Clustering Tags in Enterprise and Web Folksonomies

Edwin Simpson
HP Laboratories, Bristol

HPL-2008-18
February 26, 2008*

tagging, folksonomy, clustering, similarity, navigation

Tags lack organizational structure limiting their utility for navigation. We present two clustering algorithms that improve this by organizing tags automatically. We apply the algorithms to two very different datasets, visualize the results and propose future improvements.

To be presented at International Conference on Weblogs & Social Media, Seattle, March 31st, 2008

Clustering Tags in Enterprise and Web Folksonomies

Edwin Simpson
edwin.simpson@hp.com
HP Labs, Filton Road, Stoke Gifford, Bristol, UK

Abstract

Tags lack organizational structure limiting their utility for navigation. We present two clustering algorithms that improve this by organizing tags automatically. We apply the algorithms to two very different datasets, visualize the results and propose future improvements.

Keywords

tagging, folksonomy, clustering, similarity, navigation

Introduction and Motivation

Tags are simple, ad-hoc labels assigned by users to describe or annotate any kind of resource for future retrieval. Their flexibility means they are easy to add and they capture a user's perspective of resources. The tags added by a group of users form a FOLKSONOMY. Unfortunately, folksonomies are difficult to navigate if tags are presented as long lists. Also, different users refer to the same concepts using different tags (Golder & Huberman 2006) and often add tags of little use to others, e.g. "toRead" .

It has been proposed that folksonomies contain nested groups of tags related to common topics (Heymann & Garcia-Molina 2006; Damme, Hepp, & Siorpaes 2007). This paper describes two algorithms that extract such topic groupings from TAG CO-OCCURRENCE data. Co-occurrences between tags occur when both tags are used with the same resource. The algorithms should produce evenly-sized and intuitive clusters for browsing.

We collected the first dataset from the social bookmarking service Delicious (http://del.icio.us), and the second from an internal bookmarking service, Labbies, used by a group of researchers at HPLabs. We obtained a subset of Delicious data by selecting all tags from users who have used the tag "dspace" during a 16 week period. The use of a common tag ensures there are some relationships between the tags in the dataset. Statistics of the datasets are given in Table 1.

Dataset	Labbies	Delicious
Number of Users	20	136
Number of Bookmarks	1935	95155
Number of Tags	2092	8012
No. Tag Co-occurrences	9526	61453

Table 1: Dataset Statistics

Tag Similarity Graphs

We can extract clusters from a similarity graph, where nodes represent tags and edge weights represent strength of similarity based on the number of tag co-occurrences. Extremely popular tags have high co-occurrence values with many other weakly related tags. To avoid this bias when calculating similarity, we calculate the NORMALIZED CO-OCCURRENCE, or NCO by normalizing the raw co-occurrence value relative to the popularity of two tags using the Jaccard index (Begelman, Keller, & Smadja 2006).

$$NCO = \frac{|A \cap B|}{|A \cup B|} \tag{1}$$

Here A is the set of documents tagged with tag a, and B the set of documents tagged with b. Other normalizations are also possible, such as cosine similarity.

Clustering Algorithms

The algorithms we tested are hierarchical divisive clustering algorithms. Algorithm (1) is as follows, starting with a graph containing all tag relationships, G:

1. Count the number of clusters present in G by counting the disconnected sub-graphs.
2. Evaluate the current clustering using MODULARITY (Newman & Girvan 2004), a quality measure defined as:

   $$modularity = Tr\mathbf{e}\ - ||\mathbf{e}^2||, \tag{2}$$

   where $Tr\mathbf{e}$ is the fraction of edges that connect nodes in the same cluster, and $||\mathbf{e}^2||$ is the fraction of edges that would connect nodes in the same cluster if the clusters were marked randomly in the graph. If $modularity >$ all previous modularities, set $G_{maxMod} = G$.
3. Remove the edge with the lowest NCO value from G.
4. Repeat process from step 1 until no edges are left. Heuristics may be used here to reduce the number of iterations, e.g. stop when a modularity exceeds a certain threshold.

