Integrating Specialist and Folk Knowledge with Affinity Propagation

Jeon Hyung Kang  
USC Information Sciences Institute  
4676 Admiralty Way  
Marina del Rey, CA 90292  
jeonhyuk@usc.edu

Kristina Lerman  
USC Information Sciences Institute  
4676 Admiralty Way  
Marina del Rey, CA 90292  
lerman@isi.edu

Abstract

Web users form a highly diverse group with varying interests, levels of expertise, enthusiasm, and expressiveness. This diversity is also manifested in the annotations users create to organize and label content on social Web sites. While several approaches have been proposed to mine social annotations, e.g., to create folksonomies, or taxonomies that reflect how a community organizes knowledge, these methods treat all users uniformly. We propose a framework to identify specialists, i.e., knowledgeable users who also create high quality annotations, and use their knowledge to guide the folksonomy learning problem. We illustrate the approach on social annotations extracted from the photosharing site Flickr and show that integrating specialist knowledge leads to more consistent and relevant folksonomies. Moreover, we show that including annotations from non-specialists leads to more comprehensive folksonomies than using specialist knowledge alone.

1 Introduction

Knowledge on the social Web grows each time a user annotates a resource, for example, a Web page, a scientific article, a photo, or a video. While attaching descriptive labels, known as tags, to resources is still the most popular form of annotation, some social Web sites also allow users to create structured annotations. For example, social bookmarking sites Delicious (http://del.icio.us) and Bibsonomy (http://bibsonomy.org) allow users to specify broader–narrower relations between tags, and the social photosharing site Flickr (http://flickr.com) allows users to organize photos within folder-like hierarchies. While such annotations reflect individual users’ needs and requirements for organizing the content they create, collectively social annotations provide valuable evidence for harvesting social knowledge. Folksonomies, or taxonomies of concepts automatically extracted from social annotations of many users, will eventually help us better search for, browse, organize, manage, and integrate information on the Web.

Recently, Ref. [1] proposed a method to learn more complete, complex folksonomies from large collections of small personal hierarchies created by individual Flickr users. Their method extends affinity propagation [3] to use structural information to guide the inference process to combine small personal hierarchies concurrently into a tree-like structure. The method assumed that the quality of annotation from every user was the same. However, Social Web users are a diverse group, with varying levels of expertise and expressiveness. Some users, who we call specialists, are so passionate about some topic that they are driven both to acquire more knowledge about it and to express this knowledge through detailed, comprehensive annotations. We find that the quality of annotations coming from such users is higher than others. We claim that an inference method that exploits user diversity by putting greater weight on annotations created by specialists can learn better folksonomies than the method that ignores user diversity.
In this paper we propose a framework for identifying specialists and biasing folksonomy learning process to put greater emphasis on their annotations. In Section 2 we describe how we identify specialists based on the structure of annotations and their semantic consistency. In Sec. 3 we describe the learning procedure and how it can be biased to include specialist data. We validate the approach by showing that specialist knowledge can improve folksonomies learned from user annotations.

2 Identifying Specialists on Flickr

Flickr allows users to upload photos, share them with other users, and categorize them by tagging them with descriptive labels known as tags. In addition, users can organize photos hierarchically by grouping related photos in *sets* and related sets in *collections*. For example, one user may organize her travel photos according to a place hierarchy, with the top-level collection called ‘journey’, which may contain other collections , with cities and landmarks at lower levels or organize plant photos according to type and botanical names. As is the case for tags, Flickr does not enforce any rules on how people name or use sets and collections.

Following [1], we call a personal hierarchy a user creates a *sapling*. While most users create shallow and narrow saplings consisting of a top-level collection and few constituent sets, some users create very detailed saplings about a particular topic of interest. We call such users specialists. By inspecting saplings created by different Flickr users, we found that structure and semantic consistency are two of the most important factors distinguishing specialists from other users. Specifically, we have identified the following features of a specialist:

- organizes concepts within multi-level hierarchies.
- creates a broad sapling expressing detailed knowledge (e.g., contains 20 bird species in a ‘bird’ sapling).
- generally creates many saplings.
- does not jump many levels, (e.g., attach ‘los angeles’ to ‘world’) nor mix concepts of different generality at the same level (e.g., ‘china’ and ‘disneyland’ are never siblings).
- specifies categories that are meaningful to others (provides generalizable knowledge).
- does not create conflicts (e.g., attach ‘los angeles’ to ‘journey’ in one sapling while attaching ‘journey’ to ‘los angeles’ in another).
- does not create multiple child concepts with same name (e.g., five ‘los angeles’ sets under ‘journey’).
We asked three annotators to review saplings created by 200 randomly selected Flickr users and identify specialists using the factors listed above. Each user’s saplings were laid out as trees using yEd graph visualization tool. Figure 1 shows some of the different kinds of saplings created by users identified as specialists. Deep, bushy saplings, such as the ‘journey’ sapling shown in Fig. 1(a), were commonly associated with specialists. However, a user could also create broad but shallow saplings, as in Fig. 1(b), and still be classified as a specialist as long as these saplings were related to the same higher-level concept, since the saplings contain consistent and generalizable knowledge. Each annotator identified 20–45 out of 200 users as specialists. We use Fleiss’ kappa [7] to measure inter-rater agreement, which was calculated to be $\kappa = 0.6$. All annotators agreed on 18 specialists.

