Utilising Document Content for Tag Recommendation in Folksonomies

Nikolas Landia
University of Warwick
Coventry CV4 7AL
UK
N.Landia@warwick.ac.uk

ABSTRACT

Real-world tagging datasets have a large proportion of new/ unseen documents. Few approaches for recommending tags to a user for a document address this new item problem, concentrating instead on artificially created post-core datasets where it is guaranteed that the user as well as the document of each test post is known to the system and already has some tags assigned to it. In order to recommend tags for unseen documents, approaches are required which model documents not only based on the tags assigned to it in the past (if any), but also the content.

The focus of my research is on utilising the content of documents in order to address the new item problem in tag recommendation. I apply this methodology first to simple baseline tag recommenders and then the more advanced tag recommendation algorithm FolkRank [3][4].

One of my main contributions is a novel adaptation to the FolkRank graph model to use multiple word nodes instead of a single document node to represent each document. This enables FolkRank to recommend tags for unseen documents and makes it applicable to full real-world tagging datasets, addressing the new item problem in tag recommendation.

Categories and Subject Descriptors

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

General Terms

Algorithms

Keywords

tag recommendation, social bookmarking, FolkRank

1. INTRODUCTION

With the advancement of Web 2.0, tagging is a popular methodology for many user-driven document organisation applications, such as social bookmarking websites. The automatic generation of tag recommendations aid the social tagging process and lead to a more valuable document organisation overall. Tags are an unstructured organization method where each user has the liberty of choosing and/or making up any string of characters to be used as a tag for a document. The task of tag recommendation is to automatically suggest a set of tags to a user for a document that he is in the process of tagging.

The data contained in social tagging systems is often described as a folksonomy. A folksonomy is a tuple (U, D, T, A) where U is the set of users, D is the set of documents, T is the set of tags and $A \subseteq U \times D \times T$ is the set of tag assignments. A tag assignment $a \in A$ is a triplet (u, d, t) and means that user u has assigned tag t to document d [4]. Thus a folksonomy can be modelled as a hyper-graph with the adjacency tensor given by a 3-dimensional binary matrix $F = [f_{i,j,k}]_{|U|\times|D|\times|T|}$ where each entry $f_{i,j,k} \in \{0, 1\}$ specifies whether or not user u_i tagged document d_j with tag t_k. A post in the folksonomy consists of a **set** of tags T_{ij} assigned by a user u_i to a document d_j. The set of posts is given as $P \subseteq U \times D \times 2^T$ where each post $p \in P$ is a triplet (u_i, d_j, T_{ij}). We use the notation $(u_i, d_j, \emptyset)$ for query posts where the set of tags T_{ij} is unknown and to be predicted.

2. RELATED WORK

Existing tag recommendation solutions can be categorised into approaches which model and analyse the folksonomy data in order to come up with recommendations; and content-based approaches where the textual content and/or metadata of documents is considered.

Methodologies relying on the folksonomy data include Hypergraph [12][10], Graph [4][9][5] and Collaborative [1][7][13] approaches. While hypergraph approaches try to capture and analyse all characteristics of the data in their models, graph-based and collaborative approaches can be described as reductionist methods since they reduce the 3-dimensional folksonomy data to one or several 2-dimensional projections. In content-based approaches, the textual content of the documents is used for either Tag/Keyword Extraction [7] or with document classification techniques [2][11].

Hybrid approaches which combine several tag recommenders also exist and have been explored in [8]. The challenge with hybrid approaches is how to combine the recommendations given by the individual components of the system to achieve the best effect.

The effectiveness of tag recommendation algorithms is usually evaluated against simple baseline recommenders. These

RecSys'12, September 9–13, 2012, Dublin, Ireland.
Copyright 2012 ACM 978-1-4503-1270-7/12/09 ...$15.00.

include recommending the most popular tags in the system, the most used tags of the user and the most popular tags of the document.

3. PERSONALISING CONTENT-BASED TAG RECOMMENDATION

My first contribution was a hybrid tag recommender which generates predictions based on the content of documents and personalises the resulting tag list to reflect the preference of the user which is learnt from the user profile [6]. The existing documents in the system are represented by their fulltext content using a bag-of-words representation with each word having an importance score with regard to the document (Tf-Idf score). The known training documents are organised into groups based on their content by a hierarchical clustering technique. This allows the system to assign the query document to a cluster in the hierarchy and thus identify a content-based neighbourhood of existing documents. The initial set of candidate tags then consists of past tags not only related to the query document but also to documents in the neighbourhood. This gives the system the potential to recommend tags for new documents that have not been tagged yet. The hierarchical clustering also provides a set of content words which are deemed most important for each cluster of documents. If the set of candidate tags for a test post does not contain a sufficient number of tags with scores above a preset threshold, these cluster-related words can be used to generate new tags for the test post.

