t \mid r) = \frac{T_r}{T_r + \lambda} \cdot \frac{n(t, r)}{T_r} + \left(1 - \frac{T_r}{T_r + \lambda}\right) \cdot \frac{n(t, R)}{|T|} \quad (1)$$

where $n(t, r)$ is the number of times that tag t has been posted to the resource r and $n(t, R)$ is the number of times that tag t has been posted in the training data. T_r is the number of tags posted to the resource r and λ is the Dirichlet smoothing factor and is commonly set according to the average document length, i.e. $|T|/|R|$

In order to take the users' similarities into consideration, we change the equation (1) to the following equation:

$$P(t \mid r, u) = \frac{T_r}{T_r + \lambda} \cdot \frac{\sum_{u' \in U'} (sim(u, u') \cdot m(t, u, r))}{T_r} + \left(1 - \frac{T_r}{T_r + \lambda}\right) \cdot \frac{\sum_{u' \in U'} (sim(u, u') \cdot m(t, u, R))}{|T|} \quad (2)$$

where $U' = \{u \mid (r, t, u) \in D\}$ for a given resource r , $m(t, u, r)$ is the number of times tag t has been posted to the resource r by the user u . The similarity of users $sim(u, u')$ is define as follows,

$$sim(u, u') = \frac{\sum_{t \in T} n(t, u) \cdot n(t, u')}{\sqrt{\sum_{t \in T} n(t, u)^2} \cdot \sqrt{\sum_{t \in T} n(t, u')^2}} \quad (3)$$

For a given user u and a given resource r , the set of recommended tags will be: $\tilde{T}(u, r) := \arg_{t \in C(u, r)}^n P(t \mid r, u)$ where n is the number of recommended tags.

3.3 FolkRank algorithm

FolkRank is a graph-based algorithm whose basic idea is to rank all the tags and pick out tags which are relatively important given a user u and a resource r . This algorithm is derived from the PageRank algorithm, which is used by the Google Internet search engine that assigns a numerical weighting to each element of a hyperlinked set of documents. The purpose of PageRank is to measure the hyperlink's relative importance within the set. However, due to the structural differences between hyperlinks and our tag recommendation system, we cannot apply the PageRank to our tag recommendation system and a new FolkRank algorithm was introduced in [4, 5].

In order to apply a weight-spreading ranking scheme to recommend tags, we need to change the directed graph in PageRank to an undirected graph and change the corresponding ranking approach.

First, we convert the training dataset D into an undirected graph $G = (V, E)$. V is the set of the nodes in the graph, which is composed of all the tags, resources and users in the training file, i.e. $V = T \cup R \cup U$. E is the set of the edges in the graph, which is defined as the co-occurrences of tags and users, users and resources, tags and

resources. $E = \{\{u, t\}, \{t, r\}, \{u, r\} \mid \{r, t, u\} \in D\}$ and each edge $\{u, t\} \in E$ has a weight $|\{r \in R \mid \{r, t, u\} \in D\}|$, each edge $\{t, r\} \in E$ has a weight $|\{u \in U \mid \{r, t, u\} \in D\}|$ and each edge $\{u, r\} \in E$ has a weight $|\{t \in T \mid \{r, t, u\} \in D\}|$. After having the graph format of the posts, we can spread the weight like PageRank as follows:

$$\bar{w} \leftarrow dA\bar{w} + (1-d)\bar{p} \quad (3)$$

where A is the adjacency matrix of G , \bar{p} is the random surfer component, and $d \in [0,1]$ is a constant which controls the influence of the random surfer.

Usually, \bar{p} is set to the vector where all values equal to 1. But in order to recommend tags relevant to certain user and certain resource, we can change the \bar{p} to express user preferences. In our tag recommendation system, each user, tag, and resource get a preference weight of 1 but the active user and resource for recommendation get a preference of $1+|U|$ and $1+|R|$ respectively.

The FolkRank algorithm has a differential approach to see the ranking around the topics defined in the preference vector. This approach is to compare the rankings with and without the preference vector \bar{p} . Assuming that w_0 is the ranking after iteration with $d = 1$ while w_1 is the ranking after iteration with $d = 0.625$, then the final weight will be $w = w_1 - w_0$. Details can be found in Algorithm 1.

Input: the graph information of the training file, i.e. $G = (V, E)$ where $V = T \cup R \cup U$ and $E = \{\{u, t\}, \{t, r\}, \{u, r\} \mid \{r, t, u\} \in D\}$, the adjacency matrix A , the given resource r and the given user u .

