Incremental Tag-Aware User Profile Building to Augment Item Recommendations

Ho Keung Tsoi
Department of Computer Science
Hong Kong Baptist University
Kowloon Tong, Hong Kong SAR
China
hktsoi@comp.hkbu.edu.hk

Li Chen
Department of Computer Science
Hong Kong Baptist University
Kowloon Tong, Hong Kong SAR
China
lichen@comp.hkbu.edu.hk

ABSTRACT

Folksonomic system allows users to use tags to describe items. These tags do not just exist in the form of textual description, and they actually bear more meaning underneath, such as user preference. In this paper, we first show the distribution of preferences and semantic categories across a folksonomic system, and then develop a hybrid design to cope with the cold-start problem.

Specifically, we speculate that the semantic categories formed in users' perspective and in items' perspective are different. They represent different preferences and meaning and are believed to be crucial in recommender algorithm design. Through a dimensionality reduction technique, the Latent Dirichlet Allocation, we demonstrate our speculation is correct. In this regards, we design a hybrid strategy for a movie recommender system. Our system consists of two core modules, namely tag recommendation and profile-based item recommendation. The former module leverages user-tags and item-tags to generate recommended tags, whereas the later one enables the use of the mixture of tags and keywords during the recommendation process. It is worth noting that such design requires no prior information of users and hence the cold-start problem prevented.

Author Keywords

Recommender systems, tagging, profile.

ACM Classification Keywords

H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval—Information Filtering; H.5.2 [Information Interfaces and Presentation] Information Interfaces and Presentation—User Interfaces

Workshop SRS'11, March 19-23, 2011, Hangzhou, China. In Conjunction with CSCW 2011

INTRODUCTION

The boom of information from the Internet makes users difficult to gather useful information. Fortunately, the advent of folksonomic systems brings a hope to this situation. A folksonomic system is a complex network of interrelated users, items and tags. It allows users to apply tags, to collectively classify and find information. Users have the greatest freedom of controlling the way they organize tags and thus items. A tag is a short phrase or keyword applied by users and it is more than just a textual description; it can also be used to model user profile and item profile for personalized search results [5]. And therefore the folksonomic system has the ability to explore the large information space, free from a rigid predefined conceptual hierarchy.

Also featured in a folksonomic system is the building of user profile. A profile can be used to store the description of the characteristics of user; it can be represented by tags in a folksonomic system. The system keeps track of what tags have been applied by users, so it can model users by their characteristics and preferences.

Real life examples include Flickr[1], a tag-enabled online album, in which users can upload, share, and annotate photos. Last.FM[2] is a music collection platform, and users are allowed to tag their music, and recommendation service s based on the wisdom of the crowd is available. Delicious[3], is a social bookmarking web service for storing, sharing, and discovering web bookmarks.

However, many of the recommendation algorithms developed for folksonomic systems, such as k-nearest neighbor [8], collaborative filtering [4], etc., rely on prior user information like ratings and tags. Under the lack of user's information, and the sparsity nature of folksonomic system [2], these algorithms do not perform well. This issue is framed as the cold-start problem [7]. In this regards, we are interested in investigating how to utilize other users'

[1] http://www.flickr.com

[2] http://last.fm

[3] http://www.delicious.com

tags to assist a new user during the recommendation process.

More specifically, we speculate that the tags bear more meaning underneath, such as user preferences. This motivates us to probe the underlying meaning of the tags, and use these tags to help new users to construct a user profile, which is supposed to reflect the user's preferences. We then implemented a prototype of movie recommender system accordingly.

This paper is hence organized as follows. We first introduce the dataset for analysis and system implementation in the next section, and then introduce our system flow. The examination of the distribution of preferences and semantic categories across the database follows. We then present our hybrid design with both of tag recommendations and user profile based item recommendations alongside with the movie recommender system.

DATASET

Two sources of data are used throughout this paper. The first dataset is from Movielens, a movie recommender system maintained by GroupLens Research, University of Minnesota. This dataset contains ratings, tags, movies and user IDs. The 10M Ratings and 100k Tags Dataset are used in our study. The second data source is from the IMDB, an online Internet Movie Database. Movie descriptions on IMDB corresponding to the movies in Movielens are crawled for use in this paper.