4. EXTENDING FOLKRANK WITH CONTENT DATA

While utilising the immediate neighbourhood of the query document and user is a simple and effective way to generate and rank a small set of candidate tags, there is a limit to how many of the real tags can be found this way with a reasonable accuracy. If the user decides to use a tag that he has not used before and that also has not been used by his peers for the query document (or its immediate neighbourhood), then a more advanced ranking technique is required to filter out the most probable tags from the large number of total tags in the system.

FolkRank is a graph-based tag recommendation algorithm [3][4] which is modelled on Google's PageRank. Similarly to PageRank, the key idea of FolkRank is that a document which is tagged by important users with important tags becomes important itself. The same holds symmetrically for users and tags. Users, documents and tags are represented as nodes in an undirected multi-edge tri-partite graph with uniform edge weights (all edges have weight 1). All co-occurrences of users and documents; users and tags; and documents and tags are edges between the corresponding nodes.

The importance or rank of each node is calculated by an iterative weight-spreading algorithm, in a similar fashion to PageRank. To find a set of tag recommendations for a new post $(u_i, d_j, \emptyset)$, where the set of tags T_{ij} is not known (and is denoted by the empty set), the nodes in the graph representing u_i and d_j are given a high preference weight and the iterative weight-spreading algorithm is executed until the weights in the graph stabilise. The nodes which represent tags are then ranked in descending order of weight, and the top K ranked tags are selected as tag recommendations, where K is a predefined number usually set to a value between 1 and 10.

One drawback of the FolkRank algorithm is its complexity and long runtime. However, more importantly, FolkRank can only generate successful recommendations for posts where the user and document are already known to the system. This makes FolkRank only applicable to artificially created post-core datasets of level 2 or higher with any success. When trying to apply FolkRank to a query post with a new user and/or new document, it defaults to recommending either the most popular tags of the user, the most popular tags of the document or the most popular tags in the system, depending on whether the document, the user or both are new. While the new user problem is not as prominent since each user is only new during his first post, the new document problem is present in the vast majority of posts. Post-cores of level 2 or higher thus only capture a small fraction of the real-world tag recommendation problem. In the Bibsonomy dataset, only about 15.2% of all posts are included in the post-core level 2 subset.

In the full Bibsonomy dataset, the number of posts where the user is new (first encountered) is equal to the number of users; this is the case in roughly 1% of all posts. The number of posts where the document is new (first encountered) is equal to the number of documents; which comprises 90% of all posts. A further consideration is that FolkRank cannot generate and recommend new tags which do not exist in the training data. In Bibsonomy the number of tag assignments where the tag is new (first encountered) is equal to the number of tags and is the case in 7% of all tag assignments.

4.1 Extension of Folkrank with Content Data

In order to overcome the new item problem and make FolkRank applicable to full real-world datasets, we include the content of documents in the tag recommendation process. For test posts where the query user is new (as well), we have to default to the most popular tags found to be related to the query document and cannot personalise these to the user, which is acceptable since the user does not have a tagging profile yet.

4.1.1 ContentFolkRank

Our approach for including content data into FolkRank is to include the word content of documents directly into the FolkRank graph. We adapt the graph to use triplets $(user, word, tag)$ instead of $(user, document, tag)$. Each tag assignment in the training data (u, d, t) is converted to a set of tag assignments with words instead of documents $\{(u, w_1, t), (u, w_2, t), ..., (u, w_k, t)\}$ where each of the words $w_l \in d$. The test vector for each test post (u_q, r_q) is then given by $(u, w_1, w_2, ..., w_k)$ where each $w_q \in d_q$.

We first change the FolkRank graph from having multiple edges with uniform edge weights to a single-edge graph with weighted edges. If we set the weight of the edge between two nodes n_1 and n_2 to the number of nodes N which have an edge to both n_1 and n_2, our graph configuration is equivalent to the original FolkRank multi-edge graph. However, using single weighted edges reduces the runtime of the recommender considerably and allows for easier manipulation of edge weights.

