FoldCons: A Simple Way To Improve Tag Recommendation

Modou Gueye
Université Cheikh Anta DIOP
Dakar, Sénégal
gmodou@ucad.sn

Talel Abdessalem
Institut Telecom - Telecom ParisTech
Paris, France
Talel.Abdessalem@enst.fr

Hubert Naacke
LIP6, UPMC Sorbonne Universités - Paris 6
Paris, France
Hubert.Naacke@lip6.fr

ABSTRACT

Tag recommendation is a major aspect of collaborative tagging systems. It aims to recommend tags to a user for tagging an item. In this paper we present a part of our work in progress which is a novel improvement of recommendations by re-ranking the output of a tag recommender. We mine association rules between candidates tags in order to determine a more consistent list of tags to recommend.

Our method is an add-on one which leads to better recommendations as we show in this paper. It is easily parallelizable and morever it may be applied to a lot of tag recommenders.

The experiments we did on five datasets with two kinds of tag recommender demonstrated the efficiency of our method.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Information Filtering

General Terms

Algorithms, Experimentation

Keywords

Tag Recommendation, Association Rules Mining, Social Network, Factorization Models

1. INTRODUCTION

Social (i.e. collaborative) tagging is the practice of allowing users to annotate content. Users can organize, and search content with annotations called tags. Nowadays the growth in popularity of social media sites has made the area of recommender systems for social tagging systems an active and growing topic of research [4]. Tag recommenders aim to recommend the most suited tags to a user for tagging an item. They are a salient part of the web 2.0 where applications are user-centred.

In this paper, we propose a novel improvement of tag recommendation. We present *FoldCons* an add-on method which can fold more consistency in recommendations. Furthermore it is applicable on top of many tag recommenders and is very fast to compute. The main idea behind FoldCons is that the first of recommended tags computed by a tag recommender plays more important role than the rest. We may think that it is the most interesting tag since it has the highest score. Thus FoldCons relies on this first tag to sort the rest in order to achieve better consistency and improvement. Of course, the same reasoning can be used with the second tag, then the third one and so on.

To validate the efficiency of FoldCons who chose two kinds of tag recommender as candidates. One which proves itself is the pairwise interaction tensor factorization model (PITF) of Rendle and Schmidt-Thieme which wins the task 2 of ECML PKDD Discovery Challenge 2009 [7]. Currently one of the best tag recommenders in literature. The other is an adaptation of the network-aware search in online social bookmarking applications of [6] to tag recommendation, we called STRecIt is a network-based tag recommender which considers the opinions of users' neighbourhood. The experiments we did on five datasets with these two tag recommenders demonstrated the efficiency of FoldCons.

The remainder of this paper is organized as follows. In Section 2 we present some preliminaries and describe briefly PITF and STRec. Section 3 details the FoldCons method. In Section 4, we present experimentations of our proposal. Finally Section 5 summarizes the related work while Section 6 concludes the paper.

2. PRELIMINARIES

A folksonomy is a system of classification that allows its users creating and managing tags to annotate and categorize content. It is related to the event of social tagging systems. A folksonomy can be defined as a collection of a set of users U, set of tags T, set of items I, and a ternary relation between them $S \subseteq U \times I \times T$. A tagging triple $(u, i, t) \in S$ means that user u has tagged an item i with the tag t. A user can tag an item with one or more distinctive tags from T. We assume that a user can tag an item with a given tag at most once.

The interest of a tag t for a given user u and an item i is generally estimated by a score $score(t|u, i)$. Thus the purpose of a tag recommender is to compute the top-K highest scoring tags for a post (u, i) what represents its recommendations.

$$Top(u,i,K) = \underset{t\in T}{\overset{K}{argmax}}\ score(t|u,i) \tag{1}$$

In the next subsections we describe how PITF and STRec model the scores of tags.

