Add-A-Tag: Learning Adaptive User Profiles from Bookmark Collections

Elke Michlmayr[◇] michlmayr@wit.tuwien.ac.at
Steve Cayzer[‡] steve.cayzer@hp.com
Paul Shabajee[‡] paul.shabajee@hp.com

[◇]Women's Postgraduate College for Internet Technologies (WIT)[*], Vienna University of Technology, Vienna, Austria
[‡]HP Labs, Filton Road, Stoke Gifford, Bristol BS34 8QZ, United Kingdom

Abstract

In this paper we construct adaptive user profiles from tagging data. We present and evaluate an algorithm for creating such profiles, characterizing its behavior through statistical analysis.

Keywords

tagging user profiles adaptivity

1. Introduction

Collaborative tagging systems, also called folksonomies or social bookmarking services, allow their users to manage bookmarks online and to annotate them with free-text keywords (tags) for improving re-discovery of information. The most well-known service, del.icio.us [4], was started in 2003 and many other services have followed since. Since many users of a folksonomy stick with the same bookmark collection for years, these data contain very fine-grained information about a user's changing interests over time.

This paper describes how to utilize data from a social bookmarking service to create user profiles that can in turn be used for Information Filtering (IF). Unlike many other profile learning mechanisms, which usually rely on relevance feedback from the user, our method does not require any additional user input. Moreover, since tagging data is time-based, it allows us to create user profiles that dynamically adapt to drifts in users' interests. The profile should represent the most important parts of a users' behavior (that is, some compression, clustering or summarization needs to be performed). Both persistent long-term interests and transient short-term interests should co-exist in the profile.

2. The Add-A-Tag algorithm

We focus not only on which tags have been used, but rather on which tags have been used in combination. This can be achieved by relying on the co-occurrence technique. If two tags are used in combination (*co-occur*) by a certain user for annotating a certain bookmark, there is some kind of semantic relationship between them. The more often two tags are used in combination, the more intense this relationship is. This is represented by a graph with labeled nodes and undirected weighted edges in which nodes correspond to tags and edges correspond to the relationship between tags. Each time a new tag is used, a new node for this tag is added to the graph. Each time a new combination of tags is used, a new edge with weight α between the corresponding nodes is created in the graph. If two tags co-occur again, the weight for the corresponding edge is increased by β.

To include the age of the bookmarks in the user profile we extend the co-occurrence approach with the evaporation technique known from ant algorithms [2]. Evaporation is a simple method to add time-based information to the weights of edges in a graph: Each time the profile graph is updated with tags from a newly added bookmark, the weights of each edge in the graph is decreased slightly by removing a small percentage of its current value.

Consider a user u adding a bookmark item b tagged with tags $t_1, \ldots, t_n$ to his or her bookmark collection. The profile graph $G_u = (V, E)$ where $V = v_1, \ldots, v_n$ is the set of vertices (which correspond to tags) and $E = e_1, \ldots, e_n$ is the set of edges, is updated as follows. Firstly, the existing information in the graph is changed by applying the evaporation formula shown in Equation 1 to every edge $e_x \in E$

$$w_{e_x} \leftarrow w_{e_x} - \rho \cdot w_{e_x} \ , \tag{1}$$

where $\rho \in [0, 1]$ is a constant and w_{e_x} is the weight of edge e_x.

Secondly, the n new tags from bookmark b: $t_1, \ldots, t_n$ are added to the graph. For every combination $t_i t_j$ where $i, j \in 1, \ldots, n$ and $i < j$, the following procedure is executed:

1. For every tag t_x ($x \in i, j$), add a corresponding vertex v_x to graph G_u, if v_x does not exist.
2. If it does not yet exist, add an edge with weight α between vertex v_i and vertex v_j to graph G_u, where constant α is a real number and $\alpha > 0$.
3. Otherwise, if an edge between vertex v_i and vertex v_j exists, increase its weight by β. Constant β is a real number and $\beta > 0$.

This is executed each time the user adds an bookmark item to the bookmark collection. It creates a profile graph. Extracting the user profile from the profile graph is defined as follows.

1. Create a ordered set E_s from $E = e_1, \ldots, e_n$. E_s contains all edges e_x ($x \in 1, \ldots, n$) from graph G_u ordered in decreasing order by their weights w_{e_x}.

[*]This research has partly been funded by the Austrian Federal Ministry for Education, Science, and Culture (bm:bwk), and the European Social Fund (ESF) under grant 31.963/46-VII/9/2002.

2. Create set E_k by extracting the top k elements from set E_s, where k is a natural number and $k > 0$.

3. Create graph $G_{u'}$ which contains all edges from E_k and all vertices v_x from graph G_u which are incident to one of the edges in E_k.

The size of the user profile $G_{u'}$ is determined by the value chosen for parameter k.

3. Evaluation of profile adaptivity

The amount of change in the user profile depends on the profile creation mechanism, but also on the user's activity pattern. We assess it by comparing the Add-A-Tag algorithm (1) with and (2) without using the evaporation feature. The user's activity pattern serves as a reference value.