We then create custom rules for setting the weights of edges connecting different types of nodes, namely user - word edges, word - tag edges and user - tag edges. We want the sum of the weights of edges connecting any user u to words nodes from any one document d to be either zero or equal to a pre-defined constant. This would mean that regardless of the number of words that a document is represented by, the sum of weights of edges connecting u to the word nodes representing d will always be the same. To achieve this and additionally include the varying importance of different content words to the document, we use the Tf-Idf scores of the words in the document, where the Tf-Idf scores are normalised to sum 1. Since several documents, for example d_a and d_b, tagged by the same user u_a can contain the same word w_a, the weight of the edge between u_a and w_a is set to the sum of the Tf-Idf scores of w_a in d_a and d_b. The same holds for edges between word and tag nodes. The weight of the edges between user and tag nodes, u_a and t_a, is set to the number of posts (documents) in which u_a has used t_a.

The preference vector for each test post is given by $(u, w_1, w_2, ..., w_k)$ where each $w_q \in d_q$. The preference weight of the user u is set to a predefined constant PW, while the preference weight for each word w_q is set proportional to its Tf-Idf score in r_q, and is given by $pw(w_q) = PW * tfidf(w_q, r_q)$. Since the Tf-Idf weights are normalised to sum 1 per document, the sum of the preference weights of all words $w \in d_q$ is equal to the preference weight attributed to the user u (and is equal to PW).

5. POSTRANK

An issue which exists in plain FolkRank (as well as our adaptation ContentFolkRank) is the problem of balancing edge weights. Due to the fact that a post can have a variable number of tags assigned to it, the user and document nodes of the post can be connected to a variable number of tag nodes each. The difficulty is then deciding the weight to give to the single edge between the user and document node. One can either keep the weight at 1 regardless of the number of tags in the post or set the user-document weight relative to the number of tags in the post. Both are incomplete solutions only addressing one part of the problem. The problem is to choose edge-weights so that from the perspective of any one node in the graph, an equal total edge weight goes to different types of nodes (ie all document nodes and all tag nodes) while preserving an equal (or explicit) distribution of edge weights between nodes of the same type (ie each document node should get the same weight from a user regardless of the number of tags in their respective posts). In order to address the edge-weight balancing problem, I am currently working on a solution which includes an additional type of node, representing posts themselves, in the graph. This allows to balance edge weights in the context of each post without affecting the weights in other posts which contain the same user, document or tag.

6. EXPERIMENTAL EVALUATION

6.1 Bibsonomy Dataset

The dataset consists of tagging data from the social bookmarking website Bibsonomy[1] [3]. The system and data is divided into web site bookmark tagging data and publication BibTex tagging data. We concentrate on website bookmarks and evaluate our recommenders on this subset of Bibsonomy.

6.2 Evaluation Methodology

6.2.1 Train-Test Split

We split each dataset by date to produce a training and test set, similarly to [7], where the last 3 months of tagging activity is the test set. Table 1 shows the percentage of new users, documents and tags in our test set for the Bibsonomy Bookmark Full dataset.

	% New Items	% Posts with New Items
Users	68%	17.5%
Documents	92.5%	91.6%
Tags	39%	17% *

* This is the average percentage of new tags per test posts, since one post can have multiple tags

Table 1: Bibsonomy Bookmark Test Set New Items

6.2.2 Evaluation Measures

We use Recall@N to evaluate the success rate of the recommenders where N is the predefined number of tags to recommend. Recall is calculated per test post and then averaged over posts to give the overall recall on the test set.

6.3 Results

Figure 6.3 shows the recall on the Bibsonomy Bookmark Full dataset when recommending tag sets of different sizes. In these initial results, the source of content words which are used in the ContentFolkRank recommender is the title of the documents. As expected, FolkRank performs better than the baseline recommenders at all levels since it is able to aggregate much wider-reaching connections in the data than the immediate neighbourhoods from which the baseline recommendations are constructed. By including content in the FolkRank graph, the ContentFolkRank recommender shows a considerable improvement over plain FolkRank.

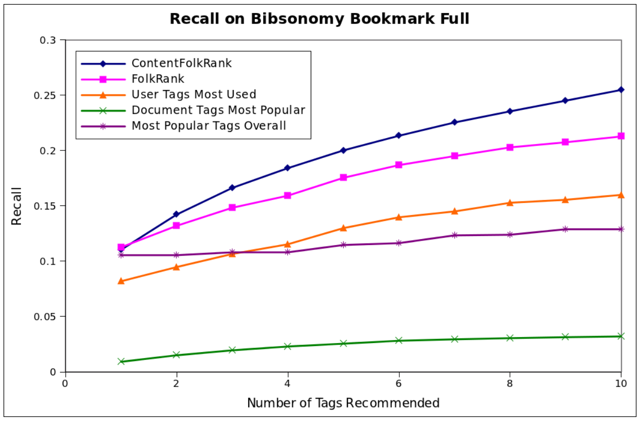

Figure 1: Recall on Bibsonomy Bookmark Full

7. CONCLUSION

In order to utilise tag recommendation algorithms in real-world tagging systems, it is important that the algorithms