2.1 Factor Models for Tag Recommendation

Factorization models are known to be among the best performing models. They are a very successful class of models for recommender systems. For tag recommendation they outperform the other approaches like Folkrank and adapted Pagerank [8]. We chose the pairwise interaction tensor factorization model (PITF) of Rendle and Schmidt-Thieme in our experimentations due to its efficency [7]. It took the first place of the task 2 of ECML PKDD Discovery Challenge 2009 indeed.

PITF proposes to infer pairwise ranking constraints from S. The idea is that within a post (u,i), one can assume that a tag t is preferred over another tag t' iff the tagging triple (u,i,t) has been observed and (u,i,t') has not been observed. PITF captures the interactions between users and tags as well as between items and tags. Its model equation is given by:

$$score(t|u,i) = \sum_f \hat{u}_{u,f} \cdot \hat{t}^U_{t,f} + \sum_f \hat{i}_{i,f} \cdot \hat{t}^I_{t,f} \tag{2}$$

Where $\hat{U}$, $\hat{I}$, $\hat{T}^U$ and $\hat{T}^I$ are feature matrices capturing the latent interactions. For more information regarding PITF see their paper [7].

2.2 A Network-based Tag Recommender

We also used STRec, an adaptation to tag recommendation of the network-aware search in online social bookmarking applications of [6]. We chose it as a candidate for network-based tag recommenders. STRec is fast and efficient as presented in [6]. It considers that users form an undirected weighted graph $G = (U, E, \sigma)$ (i.e. the social network) where σ is a function that associates to each edge $e = (u,v) \in E$ a value in $[0,1]$, called the proximity between u and v. Its model score of a tag t for a post (u,i) is represented by

$$score(t|u,i) = h(fr(t|u,i)) \tag{3}$$

where $fr(t|u,i)$ is the overall frequency of tag t for a user u and item i, and h a positive monotone function. In our case we took h as the identity function. They define the overall tag's frequency function $fr(t|u,i)$ as a combination of a user-network-dependent component $sf(t|u,i)$ and an item-dependent one $tf(t,i)$, and as follows:

$$fr(t|u,i) = \alpha \times tf(t,i) + (1-\alpha) \times sf(t|u,i) \tag{4}$$

The former component, $tf(t,i)$, is the frequency of t for item i, i.e., the number of times the item was tagged with this tag. The latter component stands for social frequency, an measure that depends on the neighborhood of user u. The parameter α allows to tune the relative importance of each component.

The scoring model of STRec does not take into account only the neighbors directly connected to the user. But it deals also with users indirectly connected to her, following a natural interpretation that user links (e.g., similarity) are, at least to some extent, transitive. Thus considering that each neighbour brings her own weight (proximity) to the score of a tag, the measure of tag's social frequency is defined as follows:

$$sf(t|u,i) = \sum_{v\in\{U|(v,i,t)\in S\}} \sigma(u,v) \tag{5}$$

As one may notice, STRec does not regard the use of the tag by the user for tagging items (as for the item with the tag's frequency) but it considers the opinions of the user's neighbours instead.

3. FOLDING MORE CONSISTENCY IN RECOMMENDATIONS

In this section, we present our add-on method for improving tag recommendation, we called *FoldCons*. Its functioning is to ask recommendations of a given tag recommender about a post (u,i), and improve them before to leave to the user the final top-K recommended tags. Therefore it asks for a number of tags greater than K. Then it re-ranks them and keeps the K first tags as the final recommendations. The sequel of this section details the FoldCons method besides introducing some defintions.

DEFINITION 1. *A tag's users list $U(t)$ is the set of users who used the tag t. A tag's items list $I(t)$ so is the set of items tagged by the tag t.*

DEFINITION 2. *The pairwise confidence measure, $PCM(t \rightarrow t')$, is defined between to tags t and t'. It determines to some extent the interest to use t' in addition to t. PCM takes into account both users and items as defined as follows*

$$PCM(t \rightarrow t') = \frac{|U(t)\cap U(t')|}{|U(t)|} + \frac{|I(t)\cap I(t')|}{|I(t)|} \tag{6}$$

The pairwise confidence measure mines association rules between tags from two dimensions: users and items. This allows us to account the frequency of tags' co-occurences both for the user and item of a post. Let us notice there we do not currently weight their contributions in the sum but it is a possibility.

