

Content-based Recommendation in Social Tagging Systems

Iván Cantador, Alejandro Bellogín, David Vallet

Departamento de Ingeniería Informática

Universidad Autónoma de Madrid

Campus de Cantoblanco

28049, Madrid, Spain

{ivan.cantador, alejandro.bellogin, david.vallet}@uam.es

ABSTRACT

We present and evaluate various content-based recommendation models that make use of user and item profiles defined in terms of weighted lists of social tags. The studied approaches are adaptations of the Vector Space and Okapi BM25 information retrieval models. We empirically compare the recommenders using two datasets obtained from Delicious and Last.fm social systems, in order to analyse the performance of the approaches in scenarios with different domains and tagging behaviours.

Categories and Subject Descriptors

H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval – *information filtering, retrieval models.*

General Terms

Algorithms, Experimentation, Performance.

Keywords

Recommender systems, personalization, social tagging, folksonomy.

1. INTRODUCTION

In recent years, we have witnessed an unexpected success and increasing proliferation of social tagging systems. In these systems, users create or upload content (items), annotate it with freely chosen words (tags), and share it with other users. The whole set of tags constitutes an unstructured collaborative classification scheme that is commonly known as *folksonomy*. This implicit classification is then used to search for and discover items of interest.

In such systems, users and items can be assigned profiles defined in terms of weighted lists of social tags. In general, users annotate items that are relevant for them, so the tags they provide can be assumed to describe their interests, tastes and needs. Moreover, it can be also assumed that the more a tag is used by a certain user, the more important that tag is for her. Analogously, tags assigned to items usually describe their contents. The more users annotate a certain item with a particular tag, the better that tag describes the item contents. The previous assumptions, however, have to be carefully taken into account. Tags used very often by users to annotate many items may not be useful to discern informative user preferences and items features [8]. We address this issue herein.

Adomavicius and Tuzhilin [1] formulate the recommendation

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

RecSys'10, September 26–30, 2010, Barcelona, Spain.

Copyright 2010 ACM 978-1-60558-906-0/10/09...\$10.00.

problem as follows. Let $\mathcal{U} = \{u_1, \dots, u_M\}$ be a set of users, and let $\mathcal{I} = \{i_1, \dots, i_N\}$ be a set of items. Let $g: \mathcal{U} \times \mathcal{I} \rightarrow \mathcal{R}$, where \mathcal{R} is a totally ordered set, be a utility function such that $g(u_m, i_n)$ measures the gain of usefulness of item i_n to user u_m . Then, for each user $u \in \mathcal{U}$, we want to choose items $i^{\max, u} \in \mathcal{I}$, unknown to the user, which maximise the utility function g :

$$\forall u \in \mathcal{U}, \quad i^{\max, u} = \arg \max_{i \in \mathcal{I}} g(u, i)$$

In content-based recommendation approaches, g is formulated as:

$$g(u_m, i_n) = \text{sim}(\text{ContentBasedUserProfile}(u_m), \text{Content}(i_n)) \in \mathcal{R}$$

where $\text{ContentBasedUserProfile}(u_m) = \mathbf{u}_m = (u_{m,1}, \dots, u_{m,K}) \in \mathbb{R}^k$ is the content-based preferences of user u_m , i.e., the item content features that describe the interests, tastes and needs of the user, and $\text{Content}(i_n) = \mathbf{i}_n = (i_{n,1}, \dots, i_{n,K}) \in \mathbb{R}^k$ is the set of content features characterising item i_n . These descriptions are usually represented as vectors of real numbers (weights) in which each component measures the “importance” of the corresponding feature in the user and item representations. The function *sim* computes the similarity between a user profile and an item profile in the content feature space.