[1] http://www.bibsonomy.org/

can recommend tags for previously untagged documents since the vast majority of posts in these systems contains new documents. The new item problem for documents can be addressed by building recommenders which consider the content of the documents in their models. My research is focused on this task and I have applied this methodology first to baseline recommenders [6] and then to the more potent graph-based recommender FolkRank. The inclusion of document content data in tag recommendation systems shows an improvement over purely folksonomy-based recommenders and makes the content-aware recommenders viable for use in real-world tag recommendation systems.

8. FUTURE WORK

In the future, I plan to fully evaluate the suggested ContentFolkRank recommender on further datasets such as CiteULike[2] and Delicious[3] as well as complete the development of the PostRank recommender. I also plan to examine the usefulness of topic models with regard to FolkRank. After learning a set of topics from the content of the training documents, the FolkRank graph could be created with nodes which represent topics instead of documents or words. A query document would then be mapped onto a set of topics and the document-related part of the preference vector would be a set of weights which represent the membership of the document to each of the topics. With ContentFolkRank, especially when using content information from metadata and fulltext content, the need arises to reduce the dimensionality of the document representation and size of the graph. While topic models would be one way to achieve this, I also plan to explore traditional dimensionality reduction alternatives.

9. REFERENCES

[1] J. Gemmell, T. Schimoler, M. Ramezani, and B. Mobasher. Adapting k-nearest neighbor for tag recommendation in folksonomies. In *Proceedngs of the 7th Workshop on Intelligent Techniques for Web Personalization and Recommender Systems*, 2009.

[2] P. Heymann, D. Ramage, and H. Garcia-Molina. Social tag prediction. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 531–538, New York, NY, USA, 2008. ACM.

[3] A. Hotho, R. Jäschke, C. Schmitz, and G. Stumme. Information retrieval in folksonomies: Search and ranking. In Y. Sure and J. Domingue, editors, *The Semantic Web: Research and Applications*, volume 4011 of *Lecture Notes in Computer Science*, pages 411–426. Springer, 2006.

[4] R. Jaeschke, L. B. Marinho, A. Hotho, L. Schmidt-Thieme, and G. Stumme. Tag recommendations in folksonomies. In *Knowledge Discovery in Databases: PKDD 2007, 11th European Conference on Principles and Practice of Knowledge Discovery in Databases*, 2007.

[5] H.-N. Kim and A. El Saddik. Personalized pagerank vectors for tag recommendations: inside folkrank. In *Proceedings of the fifth ACM conference on Recommender systems*, RecSys '11, pages 45–52, New York, NY, USA, 2011. ACM.

[6] N. Landia and S. Anand. Personalised tag recommendation. In *Proceeding of the Recommender Systems and the Social Web Workshop, held in conjunction with the ACM conference on Recommender Systems*, 2009.

[7] M. Lipczak, Y. Hu, Y. Kollet, and E. Milios. Tag sources for recommendation in collaborative tagging systems. In *Proceedings of the ECML/PKDD 2009 Discovery Challenge Workshop, part of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, 2009.

[8] M. Lipczak and E. Milios. Learning in efficient tag recommendation. In *RecSys '10: Proc. the 4th ACM Conference on Recommender Systems*, pages 167–174. ACM, 2010.

[9] M. Ramezani, J. Gemmell, T. Schimoler, and B. Mobasher. Improving link analysis for tag recommendation in folksonomies. In *Proceedings of the 2nd Recommender Systems and the Social Web held in conjunction with the 4th ACM conference on Recommender systems.*, 2010.

[10] S. Rendle, L. Balby Marinho, A. Nanopoulos, and L. Schmidt-Thieme. Learning optimal ranking with tensor factorization for tag recommendation. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '09, pages 727–736, New York, NY, USA, 2009. ACM.

[11] Y. Song, Z. Zhuang, H. Li, Q. Zhao, J. Li, W.-C. Lee, and C. L. Giles. Real-time automatic tag recommendation. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 515–522, 2008.

[12] 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.

[13] Z. Xu, Y. Fu, J. Mao, and D. Su. Towards the semantic web: Collaborative tag suggestions. In *Proceedings of the Collaborative Web Tagging Workshop at the WWW 2006*, Edinburgh, Scotland, 2006.

[2] http://www.citeulike.org/

[3] http://delicious.com/