The datasets were processed as follows. For the Movielens dataset, since we are working on the tag profile, we only considered the users who had at least 30 tags, and the set of movies related to these users. The dataset of IMDB was compared against the standard stop-word list, leaving the non-trivial keywords. As a result, we have 271 users, 6409 distinct tags and 5840 movies in our dataset. Table 1 shows the summary.

	Movielens
No. of users	271
No. of distinct tags	6,409
No. of movies	5,840

Table 1: Summaries of our dataset

SYSTEM FLOW

This paper suggests a hybrid design of recommender system as shown in Figure 1. The hybrid design combines the tag recommendation and the profile-based item recommendation, and it consists of different modules. The input to this system is the querying tag; the system utilizes it together with the tags from the tag recommendation module to generate items recommendation. The details of each module are illustrated in the subsequent sections.

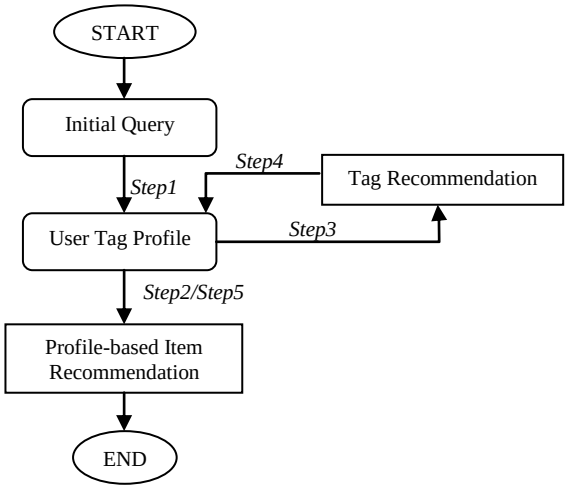

Figure 1: The overall flow of hybrid design

Step 1. *Inputted tag(s) as the user's profile.* To begin the user inputs tag(s) to the system and the tag(s) will be used in the user tag profile.

Step 2. *Profile-based item recommendation.* Based on the tags in the user's profile, the system generates a set of item recommendations (i.e., movies in our prototype) to user.

Step 3. *Tag recommendation.* The tags in the user tag profile are fed into the tag recommendation module.

Step 4. *Incremental profile building.* User can add from the recommended tags to enrich his/her user tag profile.

Step 5. *Returning a new set of item recommendations.* In the next cycle, the refined user tag profile is used to generate a finer-grained item recommendation.

The process from Step 2 to Step 5 continues till the user does not make any changes on his/her profile and selects a most preferred item.

TAG CATEGORIZATION AND RECOMMENDATION

Semantic categories can be identified in the natural language and this helps to understand the meaning of sentences. For example, a common practice for news classification in the data mining field is based on the semantic categorization method. As part of the natural language, tags can be understood in the same way. Furthermore, the better understanding of the tags can provide more sound recommendation that share similar semantic properties. This leads us to attempt recognizing the semantic meaning of tags by categorizing approach.

In a folksonomic system, user is allowed to assign one or more tag(s) to items, and each user owns a list of tags. Analogously, an item also owns a list of tags in the system, which is contributed by different individuals. We consider these tags can be interpreted in two perspectives, namely user-tags and item-tags. We speculate the two perspectives constitute different semantic categories, and we examine this phenomenon as follows.

User-Tags

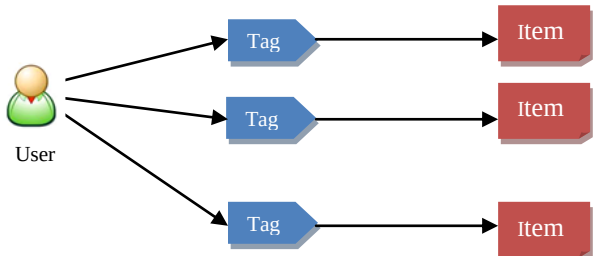

Figure 2: The schematic representation of user-tag

Because each individual in a folksonomic system would use tags to annotate different items based on his own preference, we assume that the user-tags bear the general preferences among users. With the varying preferences, each individual exhibits different tags of choices. Figure 2 shows the schematic representation of user-tags.