3 A Framework for Integrating Specialist Knowledge

Relational affinity propagation (RAP) was proposed [1] as a method to aggregate structured data in a common, more complex structure. RAP extends the powerful clustering algorithm called affinity propagation (AP) [3], which offers a natural framework to incorporate structural information contained in personal hierarchies. AP identifies a set of exemplars that represent all data items. Exemplars emerge as messages are passed between data items, with each item assigned to an exemplar. AP avoids assigning exemplars which violate these constraints.

In addition to constraints, a similarity function $S(\cdot)$ is used to measure how similar a node is to its exemplar. If $c_{ij} = 1$, then we add $S(c_{ij})$ to our objective function as a similarity between nodes $i$ and $j$; otherwise, $S(c_{ij}) = 0$. The self-similarity, $S(c_{jj})$, also called preference, is usually set to be less than the maximum similarity value in order to avoid creating a configuration with $N$ exemplars.

In general, the higher the value of preference for a particular item, the more likely it is to become an exemplar. Setting all preferences to the same value indicates that all items are equally likely to become exemplars.

AP defines a global objective function, which measures how good the present configuration (exemplars and items assigned to them) is:

$$S(c_{11}, \ldots, c_{NN}) = \sum_{i,j} S_{ij}(c_{ij}) + \sum_i I_i(c_{i1}, \ldots, c_{iN}) + \sum_j E_j(c_{1j}, \ldots, c_{jN}).$$

A message passing algorithm is used to find a configuration that maximizes the net similarity $S$, while not violating $I$ and $E$ constraints.

3.1 Relational Affinity Propagation

Ref. [1, 2] proposed a relational affinity propagation (RAP) framework to cluster structured data. RAP adds a new constraint, the $F$-constraint, which allows a node to select another node as an exemplar only if their parents belong to the same exemplar (i.e., cluster). This “single parent” constraint ensures that the learned structure forms a tree.

The $F$-constraint can be written as:

$$F_j(c_{1j}, \ldots, c_{Nj}) = \begin{cases} -\infty & \text{if node i is a child node,} \\ 0 & \exists i : c_{ij} = 1, explr(pa(i)) \neq explr(pa(ne(j))), \\ & \text{otherwise} \end{cases}$$

where $ne(\cdot)$ returns a set of nodes that share the exemplar of its argument, $pa(\cdot)$ returns index of the parent of its argument, and $explr(\cdot)$ returns the index of the argument’s exemplar. The objective function in Eq. 1 is modified by the addition of a new term $\sum_j F_j(c_{1j}, \ldots, c_{jN})$. 
Consider clustering structured data shown in Fig. 2, where child nodes $i$ and $k$ are deciding whether to merge with node $j$. Exemplars are colored orange, and shaded boxes are drawn around nodes that have been assigned to the same exemplar (i.e., clustered together). Figure 2(a) shows an undesirable situation in which nodes $i$ and $k$ are assigned to exemplar $j$, but their parents belong to different exemplars, $h$ and $n$ respectively. The $F$-constraint has node $i$ check whether the exemplar of its parent (parent exemplar) is the same as the parent exemplar of any of $j$’s neighbors. Suppose that $k$ is found similar enough to $j$ so that they can be merged, then $i$ won’t be able to pick $j$ as an exemplar, preventing the configuration shown in Fig. 2(a). The original formulation of the $F$-constraint proposed in Ref. [2] was imposed on child nodes only, and forced them to merge in the configuration shown in Fig. 2(b), losing connection between structures above node $j$ and below node $j$. To improve the structure learned by RAP, we slightly modify the constraint, imposing it on parent and child nodes if the parent node is an exemplar. Using the modified constraint, nodes $i$ and $k$ are no longer both assigned to the exemplar $j$, but depending on the similarity values, one of them will pick $j$ as exemplar.