We need a way for determining the change of a profile over time. If we compute the user profile of user u at time t_1 and again at time t_2, we need to be able to measure the difference (distance) between these two user profiles. Since measuring graph distances is a only partly solved issue [1], we map the graphs onto a simpler structure which only contains the information we need for the comparison. This structure is a set of edges in decreasing weight order. We define the metric $dist(S_1, S_2)$ for the distance between two sets S_1 and S_2 based on the Kendall τ coefficient [3] – a standard measure for comparing ordered sets that includes rank correlation – as shown in Equations 2a to 2c. The result values for $dist(S_1, S_2)$ are in the range of 0 (if S_1 and S_2 are the same, that is, equally ranked) to 1 (if S_1 and S_2 are in reverse order).

$$dist(S_1, S_2) = 1 - \frac{2 * \tau(S_1, S_2)}{n * (n-1)} \text{ , where} \tag{2a}$$

$$\tau(S_1, S_2) = \sum_{i,j \in P} \bar{\tau}_{i,j}(S_1, S_2) \text{ , and} \tag{2b}$$

$$\bar{\tau}_{i,j}(S_1, S_2) = \begin{cases} 0 & \text{if } i \text{ and } j \text{ are in same order in } S1 \text{ and } S2 \\ 1 & \text{otherwise} \end{cases} \tag{2c}$$

In Equation 2a, variable n is the size of the sets. In Equation 2b, P is the set of pairs of distinct elements in $S1$ and $S2$. The Kendall τ is applicable only for sets which have the same members and – consequently – are of same size. For our setting, this means that those set members that are present in only one of the sets need to be added to the other one. We append the missing set members to the end of the set in order not to affect the ranking of the pairs.

Now we compute the user profiles for a sample user's bookmark collection. We incrementally create the profile graph by adding the bookmark items in their temporal order, and – each time after adding all bookmarks that were created by the user within the time span of one week – we extract the user profile from the profile graph. Using this procedure we retrieve a set of user profiles

$$\overline{G_{u'}} = \{G_{u'}^{w_x} | x = \{1, \ldots, n\}\}$$

for user u and each week w_x. In the next step, we apply the metric *dist* to these data in order to assess the amount of change between the weekly snapshots of the profiles. The user profile $G_{up}^{w_{x+1}}$ for week w_{x+1} is compared to the user profile $G_{up}^{w_x}$ for the previous week w_x.

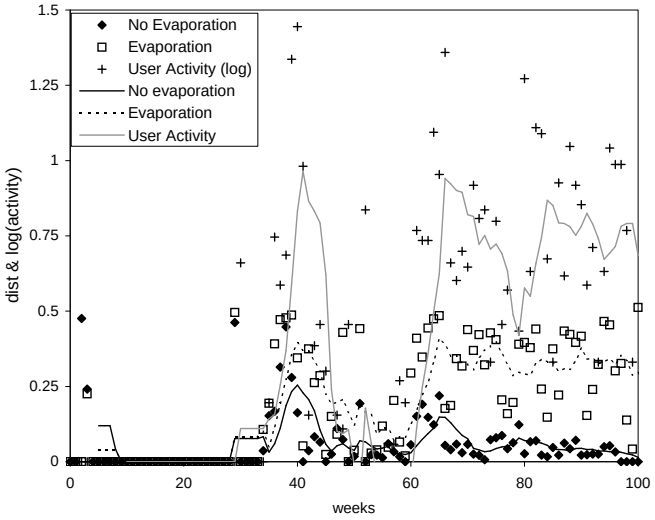

Fig. 1: *Adaptivity comparison for the co-occurrence ($\alpha = 1.0$, $\beta = 1.0$, $\rho = 0$, $k = 20$) and the Add-A-Tag approach ($\alpha = 1.0$, $\beta = 1.0$, $\rho = 0.01$, $k = 20$)*

The result is shown in Figure 1. What the trend lines reveal is that both approaches exhibit a change pattern that is proportional to the user's activity pattern, but the Add-A-Tag approach with evaporation (dashed curve) shows a considerably higher amount of change and fits better with the activity pattern (solid grey curve). If the evaporation feature is not used (solid black curve), the degree of change in the user profile decreases over time. This is particularly the case for the results in the time span between week 80 and week 100. Although the trend line for user activity shows that the user is adding new bookmarks to the collection, the most often used tag combinations are dominant and prevent newly arising tag combinations from being included in the profile.

4. Conclusion

In this paper we proposed the Add-A-Tag algorithm for learning adaptive user profiles from bookmark collections, which is based on a combination of (1) the co-occurrence technique for determining the relationships between tags and (2) an evaporation feature as known from ant algorithms for adapting the user profile to trends over time. We evaluated the algorithm by defining a metric appropriate for quantifying the amount of change over time. What we can show is that the user profiles created with Add-A-Tag are adaptive in the sense that they change according to changes in tag usage in a continuous stream of tagging data.

References

[1] F. Buckley and F. Harary. *Distance in graphs*. Addison-Wesley, 1990.

[2] M. Dorigo and G. D. Caro. *New Ideas in Optimization*, chapter The Ant Colony Optimization Meta-Heuristic, pages 11–32. McGraw-Hill, 1999.

[3] R. Fagin, R. Kumar, and D. Sivakumar. Comparing Top k Lists. *SIAM Journal on Discrete Mathematics*, 17(1):134–160, October 2003.

[4] Yahoo! Inc. Del.icio.us Social Bookmarking Service. `http://del.icio.us`.