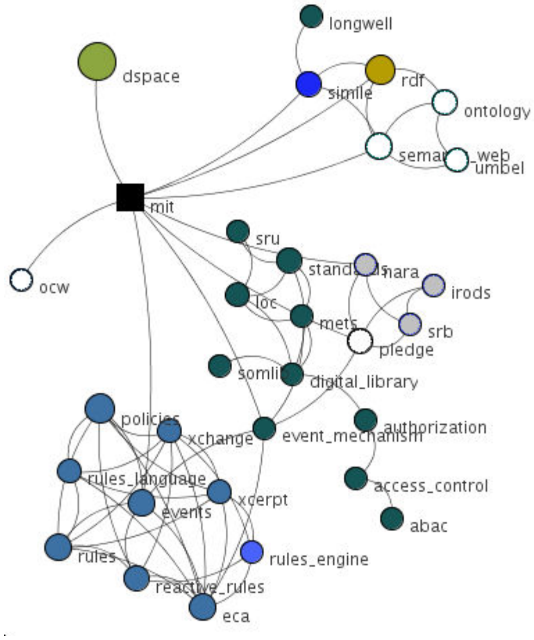

Figure 1: Clusters in Labbies (NCO in step 3.) root=”mit”; cluster distance<2; min. tag use>2; NCO>0.06

5. Determine the tag clusters from the disconnected subgraphs in G_{maxMod}.

Algorithm (2) differs only in step 3. Instead of removing edges with low NCO we remove edges with high BETWEENESS (Brandes 2001). Edges that lie on many shortest paths between nodes in the graph have high betweenness. These edges lie in sparse parts of the graph connecting two clusters, so high betweenness indicates the boundary between clusters.

Visualization

We used a Spring Layout (Eades 1984) to position nodes to visualize tag co-occurrence graphs generated from the datasets. This places nodes closer together if they have strong connecting edges. Nodes are colored to indicate different clusters. Due to the large amounts of data, we remove edges where NCO value is below a threshold, and remove tags with a low number of uses. We then select a sub-graph to visualize by specifying a root cluster, and all tags within a given distance, where distance is the number of edges between a given tag and the root cluster.

Results and Evaluation

Using visualizations we observed that both algorithms produced clusters reflecting the topics in Labbies. Figure 1 demonstrates this with clusters related to the topics ‘policies”, “digital_library” and “semantic_web”. The algorithm has recognised the close relationship between the tags “reactive_rules”, “policies” and “eca”, demonstrating that co-occurrence data can be used to identify topics. Popular nodes connected to many clusters, such as “mit”, were separated into their own clusters. This is a sensible outcome as these are general or ambiguous terms, which do not belong in just one of specific cluster.

Algorithm (2) produced only 133 “clusters” with just one tag, compared to 338 using algorithm (1). The largest cluster was smaller with 126 rather than 191 tags. Since both singleton clusters and extremely large clusters do not divide the folksonomy into useful topics, algorithm (2) showed better performance with this dataset.

The clustering of Delicious data is also reasonable, but the largest cluster for algorithm (1) has 716 tags, which is likely to be difficult for users to browse. Algorithm (2) performed worse in this respect, with 4863 tags in the largest cluster. Tags such as “Spain”, “university” and “relationships” are placed in the largest cluster, despite not being semantically close, suggesting the algorithms were unable to divide the cluster sufficiently.. A possible improvement is to modify the algorithms to prefer removing edges from the largest clusters.

Another cause may be noisy data, so we tested removing tags with less than 25 occurrences. With algorithm (1) this reduces the largest cluster size to 87, and increase the proportion of clusters with 4 to 40 tags, suggesting that removing low occurrence tags may help create better clusterings for browsing. However, with algorithm (2), the largest cluster still has 1115 of 1594 tags, suggesting that it would not produce a good clustering for this type of data.

Conclusions

We tested two hierarchical clustering algorithms on two datasets and visualized the results using a Spring layout. We showed that some clusters produced may be too large for human navigation, and that removing unpopular tags before clustering can reduce this. Algorithm (2) used betweenness to select edges to remove, which was effective for Labbies, but performed poorly on densely inter-related tags in the Delicious dataset.

References

Begelman, G.; Keller, P.; and Smadja, F. 2006. Automated tag clustering: Improving search and exploration in the tag space.

Brandes, U. 2001. A faster algorithm for betweenness centrality.

Damme, C. V.; Hepp, M.; and Siorpaes, K. 2007. Folksontology: An integrated approach for turning folksonomies into ontologies. *European Semantic Web Conference*.

Eades, P. A. 1984. A heuristic for graph drawing. In *Congressus Numerantium*, volume 42, 149–160.

Golder, S., and Huberman, B. 2006. Usage patterns of collaborative tagging systems. *Journal of Information Science* 32(2):198–208.

Heymann, P., and Garcia-Molina, H. 2006. Collaborative creation of communal hierarchical taxonomies in social tagging systems. *InfoLab Technical Report*.

Newman, M. E. J., and Girvan, M. 2004. Finding and evaluating community structure in networks. *Physical Review E* 69:026113.