Output: the ranking w of all tags $\in T$

begin

//Initialize

foreach $t \in T, r \in R$ and $u \in U$ **do**

$w_0[t] = w_1[t]=1, w_0[r] = w_1[r]=2$ and $w_0[u] = w_1[u] = 2$

end

foreach $t \in T, r \in R$ and $u \in U$ **do**

$p[t]=p[r]=p[u]=1$

end

$p[r] = 1+|R|$

$p[u] = 1+|U|$

$d = 0.625$

//iteration for w_1

repeat

$\bar{w}_1 = dA\bar{w}_1 + (1-d)\bar{p}$

until convergence

//iteration for w_0

repeat

$\bar{w}_0 = A\bar{w}_0$

until convergence

$w = \bar{w}_1 - \bar{w}_0$

end

Algorithm 1: The FolkRank algorithm used in our tag recommendation system

3.4 Combination

We have proposed two different but similar methods for our tag recommendation system. Both are suitable to our case that the test data have already appeared in the training file, both make use of the similarity of users and resources, but the first method focuses more on the collaborative information while the second one focus more on the graph nodes and can spread the weight according to the co-occurrences. We hope to combine these two methods and get a better result.

We have tried some different approaches to combine these two methods. A simple method of combination is to multiply the scores of these two models and recommend tags with highest scores after combination. Details can be found in Algorithm 2.

```
Input: a given resource  $r$  and a given user  $u$  and the result of the two methods  
Output: the set of recommended tags  $\tilde{T}(u, r)$   
begin  
  //collaborative method  
  the candidate set  $C(u, r) \leftarrow \{t|(r, t, u') \in D, u' \in U\}$   
  foreach  $t \in C$  do  
    score1[t] =  $P(t|r, u)$  in equation(2)  
  end  
  //FolkRank algorithm  
  foreach  $t \in T$  do  
    score2[t] =  $w$ , the output of the algorithm 1  
  end  
  //combination  
  foreach  $t \in T$  do  
    score[t] = score1[t] × score2[t]  
  end  
   $\tilde{T}(u, r) := \operatorname{argmax}_{t \in T}^n \text{score}[t]$   
end
```

Algorithm 2: the combination method used in our tag recommendation system

4 Experimental Results

4.1 Dataset

We evaluate our experimental results using the evaluation methods provided by the organizers of ECML PKDD discovery challenge 2009. The training set and the test set are strictly divided and we use the post-core level 2 training file as our training dataset for our tag recommendation system.

The general statistical information of training data and test data can be found in the table 2 and table 3.

	tag assignments	D	R	U	T	average no. of tags
bookmark	916,469	263,004	235,328	2679	50,855	3.48
bibtex	484,635	158,924	143,050	1790	56,424	3.05
total	1,401,104	421,928	378,378	3617	93,756	3.32

Table2: the general statistical information about the training dataset

	tag assignments	D	R	U	T	average no. of tags
bookmark	1,465	431	387	91	587	3.40
bibtex	1,139	347	280	81	397	3.28
total	2,604	778	667	136	862	3.35

Table 3: the general statistical information about the test dataset

4.2 Experimental Result

As performance measures we use precision, recall and f-measure. For a given user u and a given resource r , the true tags are defined as $TAG(u,r)$, then the precision, recall and f-measure of the recommended tags $\tilde{T}(u,r)$ are defined as follows:

$$\text{recall}(\tilde{T}(u,r)) = \frac{1}{|U|} \sum_{u \in U} \frac{|TAG(u,r) \cap \tilde{T}(u,r)|}{|TAG(u,r)|}$$

$$\text{precision}(\tilde{T}(u,r)) = \frac{1}{|U|} \sum_{u \in U} \frac{|TAG(u,r) \cap \tilde{T}(u,r)|}{|\tilde{T}(u,r)|}$$

$$\text{f-measure}(\tilde{T}(u,r)) = \frac{2 \times \text{recall} \times \text{precision}}{\text{recall} + \text{precision}}$$

4.2.1 Performance of Collaborative Filtering method

In table 4, we show the performance of collaborative filtering method on the test data provided by the organizers of ECML PKDD challenge 2009. From the table, we can see that this method has a highest f-measure of 30.002% when the number of recommended tags is 5.

4.2.2 Performance of FolkRank method

In table 4, we show the performance of FolkRank algorithm on the test data. From the table, we can find that the first method performs a little bit better than FolkRank and FolkRank has a highest f-measure of 28.837% when the number of recommended tags is 4.

	collaborative method	FolkRank algorithm	Combined result
1	13.000/37.147/19.262	14.132/40.231/20.917	14.400/41.512/21.381
2	20.220/31.362/24.588	22.827/33.419/27.126	21.309/38.368/27.400
3	26.760/28.813/27.749	28.326/29.092/28.704	25.117/37.125/29.962
4	32.571/27.035/29.546	32.783/25.739/28.837	27.744/36.739/31.614
5	36.569/25.435/30.002	36.229/23.342/28.392	28.670/36.225/32.008
6	39.079/23.811/29.592	38.826/21.208/27.423	29.409/35.981/32.364
7	41.205/22.670/29.248	40.733/19.262/26.155	29.763/35.935/32.560
8	42.860/21.896/28.985	42.096/17.625/24.847	29.901/35.880/32.619
9	43.863/21.089/28.483	43.227/16.195/23.570	29.933/35.803/32.606
10	45.367/20.591/28.325	44.620/15.077/22.539	29.984/35.769/32.622