3.1 FoldCons' functioning

FoldCons works simply as an add-on tool which takes in entry tags from a tag recommender sorted by their scores and returns a short list of final recommended tags. Its challenge is to improve the recommendations it received by giving a better top-K.

Let us denote by D the list of recommended tags received from a given tag recommender. To simply we consider that D is sorted and its highest scoring tag is $D[1]$. Let us emphasize here we assign more attention to $D[1]$ than the rest of tags in D due to the fact that it is the best choice given by the tag recommender. Therefore we fix it and compute the pairwise confidence measures of all the tags compared to it. Then we sort D again with the new scoring function for each of its tags t

$$score^{PCM}(t|u,i) = (1 + PCM(D[1] \rightarrow t)) \cdot score(t|u,i)$$

Thus we introduce a certain consistency in the recommendations by taking account the tags which appear generally next to the first recommended tag in D from both the user's point of view and the one of the item. This approach improves noticeably the quality of recommendations as shown in our experimentations. Morever, as PCM at best doubles the initial score of a tag (i.e. $PCM(t \rightarrow t') \in [0,1]$), we can keep in entry only the tags whose scores exceed or equal the half-score of the last tag in the top-K of D. Indeed the other tags can not change the top-K.

3.2 Still ensuring better recommendations

Some tag recommenders are not both user and item-centred. STRec may be an example. Despite it takes into account the opinions of a user's neighbourhood, it does not consider the user's frequent used tags. In these cases we experimented that the application of FoldCons may slightly fall to improve the recommendations in the user or item-dimension. Thus we adapted FoldCons to these cases. Depending on the recommender we consider the user profile and/or the item one, we define below.

DEFINITION 3. *A user profile $T(u)$ is the set of all the tags used by user u to tag items. An item profile $T(i)$ so is the set of tags used to annotate item i.*

We determine if FoldCons brings better recommendations by estimating its contribution. We take account of the number of common tags between the top-K recommended tags and the item and/or the user profile before and after its computation. The difference represents the contribution of FoldCons.The recommendations of FoldCons are considered better when its contributions are positive else the primarily list of tags D remains unchanged. This approach ensures, in almost all cases, that the recommendation quality does not decrease after application of FoldCons when recommenders are not both user and item-centred.

4. EXPERIMENTATIONS

Due to the length of short paper, four pages, we expose only there the ability of FoldCons to improve recommendations. More results are available at [3]. We discussed there subjects like the pertinence of using just the first tag as reference and not another tag neither several tags at the same time.

4.1 Datasets

We chose five datasets from four online systems: del.icio.us[1], Movielens[2], Last.fm[3], and BibSonomy[4].

We take the ones of del.icio.us, movielens, and last.fm from *HetRec 2011* [2] and the two other ones from Bibsonomy: a post-core at level 5 and a one at level 2 [1, 4]. We call them respectively *Bibson*5 and *dc*09).

*dc*09 is the one of the task 2 of ECML PKDD Discovery Challenge 2009[5]. This task was especially intended for methods relying on a graph structure of the training data only. The user, item, and tags of each post in the test data are all contained in the training data's, a post-core at level 2.Let us remaind that a post-core at level p is a subset of a folksonomy with the property, that *each user, tag and item has/occurs in at least p times.*Table 1 presents the caracteristics of these datasets.

[1]http://www.del.icio.us.com
[2]http://www.grouplens.org
[3]http://www.lastfm.com
[4]http://www.bibsonomy.org
[5]http://www.kde.cs.uni-kassel.de/ws/dc09/

Table 1: Caracteristics of the datasets

dataset	$\lvert U\rvert$	$\lvert I\rvert$	$\lvert T\rvert$	$\lvert T(u,i)\rvert$
Bibson5	116	361	412	2,526
dc09	1,185	22,389	13,276	64,406
del.icio.us	1,867	69,226	53,388	104,799
Last.fm	1,892	17,632	11,946	71,065
Movielens	2,113	10,197	13,222	27,713

4.2 Evaluation Measures and Methodology

To evaluate our proposal, we used a variant of the leave-one-out hold-out estimation called LeavePostOut [4]. In all datasets except *dc*09, we picked randomly and for each user u, one item i, which he had tagged before. Thus we create a test set and a training one. The task of our recommender was then to predict the tags the user assigned to the item.