From the previous formulations, in this paper, we consider social tags as the content features that describe both user and item profiles, and we investigate two issues. First, we study several weighting schemes to measure the “importance” of a given tag for each user and item. Some of these weighting schemes are based on single profiles, while others make use of the whole folksonomy. Second, for each tag weighting scheme, we propose and evaluate a number of similarity functions. The studied approaches are adaptations of the well known Vector Space and Okapi BM25 information retrieval models [2]. By using two datasets obtained from Delicious and Last.fm social systems, we analyse the performance of the approaches in scenarios with different domains and tagging behaviours.

The rest of the paper has the following structure. Section 2 introduces definitions and notation used in the studied profile and recommendation approaches. Sections 3 and 4 present such approaches. Section 5 describes the datasets utilised to evaluate the recommendation models, while Section 6 explains the conducted experiments and obtained results. Finally, Section 7 ends with some conclusions and future research lines.

2. PRELIMINARIES

A folksonomy \mathcal{F} can be defined as a tuple $\mathcal{F} = \{\mathcal{T}, \mathcal{U}, \mathcal{I}, \mathcal{A}\}$, where $\mathcal{T} = \{t_1, \dots, t_L\}$ is the set of tags that comprise the vocabulary expressed by the folksonomy, $\mathcal{U} = \{u_1, \dots, u_M\}$ and $\mathcal{I} = \{i_1, \dots, i_N\}$ are respectively the set of users and the set of items that annotate and are annotated with the tags of \mathcal{T} , and $\mathcal{A} = \{(u_m, t_l, i_n)\} \in \mathcal{U} \times \mathcal{T} \times \mathcal{I}$ is the set of assignments (annotations) of each tag t_l to an item i_n by a user u_m .

Based on this notation, the simplest way to define the profile of user u_m is as a vector $\mathbf{u}_m = (u_{m,1}, \dots, u_{m,L})$, where $u_{m,l} = |\{(u_m, t_l, i) \in \mathcal{A} | i \in \mathcal{I}\}|$ is the number of times the user has annotated items with tag t_l . Similarly, the profile of item i_n can be defined as a vector $\mathbf{i}_n = (i_{n,1}, \dots, i_{n,L})$, where $i_{n,l} = |\{(u, t_l, i_n) \in \mathcal{A} | u \in \mathcal{U}\}|$ is the number of times the item has been annotated with tag t_l .

In this work, we extend the previous definitions of user and item profiles by using different expressions for the vector component weights. These expressions are explained in Section 4.

The proposed user and item profiles are then exploited by a number of different content-based recommendation models. These recommenders are adaptations of the well known Vector Space and Okapi BM25 ranking models [2], and are described in detail in Section 3.

For a better understanding of both user/item profiles and content-based recommenders, Table 1 gathers the definition of common elements appearing in the profile and recommendation models.

Table 1. Definition of elements used in the proposed profile and recommendation models.

Element	Definition
User-based tag frequency	$tf_{u_m}(t_l)$ Number of times user u_m has annotated items with tag t_l
Item-based tag frequency	$tf_{i_n}(t_l)$ Number of times item i_n has been annotated with tag t_l
User-based inverse tag frequency	$iuf(t_l) = \log \frac{M}{n_u(t_l)}$, $n_u(t_l) = \{u_m \in \mathcal{U} u_{m,l} > 0\} $
Item-based inverse tag frequency	$iif(t_l) = \log \frac{N}{n_i(t_l)}$, $n_i(t_l) = \{i_n \in \mathcal{I} i_{n,l} > 0\} $
User profile size	$ u_m = \sum_{l=1}^L u_{m,l}$
Item profile size	$ i_n = \sum_{l=1}^L i_{n,l}$

3. USER AND ITEM PROFILES

In this section, we present different schemes to weight the components of tag-based user and items profiles. Some of them are based on the information available in individual profiles, while others consider information from the whole folksonomy.