Table 4: performance of two methods and combination on the test data, the numbers are shown in the following format: recall/precision/f-measure

4.2.3 Performance of combination

In table 4, we show the performance after the combination of the previous two methods. We are glad to see that the results after combination outperform these two methods. We have a 2% increase compared to the first method and a 4% increase compared to the second method. We have a highest f-measure of 32.622% when recommending 10 tags. The precision-recall plot in Fig.1 reveals the quality of our recommendation system.

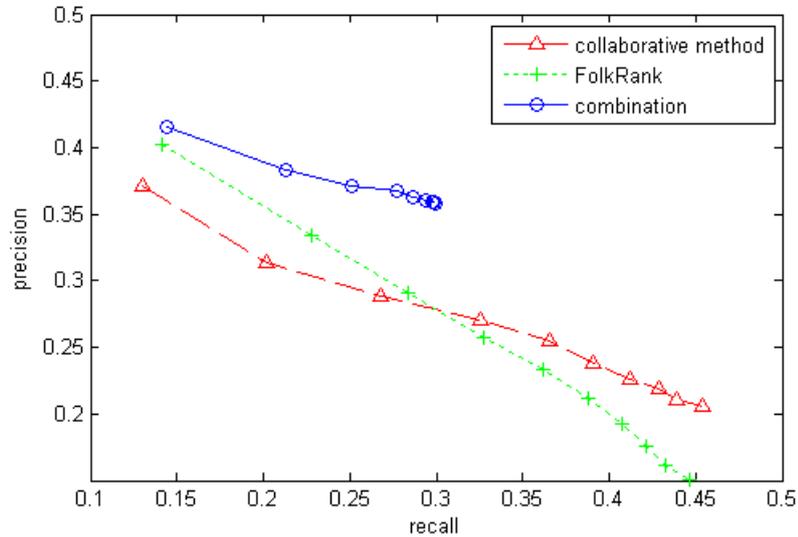


Fig.1 Recall and precision of tag recommendation system

5 Conclusions

In this paper, we describe our tag recommendation system for the second task in the ECML PKDD Challenge 2009. We exploit two different methods to recommend tags when tags, resources, users in the test data are also in the training file. The experimental results show that the combination of these two methods will gain a better result.

We need to further analyze the results to see which kind of information in the graph contributes more to the final ranking. Also, we can try to change the scoring scheme or expand the candidate set in our collaborative filtering method. Future work also includes some adaptations of PageRank for the tag recommendation system.

References

1. Marta Tatu, Munirathnam Srikanth, and Thomas D'Silva. RSDC'08: Tag Recommendations using Bookmark Content. In Proc. of ECML PKDD Discovery Challenge (RSDC 08), pp. 96-107
2. Marek Lipczak. Tag Recommendation for Folksonomies Oriented towards Individual Users. In Proc. of ECML PKDD Discovery Challenge(RSDC 08), pp. 84-95
3. Andreas Hotho, Robert Jaschke, Christoph Schmitz, and Gerd Stumme. BibSonomy: A Social Bookmark and Publication Sharing System. In Proc. the First Conceptual Structures Tool Interoperability Workshop at the 14th Int. Conf. on Conceptual Structures, pages 87-102, Aalborg, 2006. Aalborg Universitetsforlag.
4. Andreas Hotho, Robert Jaschke, Christoph Schmitz, and Gerd Stumme. FolkRank: A Ranking Algorithm for Folksonomies. In Proc. of FGIR 2006
5. Robert Jaschke, Leandro Marinho, Andreas Hotho, Lars Schmidt-Thieme, and Gerd Stumme. Tag Recommendations in Folksonomies. In Knowledge Discovery in Database: PKDD 2007, 11th European Conference on Principles and Practice of Knowledge Discovery in Database, Warswa, Poland, September 17-21, 2007, Proceedings, volume 4702 of LNCS, pages 506-514. Springer, 2007.
6. Sanjay C. Sood, Sara H.Owsley, Kristian J.Hammond, and Larry Birnbaum. TagAssist: Automatic tag suggestion for blog posts. In Proc. the International Conference on Weblogs and Social Media(ICWSM 2007),2007.
7. Gilad Mishne. Autotag: a collaborative approach to automated tag assignment for weblog posts. In WWW'06: Proceedings of the 15th international conference on World Wide Web, pages 953-954, New York, NY, USA, 2006. ACM Press.Web, pages 953-954, New York, NY, USA, 2006. ACM Press.