Moreover we generate, for each training set, a social network by computing the Dice coefficient of common users' tagged items. Let us notice that we fixed the parameter α of STRec to 0.05 for all experimentations. We kept this value after a calibration over the dataset dc09. What is of course not necessary optimal for all the others. For Tagrec we keep the default parameters given by the authors but with 2,000 iterations[6].
We used the F1-measure as performance measure.

4.3 Results

In this section we present the some results of our experimentations on the datasets. On each of them, we run STRec and Tagrec. Then we apply FoldCons on their proposed top-K tags. We call respectively by STRec++ and Tagrec++ the application of FoldCons on them. Let us notice that for STRec we specially apply the adapted FoldCons presented in Section 3.2.

4.3.1 Contribution of FoldCons

The tables below show the gains brought by FoldCons when it is applied. We compute the top-*5* to top-*10* recommended tags and their F1-measures.

Table 2: The benefits of FoldCons on dc09

#tags	5	6	7	8	9	10
tagrec	0.296	0.286	0.272	0.258	0.246	0.236
tagrec++	0.301	0.290	0.279	0.265	0.251	0.241
Gain (%)	**1.68**	**1.35**	**2.74**	**2.52**	**2.16**	**2.23**
STRec	0.305	0.302	0.298	0.291	0.286	0.282
STRec++	0.309	0.312	0.306	0.297	0.292	0.285
Gain (%)	**1.56**	**3.25**	**2.55**	**1.91**	**2.13**	**1.18**

5. RELATED WORK

Finding suited tags to put in the same recommendations is an important point for tag recommendation. Many approaches and methods can be used to achieve this point. We can cite the work of Lipczak which focused on content-based

[6]http://www.informatik.uni-konstanz.de/rendle/software/tag-recommender/

Table 3: The benefits of FoldCons on del.icio.us

#tags	5	6	7	8	9	10
tagrec	0.188	0.182	0.173	0.165	0.157	0.151
tagrec++	0.191	0.185	0.177	0.169	0.162	0.154
Gain (%)	**1.62**	**2.09**	**2.32**	**2.63**	**2.67**	**2.40**
STRec	0.103	0.105	0.107	0.107	0.108	0.109
STRec++	0.108	0.110	0.111	0.111	0.111	0.111
Gain (%)	**5.64**	**4.73**	**3.93**	**3.63**	**3.02**	**2.29**

Table 4: The benefits of FoldCons on last.fm

#tags	5	6	7	8	9	10
tagrec	0.328	0.309	0.290	0.272	0.256	0.242
tagrec++	0.333	0.314	0.295	0.278	0.261	0.246
Gain (%)	**1.61**	**1.86**	**1.95**	**1.95**	**1.81**	**1.76**
STRec	0.274	0.260	0.246	0.235	0.224	0.215
STRec++	0.277	0.262	0.248	0.237	0.225	0.216
Gain (%)	**1.15**	**0.80**	**0.99**	**0.87**	**0.80**	**0.58**

Table 5: The benefits of FoldCons on bibson5

#tags	5	6	7	8	9	10
tagrec	0.449	0.426	0.409	0.390	0.371	0.353
tagrec++	0.450	0.429	0.412	0.390	0.373	0.357
Gain (%)	**0.30**	**0.58**	**0.67**	**0.05**	**0.65**	**1.07**
STRec	0.389	0.373	0.360	0.349	0.340	0.334
STRec++	0.397	0.379	0.364	0.352	0.343	0.337
Gain (%)	**2.00**	**1.80**	**0.97**	**0.97**	**0.88**	**0.94**