3.1 TF Profile Model

The simplest approach for assigning a weight to a particular tag in a user or item profile is by counting the number of times such tag has been used by the user or the number of times the tag has been used by the community to annotate the item. Thus, our first profile model for user u_m consists of a vector $\mathbf{u}_m = (u_{m,1}, \dots, u_{m,L})$, where

$$u_{m,l} = tf_{u_m}(t_l)$$

Similarly, the profile of item i_n is defined as a vector $\mathbf{i}_n = (i_{n,1}, \dots, i_{n,L})$, where

$$i_{n,l} = tf_{i_n}(t_l)$$

3.2 TF-IDF Profile Model

In an information retrieval environment, common keywords that appear in many documents of a collection are not informative, and may not allow distinguishing relevant documents for a given query. To take this into account, TF-IDF weighting scheme is usually applied to the document profiles [2].

We adopt that principle, and adapt it to social tagging systems, proposing a second profile model, defined as:

$$u_{m,l} = tfiuf_{u_m}(t_l) = tf_{u_m}(t_l) \cdot iuf(t_l),$$

$$i_{n,l} = tfiif_{i_n}(t_l) = tf_{i_n}(t_l) \cdot iif(t_l).$$

3.3 BM25 Profile Model

As an alternative to TF-IDF, Okapi BM25 weighting scheme follows a probabilistic approach [2] to assign a document with a ranking score given a query.

We propose an adaptation of such model by assigning each tag with a score (weight) given a certain user or item. Our third profile model has the following expressions:

$$\begin{aligned} u_{m,l} &= bm25_{u_m}(t_l) = \\ &= \frac{u_{m,l}^{(k_1+1)}}{u_{m,l} + k_1(1-b+b \cdot |u_m| / avg(|u_m|))} \cdot iuf(t_l), \\ i_{n,l} &= bm25_{i_n}(t_l) = \\ &= \frac{i_{n,l}^{(k_1+1)}}{i_{n,l} + k_1(1-b+b \cdot |i_n| / avg(|i_n|))} \cdot iif(t_l), \end{aligned}$$

where b and k_1 are set to the standard values of 0.75 and 2, respectively.

4. CONTENT-BASED RECOMMENDERS

In this section, we describe a number of content-based recommendation models that are defined as similarity measures between user and item profiles introduced in Section 3. Two of these models are state of the art approaches [5][8], and others were investigated by the authors in the context of personalised Web search [7].

4.1 TF-based Similarity

To compute the preference of a user for an item, Noll and Meinel [5] propose a personalised similarity measure based on the user's tag frequencies:

$$g(u_m, i_n) = tf_u(u_m, i_n) = \frac{\sum_{l: i_{n,l} > 0} tf_{u_m}(t_l)}{\max_{u \in \mathcal{U}, t \in \mathcal{T}} (tf_u(t))}$$

The model utilises the user's usage of tags appearing in the item profile, but does not take into account their weights in such profile. In the formula, we introduce a normalisation factor that scales the utility function to values in the range [0,1], without altering the user's item ranking.

To measure the impact of personalisation in Noll and Meinel's approach, we propose a similarity measure based on the tag frequencies in the item profiles:

$$g(u_m, i_n) = tf_i(u_m, i_n) = \frac{\sum_{l: u_{m,l} > 0} tf_{i_n}(t_l)}{\max_{i \in \mathcal{I}, t \in \mathcal{T}} (tf_i(t))}$$

4.2 TF Cosine-based Similarity

A direct extension of Noll and Meinel's approach is to exploit the weights of both user and item profiles by computing the cosine between their vectors as similarity measure:

$$g(u_m, i_n) = \text{cos}_{tf}(u_m, i_n) = \frac{\sum_l tf_{u_m}(t_l) \cdot tf_{i_n}(t_l)}{\sqrt{\sum_l (tf_{u_m}(t_l))^2} \cdot \sqrt{\sum_l (tf_{i_n}(t_l))^2}}$$