To examine the underlying semantic categories in user-tags, we consider each user as a document, and the list of tags assigned by the user as words. Then we deploy the Latent Dirichlet Allocation [3], a dimensionality reduction technique to extract topics in information retrieval [12]. We also use the top three representative tags to represent each topic discovered in LDA. The representativeness of a tag is in turn defined by the *term frequency* within the topic [6].

As expected, the results of LDA in this perspective demonstrated the general preferences among users, because the resulted clusters consist of tags that are appeared in user-tags with high probabilities. For example, the cluster {action, Comedy, Drama} indicates the preference of watching comedic drama movie, while the cluster {70mm, Betamax, DVD-video} indicates the preference of media used.

Item-Tags

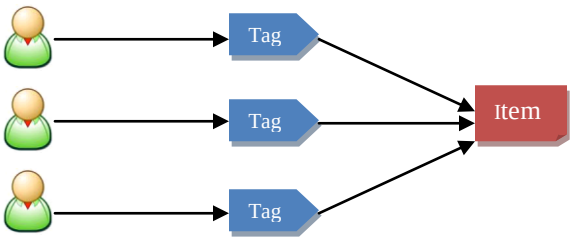

Figure 3: The schematic representation of item-tag

For the item-tags, those frequently occurred in an item are commonly agreed by various parties, and therefore we speculate that the item-tags can deliver the subjective meaning of items. Figure 3 shows the item-tags.

The same procedure to examine user-tags is used to probe the item-tags. The results proved our speculation is correct. For example, the two clusters {Disney, Animation, Pixar} and {anime, comic book, Japan} are referring to Disney animation and Japan's animation respectively.

Tag Recommendation

The discovered semantic categories form the basis of the tag recommendation algorithm. The identified general preferences and subjective descriptions in user-tags and item-tags can be used for generating recommendations. For instance, if a user is interested in tags with horrible meaning, we can recommend to him a set of tags that are semantically similar. More details of the tag recommendation module are discussed later. Figure 4 is a snapshot of the recommended tags.

Figure 4: snapshot of the recommended tags

Tag Classification

Besides the underlying semantic categories, tags in a folksonomic system can also be classified into the categories of *Factual Tags*, *Subjective Tags* and *Personal Tags* as proposed by Khalifa et al. [1]. This classification is initially designed to evaluate the usage of tags and is defined as follows.

- **Personal tags:** "have an intended audience of the tag applier themselves. They are often used to organize a user's own resource (self-reference, task organization, time management) e.g. 'myblog'.
- **Subjective tags:** express people opinions related to a web resource e.g. 'cool',
- **Factual tags:** identify 'facts' about the described web resource such as people, places, or concepts e.g. 'tutorial'."

Instead of coding the tag's category manually as they do, we propose a heuristic method to achieve it. We compare the tags against the movie's descriptions obtained from IMDB. If there is a matching, we consider those tags as the *Factual Tags*. The remaining tags are then checked with the General Inquirer[4], a content analysis program which is capable of determining whether a tag is subjective or not, to

[4] http://www.webuse.umd.edu:9090/

identify if they belong to the *Subjective Tags*. Lastly, the rest of the tags are considered to be the *Personal Tags*.

INCREMENTAL USER PROFILE BUILDING

The input to the tag recommendation module is a set of tags. It has twofold value. It incrementally builds the user profile by leveraging user-tags and item-tags, as well as coping the problem of no prior ratings and tags of users, i.e. the cold-start problem. The two stages of our algorithm are discussed as below.

Profile Enhancement Stage

When user issued a query (which is treated as a tag in the user's profile), we add more tags from others to his query by using WordNet [10], a lexical database for the English language that groups English words into sets of synonyms and provides semantic relations among them. WordNet returns tags that are lexically correlated to the querying tag, and this correlatedness is in turn measured by the metric proposed by Wu and Palmer [11]. Formally, the tags inputted by a user i denoted by $\overrightarrow{T_i}$ is a vector of tags as follows:

$$\overrightarrow{T_i} = (t_{i,1}, t_{i,2}, t_{i,3} \dots t_{i,n}) \quad (1)$$

where $t_{i,x}$ is a tag that user i inputted, n indicates the n^{th} tags in the input. The number of tags returned by WordNet, denoted by $|T_{WN}|$ is determined in the following way:

$$|T_{WN}| = 5 - |\overrightarrow{T_i}| \quad (2)$$

That is, we always keep the user profile to have at least five tags, so as to provide more sound information for the next stage of our algorithm.