Table 6: The benefits of FoldCons on movielens

#tags	5	6	7	8	9	10
tagrec	0.163	0.148	0.135	0.124	0.115	0.108
tagrec++	0.164	0.148	0.136	0.125	0.116	0.108
Gain (%)	**0.70**	**0.34**	**0.70**	**0.67**	**0.69**	**0.67**
STRec	0.146	0.138	0.131	0.127	0.122	0.119
STRec++	0.148	0.140	0.133	0.128	0.124	0.120
Gain (%)	**1.57**	**1.46**	**1.56**	**1.08**	**0.98**	**1.06**

tag recommenders [5]. His approach consists in extracting basic tags from the content of items (e.g. the item title), then extending the set of potential recommendations by related tags proposed by a lexicon based on co-occurrences of tags within item's posts.

Wang et al. did a similar work enough but first applied a TF-IDF algorithm on the description of the item content, in order to extract keywords of the item [9]. Based on the top keywords, they utilize association rules from history records in order to find the most probable tags to recommend. In addition, history information is also exploited to find the most appropriate recommendations.

Many others works could be cited. However due to the length of the paper we can not cite them and furthermore they are generally closed approaches. They are for the most part content-dependent. What is not the case of FoldCons which mines association rules directly on a primary list of candidate tags. Our experimentations showed the effectiveness of this method.

6. CONCLUSION

We proposed an add-on method to improve the recommendations of tag recommender. We mine association rules on top of their recommendations, then we sort them again thanks to their confidence scores compared to the first tag. Thus we introduce a certain consistency inside the recommendations by taking account of tags which appear generally next to the first candidate tag in the initial recommended list.

This method improves up to 5% the recommendations as shown by our experimentations when the users have not in average a small number of posts.

7. ACKNOWLEDGEMENTS

We would like to thank Steffen Rendle for the software *tagrec* and also the Knowledge and Data Engineering Group of University of Kassel especially for the Benchmark Folksonomy Data (BibSonomy, version of April 30th, 2007).

8. REFERENCES

[1] D. Benz, A. Hotho, R. Jäschke, B. Krause, F. Mitzlaff, C. Schmitz, and G. Stumme. The social bookmark and publication management system BibSonomy. *The VLDB Journal*, 19(6):849–875, Dec. 2010.

[2] I. Cantador, P. Brusilovsky, and T. Kuflik. 2nd workshop on information heterogeneity and fusion in recommender systems (hetrec 2011). In *Proceedings of the 5th ACM conference on Recommender systems*, RecSys 2011, New York, NY, USA, 2011. ACM.

[3] M. Gueye, T. Abdessalem, and H. Naacke. Improving tag recommendation by folding in more consistency. `http://www.infres.enst.fr/~mogueye/foldCons13.pdf`, Sept. 2013.

[4] R. Jäschke, L. Marinho, A. Hotho, L. Schmidt-Thieme, and G. Stumme. Tag recommendations in social bookmarking systems. *AI Commun.*, 21(4):231–247, Dec. 2008.

[5] M. Lipczak. Tag recommendation for folksonomies oriented towards individual users. *ECML PKDD discovery challenge*, 84, 2008.

[6] S. Maniu and B. Cautis. Efficient top-k retrieval in online social tagging networks. *arXiv preprint arXiv:1104.1605*, 2011.

[7] S. Rendle and L. 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.

[8] P. Symeonidis, A. Nanopoulos, and Y. Manolopoulos. Tag recommendations based on tensor dimensionality reduction. In *Proceedings of the 2008 ACM conference on Recommender systems*, RecSys '08, pages 43–50, New York, NY, USA, 2008. ACM.

[9] J. Wang, L. Hong, and B. D. Davison. Rsdc '09: Tag recommendation using keywords and association rules. In F. Eisterlehner, A. Hotho, and R. Jäschke, editors, *ECML PKDD Discovery Challenge 2009 (DC09)*, volume 497, pages 261–274, Bled, Slovenia, September 2009. CEUR Workshop Proceedings.