4.3 TF-IDF Cosine-based Similarity

Xu et al. [8] use the cosine similarity measure to compute the similarity between user and item profiles. As profile component weighting scheme, they use TF-IDF¹. Following our notation, their approach can be defined as follows:

$$g(u_m, i_n) = \text{cos}_{tf-idf}(u_m, i_n) = \frac{\sum_l tf_{u_m}(t_l) \cdot iuf(t_l) \cdot tf_{i_n}(t_l) \cdot iif(t_l)}{\sqrt{\sum_l (tf_{u_m}(t_l) \cdot iuf(t_l))^2} \cdot \sqrt{\sum_l (tf_{i_n}(t_l) \cdot iif(t_l))^2}}$$

4.4 BM25-based Similarity

Analogously to the similarity based on tag frequencies described in Section 4.1, but using a BM25 weighting scheme, we propose a couple of similarity functions that only take into account the weights of either the user profile or the item profile. These two recommendation models are:

$$g(u_m, i_n) = \text{bm25}_u(u_m, i_n) = \sum_{(l|i_{n,l}>0)} \text{bm25}_{u_m}(t_l)$$

$$g(u_m, i_n) = \text{bm25}_i(u_m, i_n) = \sum_{(l|u_{m,l}>0)} \text{bm25}_{i_n}(t_l)$$

4.5 BM25 Cosine-based Similarity

Xu et al. [8] also investigate the cosine similarity measure with a BM25 weighting scheme. They use that model on personalised Web Search. We adapt and define it for social tagging as follows:

$$g(u_m, i_n) = \text{cos}_{\text{bm25}}(u_m, i_n) = \frac{\sum_l (\text{bm25}_{u_m}(t_l) \cdot \text{bm25}_{i_n}(t_l))}{\sqrt{\sum_l (\text{bm25}_{u_m}(t_l))^2} \cdot \sqrt{\sum_l (\text{bm25}_{i_n}(t_l))^2}}$$

5. DATASETS

In order to evaluate the presented tag-based recommendation models under different domain and tagging conditions, we run them using datasets obtained from two different social systems: Delicious and Last.fm. Delicious is a social bookmarking site for Web pages. By the end of 2008, the service claimed more than 5.3 million users and 180 million unique bookmarked URLs. On the other hand, Last.fm is an on-line radio site for music. By the beginning of 2009, it claimed over 40 million active users and 7 million tracks.

As collaborative social tagging platforms, Delicious differs from Last.fm in the fact that it contains tagged items (Web pages) belonging to practically any domain, while Last.fm tagged items (tracks) belong to the music domain. Moreover, the users' tagging behaviour is also different in both systems. As shown in Table 2, in Delicious dataset, the average number of tags per user is greater than in Last.fm dataset. However, taking into account the tags provided by the entire community of users, a track in Last.fm receives more tags than a Web page in Delicious. This apparent contradiction can be explained through the inverse relation between the numbers of users and items registered in such systems.

¹ Xu et al. do not specify if they use user-based or item-based inverse tag frequencies, or both. We chose to use both, since this configuration gave the best performance values.

Table 2. Description of the used datasets.

	Delicious	Last.fm
#users	1,000	1,000
#items	84,005	50,202
#tags	42,324	16,687
Avg. #items/user	95	66
Avg. #tags/user	480	149
Avg. #tags/item (per user)	5	2
Avg. #tags/item (in the community)	34	49

5.1 Delicious Dataset

We created a dataset formed by 1,000 Delicious users. These users were chosen as follows. First, we randomly selected 50 users who bookmarked the top Delicious bookmarks on 14th May 2009 and had at least 20 bookmarks in their profiles. Then, we extended this set of users through their social network in Delicious. A maximum distance of 2 user contacts was allowed in such extension. Due to limitations of Delicious API, we only extracted the latest 100 bookmarks of each user. The final dataset contained 84,005 different bookmarks and 42,324 distinct tags. On average, each user profile had 95 bookmarks and 480 distinct tags.

5.2 Last.fm Dataset

In the case of Last.fm dataset, we aimed to obtain a representative set of users, covering all music genres. We first identified the most popular tags related to the different music genres in Last.fm. Then, we used Last.fm API to get the top artists tagged with the previous tags. For each artist, we gathered his/her fans along with their direct friends. Finally, we retrieved all tags and tagged tracks of the user profiles. The final dataset contained 50,202 different tracks and 16,687 distinct tags. On average, each user profile had 66 tracks and 149 distinct tags.