Seeking for the Relevant Cluster

As mentioned in the previous section, the clusters formed in the user-tags and item-tags bear different semantic categories, which are the general preferences and subjective meanings respectively. In the second stage of our approach, we aim at seeking for the most relevant cluster to enhance user profile.

The similarities of the clusters of user-tags correspond to each cluster of item-tags are pre-assessed based on the Symmetric Jacaard Coefficient:

$$Sim\left(C_{i^{th}}^{user}, C_{j^{th}}^{item}\right) = \frac{|T_i \cap T_j|}{|T_i \cup T_j|} \quad (3)$$

where $C_{i^{th}}^{user}$ denoted the i^{th} cluster in user-tags, $C_{j^{th}}^{item}$ denoted the j^{th} cluster in item-tags. T_i is the tagset of $C_{i^{th}}^{user}$ and T_j is the tagset of $C_{j^{th}}^{item}$.

To locate an approciate cluster for recommendation, we first determine the most relevant cluster of the user-tags to the user profile. Then, the next step is to choose between the user-tags' cluster and item-tag's cluster (the most similar one to the user-tag's cluster), using the degree of relevance as defined as follows:

$$Relevance = |O| \quad (4)$$

where

$$O = (t_1, t_2, t_3, \dots t_n), \forall x, t_x \in \overrightarrow{T_i} \bigcup C_{i^{th}}^{user} \quad (5)$$

That is, the degree of conjunction is regarded as the relevance of the two tagsets. The winning cluster will be used for recommendation. As a result, we return to the user a set of relevant tags as suggestions.

Incremental Profile Building Property

With the tag recommendation strategy outlined above, user can add the recommended tags to his user profile. For the second or later cycle of interaction, the added tags together with the original tags will be fed to the tag recommendation module and produce a new set of recommended tags. Each time the user refine the user profile, a set of more relevant movies can be generated from the item recommendation module. This process is repeated until the user's profile is complete and accurate enough.

In our system, the user profile is represented by the user's inputted and applied tags. User is allowed to click on the recommended tags based on his preference, and the selected tags will be appended in the list of user profile. In case of choosing an undesired tag, he can roll back to the previous state by removing the unwanted tag in the profile by clicking the delete button. The removal of tag implies that the tag does not fit the user's preference, and this will be logged in the system to avoid recommending the same tag again. Figure 5 shows the snapshot of user profile.

Figure 5: snapshot of the user profile

PROFILE-BASED ITEM RECOMMENDATION

After constructing the user profile, we can recommend items to user. To do so, the mixture of FolkRank and content-based filtering approach is adopted to rank movies as below:

$$\boldsymbol{Score(M)} = \boldsymbol{\alpha}\big(\boldsymbol{FR(M)}\big) + \boldsymbol{(1-\alpha)(CB(M))} \quad (6)$$

where $\boldsymbol{M}$ denoted a movie, $\boldsymbol{FR(M)}$ denoted the score from FolkRank, and $\boldsymbol{CB(M)}$ denoted the score from content-based filtering. $\boldsymbol{\alpha}$ in the range of [0,1].

Figure 6: The interface of our movie recommender system.

FolkRank is a graph-based recommendation algorithm designed for folksonomic systems. It transforms the tripartite graph found in folksonomic systems into the two-dimension hyper-graph, based on the user, tag, and item relationships. The content-based filtering approach selects items based on the correlation between the content of the items and the user's preferences as opposed to a collaborative filtering system that chooses items based on the correlation between persons with similar preferences.

The tuning parameter α is currently set to 0.75. We can adjust it to control the weight of FolkRank and content-based filtering in our algorithm, which is to determine whether we rely on tags more or keywords during the recommendation process. In case that the user input is essentially keyword, without any tag, the system is still capable of locating the relevant movies, and the coverage of the recommendation is increased by considering the keywords. The list of recommended items will be updated according to the current user profile, and hence we can keep returning items that are relevant to the current user preference.

USER INTERFACE

In our prototype system, the recommended items, movies in our case, are arranged in a circular layout. This arrangement might be faster and more reliable to select from than linear layout, according to Fitts' Law [9].