6. EXPERIMENTS

In this section, we explain the experiment methodology we followed to evaluate the described recommendation models, and present the obtained results of that evaluation.

6.1 Methodology

Figure 1 depicts the followed experimental methodology.

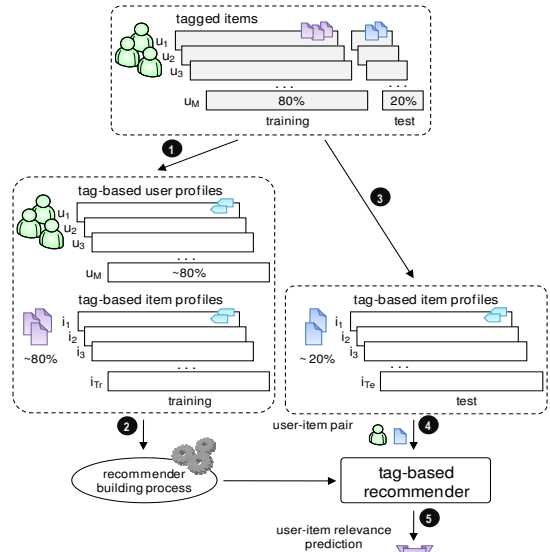


Figure 1. Description of the followed experimental methodology.

Table 3. Average results obtained by the content-based recommendation approaches. All result differences are statistically significant (Wilcoxon, $p < 0.05$) except those among the results marked with †.

	Delicious					Last.fm				
	P@5	P@10	P@20	MAP	NDCG	P@5	P@10	P@20	MAP	NDCG
tf_u	0.028	0.021	0.014	0.011	0.085	0.063	0.048	0.034	0.057	0.163
tf_i	0.001	0.001	0.000	0.001	0.027	0.040	0.029	0.020	0.037	0.120
cos_{tf}	0.234	0.109	0.059	0.041	0.202	0.104	0.076	0.054	0.084	0.225 [†]
cos_{tf-idf}	0.292	0.212	0.144	0.115	0.350	0.148	0.111	0.079	0.118	0.278
$bm25_u$	0.226	0.105	0.075	0.051	0.216	0.098	0.076	0.058	0.080	0.220 [†]
$bm25_i$	0.062	0.047	0.035	0.020	0.141	0.107	0.077	0.054	0.088	0.219 [†]
cos_{bm25}	0.364	0.260	0.172	0.145	0.390	0.166	0.126	0.091	0.132	0.297

We randomly split the set of items tagged by the users in the database in two subsets. The first subset contained 80% of the items for each user, and was used to build the recommendation models (training). The second subset contained the remaining 20% of the items, and was used to evaluate the recommenders (test).

Specifically, we built the recommendation models with the whole tag-based profiles of the training items, and with those parts of the users' tag-based profiles formed by tags annotating the training items. We evaluated the recommenders with the tag-based profiles of the test items. In the evaluation, we computed several metrics (see Section 6.2), and performed a 5-fold cross validation procedure.

6.2 Metrics

We assumed a content retrieval scenario where the system provides the user a list of N recommended items based on her content-based profile. To evaluate the performance of each recommender, we took into account the percentage and ranking of relevant items appearing in the provided lists. For that purpose, we computed three metrics often used to evaluate information retrieval systems [2]: Precision at the top N ranked results ($P@N$), Mean Average Precision (MAP), and Discounted Cumulative Gain (DCG).

Precision is defined as the number of retrieved relevant documents divided by the total number of retrieved documents. MAP is a precision metric that emphasises ranking relevant documents higher. Finally, DCG measures the usefulness of a document based on its position in a result list. In our evaluation framework, retrieved documents were all the items belonging to each test set (see Section 6.1). Thus, the whole test set contained items belonging to the active user's profile (relevant documents), and items from other users' profiles (assumed as non relevant documents for the active user).

6.3 Results

Table 2 shows the results obtained in the evaluation of the recommendation models using Delicious and Last.fm datasets. In general, as expected, the models focused on user profiles (tf_u , $bm25_u$) outperformed the models oriented to item profiles (tf_i , $bm25_i$). This is not true with BM25 model in Last.fm. We believe this is due to the small size of user profiles in that dataset.

Regarding cosine-based models, by performing a weighting scheme that exploits the whole folksonomy (cos_{tf-idf} , cos_{bm25}), we clearly enhance the classic frequency profile representation (cos_{tf}). Note that even $bm25_u$ outperforms cos_{tf} in Delicious dataset. Thus, it seems that those tags appearing in many user and item profiles have to be penalised, since they are not informative to discern relevant user preferences and item characteristics. In the same context, BM25 was better than TF-IDF weighting scheme. This could be explained by the fact that, in social tagging systems, most popular

tags should be penalised more carefully as they usually describe the item contents more precisely (through community consensus).

Comparing results from Delicious and Last.fm, we obtained higher precision values in the former. In Last.fm, users listen to music and do not always tag their favourite tracks. In contrast, the main use of Delicious is to bookmark and tag Web pages for organisational and searching purposes. Thus, user profiles in Delicious are larger than in Last.fm, and content-based recommendation performs better.

7. CONCLUSIONS

In this paper, we have evaluated a number of content-based recommendation models that make use of user and item profiles described in terms of weighted lists of social tags. The studied approaches are adaptations of the Vector Space and Okapi BM25 ranking models [2]. The presented work is only the beginning of our exploration of how the above and other information retrieval models could be applied in social recommender systems. We plan to extend our analysis in two directions. First, we want to study alternative tag-based profile and recommendation models. Specifically, as proposed in [6], we shall investigate the application of tag clustering techniques for user profiling. Second, we do not want to restrict our research to content-based recommenders. We shall investigate adaptations of collaborative filtering and hybrid recommendation strategies, as done for example in [3][4][9].

8. REFERENCES

- [1] Adomavicius, G., Tuzhilin, A. 2005. Toward the Next Generation of Recommender Systems: A Survey and Possible Extensions. *IEEE Transactions on Knowledge & Data Engineering*, 17(6), 734-749.
- [2] Baeza-Yates, R., Ribeiro-Neto, B. 1999. Modern Information Retrieval. Addison Wesley.
- [3] Hotho, A., Jäschke, R., Schmitz, C., Stumme, G. 2006. Information Retrieval in Folksonomies: Search and Ranking. *In Proceedings of ISWC 2006*, 411-426.
- [4] Niwa, S., Doi, T., Honiden, S. 2006 Web Page Recommender System based on Folksonomy Mining for ITNG'06 Submissions. *In Proceedings of ITNG 2006*, 388-393.
- [5] Noll, M. G., Meinel, C. 2007. Web Search Personalization via Social Bookmarking and Tagging. *In Proceedings of ISWC 2007*, 367-380.
- [6] Shepitsen, A., Gemmell, J., Mobasher, B., Burke, R. 2008. Personalized Recommendation in Social Tagging Systems using Hierarchical Clustering. *In Proceedings of RecSys 2008*, 259-266.
- [7] Vallet, D., Cantador, L., Jose, J. M. 2010. Personalizing Web Search with Folksonomy-Based User and Item Profiles. *In Proceedings of ECIR 2010*, 420-431.
- [8] Xu, S., Bao, S., Fei, B., Su, Z., Yu, Y. 2008. Exploring Folksonomy for Personalized Search. *In Proceedings of SIGIR 2008*, 155-162.
- [9] Zanardi, V., Capra, L. 2008. Social Ranking: Uncovering Relevant Content using Tag-based Recommender Systems. *In Proceedings of RecSys 2008*, 51-58.