More specifically, the one with the highest ranking score from the profile-based item recommendation is positioned at the top, and then the remaining movies are positioned clockwise by descending score order. For each movie's poster organized in this circular layout, user can click on it to view that movie's details. The movie details include information like director, cast, plot, link to the IMDB page and so forth as shown in Figure 7. There is also a button in the movie details' dialog that lets user to indicate that this movie is his choice.

The tag input box, displayed as the classified recommended tags, and the user profile are positioned in the middle of the interface, as shown in Figure 6. Furthermore, the recommended tags are limited to five in each category, which can prevent users from being overwhelmed by the vast amount of information.

Figure 7: The movie's details dialog. Movie's information like rating, director, cast and plot are available. User can also choose to link to the IMDB page, the trailers pare, or to view the movie's images.

CONCLUSION

In this paper, we illustrated a hybrid design for movie recommender system. This system has two core modules, namely tag recommendation and profile-based item recommendation. More specifically, the module of tag recommendation leverages user-tags and item-tags, i.e. general preferences and subjective descriptions. Whereas the profile-based item recommendation module enables the use of the mixture of tags and keywords during the recommendation process.

The system supports three major tasks as listed below.

- *Tag Recommendation*
  Based on the proposed algorithm, the system returns a bag of suggested tags for magnifying and stimulating user's preference, and presents them in a categorized manner based on our heuristic classification method.
- *Incremental User Profile Building*
  User can refine his querying tags through an iterative process, and hence complete his preferences gradually. Instead of inputting tags from sketch, user can choose from the recommended tags.
- *Profile-based Item Recommendation*
  The system utilizes the user profile to provide movie recommendations. The mixture of FolkRank and content-based filtering approach is used; this leverages tags and keywords in the process to increase the coverage of the recommendation.

Moreover, our system requires no prior information of the user, which is suitable for providing recommendation to new users and those who do not rate. Therefore, the cold-start problem can be prevented.

Future Work

The system described in this paper is part of our on-ongoing project. Various versions of interfaces will be made and we will recruit users to evaluate both the recommendation algorithm and interfaces designs.

REFERENCES

1. Al-Khalifa, H. S. (2007). Towards better understanding of folksonomic patterns. *Proceedings of the eighteenth conference on Hypertext and hypermedia* (pp. 163-166). Manchester, UK: ACM.
2. Bischoff, K. a. (2008). Can all tags be used for search? *Proceeding of the 17th ACM conference on Information and knowledge management* (pp. 193--202). Napa Valley, California, USA: ACM.
3. Blei, D. M. (2003). Latent dirichlet allocation. *J. Mach. Learn. Res.* , 993--1022.
4. Bobadilla, J. a. (2010). A new collaborative filtering metric that improves the behavior of recommender systems. *Know.-Based Syst.* , 520--528.
5. Cai, Y. a. (2010). Personalized search by tag-based user profile and resource profile in collaborative tagging systems. *CIKM '10* (pp. 969-978). Toronto, ON, Canada: ACM.
6. Dolog, F. D. (2010). Extending a hybrid tag-based recommender system with personalization. *SAC'10: Proceedings of the 2010 ACM Symposium on Applied Computing* (pp. 1723-1727). New York: ACM.
7. Lam, X. N. (2008). Addressing cold-start problem in recommendation systems. *Proceedings of the 2nd international conference on Ubiquitous information management and communication* (pp. 208--211). Suwon, Korea: ACM.
8. Lathia, N. a. (2008). kNN CF: a temporal social network. *Proceedings of the 2008 ACM conference on Recommender systems* (pp. 227--234). Lausanne, Switzerland: ACM.
9. MacKenzie, I. S. (1992). Fitts' law as a research and design tool in human-computer interaction. *Hum.-Comput. Interact.* , 91-139.
10. Miller, G. A. (1995). Wordnet: a lexical database for english. *ACM*, (pp. 38(11):39-41).
11. Palmer, Z. W. (1994). Verbs semantics and lexical selection. *In Proceedings of the 32nd annual meeting on Association for Computational Linguistics* (pp. 133-138). Morristown: Association of Computational Linguistics.
12. Wei, X. a. (2006). LDA-based document models for ad-hoc retrieval. *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 178--185). Seattle, Washington, USA: ACM.