Binary RAP may be written as a factor graph [1] shown in Fig. 2(c). The message update formulas for $\beta$, $\eta$, $\alpha$, $\rho$, $\tau$ and $\sigma$ are:

$$\beta_{ij} = s(i,j) + \alpha_{ij} + \tau_{ij},$$

$$\eta_{ij} = -\max_{k \neq j} \beta_{ik},$$

$$\alpha_{ij} = \begin{cases} \sum_{k \neq j} \max [\rho_{kj}, 0] & i = j \\ \min [0, \rho_{ij} + \sum_{k \notin \{i,j\}} \max [\rho_{kj}, 0]] & i \neq j \end{cases},$$

$$\rho_{ij} = s(i,j) + \eta_{ij} + \tau_{ij},$$

$$\tau_{ij} = \begin{cases} \sum_{k \neq j; k \in \text{ne}(i)} \max [\sigma_{kj}, 0] & i = j \\ \min [0, \rho_{ij} + \sum_{k \notin \{i,j\}; k \in \text{ne}(i)} \max [\sigma_{kj}, 0]] & i \neq j \end{cases},$$

$$\sigma_{ij} = s(i,j) + \eta_{ij} + \alpha_{ij}.$$ 

For deriving the message update equation for $\tau$, $\sigma$, $\beta$, $\rho$ we follow the max-sum message update rule from a factor node to a variable node in Chapter 8 of [6]. In (7) and (8) $\text{ne}(j)$ represents set of nodes sharing same parent exemplar as neighbors of $j$. Note that we do not need to check all neighbors of $j$, but just one child node among all neighbor nodes, since all children in $\text{ne}(j)$ must already share the same parent exemplar. These message update equation will make our model favor the valid configuration, which maximizes the objective function $S(c_1, \cdots, c_N)$. 

3.2 Using RAP to Learn Folksonomies

Ref. [1] demonstrated that folksonomies can be learned by using RAP to aggregate structured data contained in saplings extracted from Flickr users. Clustering root nodes of individual saplings extends the breadth of the learned folksonomy, while clustering root node of one sapling to a child node of another extends the depth of the folksonomy.
Figure 3: Different results made by different models (specialists nodes highlighted). (a) Before including all specialist saplings. (b) After including all specialist saplings & using higher preference values for specialist nodes, which are highlighted in yellow.

We claim that RAP also provides a natural framework for integrating specialists’ knowledge in the folksonomy learning process. We can do this simply by giving the nodes from saplings created by specialists higher preference, or self-similarity, values. This means the specialist nodes will be more likely to become exemplars, and specialist knowledge will guide the folksonomy learning process.

4 Validation on Flickr Data

We measure the impact of using specialist knowledge on folksonomies learned from Flickr data using RAP. Our data set consists of 20,759 saplings created by 7,121 users. A node is a collection or a set. The tags of all photos within a set are assigned to the set node and propagated to the collection node. We measure similarity between a pair of nodes \( i \) and \( j \) by the number of common tags \( t_{ij} \) they have among their top 40 tags: \( S(i,j) = \min(1, t_{ij}/4) \). We infer exemplars and clusters by initializing all messages to zero and each iteration we update exemplar assignments until it is converged. We checked convergence by monitoring total number of exemplars and the stability of the net similarity value.

We selected three terms as seeds for the folksonomies: “africa”, “sport”, and “animal”. To learn each folksonomy, we first need to obtain a set of relevant saplings. The baseline methodology for collecting relevant saplings is the following. First we retrieve all saplings from our data set whose root is the same as the seed term; next, we retrieve other saplings whose roots have the same name as the child nodes of these saplings. We include specialist knowledge in one of two ways: (1) use the baseline methodology above to collect relevant saplings created by the 18 specialists our annotators identified (see Sec. 2); (2) use all saplings created by the relevant specialists.

Besides varying the amount of specialist knowledge used by the learning algorithm, we can also vary the degree to which specialist knowledge is used in the inference process: (1) treat all users uniformly by setting preference values of all nodes to the mean of similarity scores; (2) use specialist knowledge by setting all preference values as above, while preference values for specialist nodes are set to twice the mean.

We ran RAP in four different settings to learn the folksonomies for each seed term: (1) relevant saplings with no differentiation between folk and specialists (same preference values); (2) relevant saplings, but higher preferences for specialists; (3) all saplings from specialists, same preference values; (4) all saplings and higher preference values for specialists. While the algorithm generally produces several trees, we evaluate only the most ‘popular’ tree, one that aggregates the greatest number of saplings. We measure the quality of the tree by net similarity. The higher the net simi-
Table 1: Four different model comparison on “africa”, “animal”, and “sport” folksonomies

<table>
<thead>
<tr>
<th>Sapling sets</th>
<th>Specialist preference</th>
<th>Net similarity</th>
<th>Total number of nodes</th>
<th>% specialist nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>africa</td>
<td>animal</td>
<td>sport</td>
</tr>
<tr>
<td>Expand from seed term</td>
<td>mean</td>
<td>48.2</td>
<td>198.5</td>
<td>36.5</td>
</tr>
<tr>
<td>Expand from seed term</td>
<td>2*mean</td>
<td>46</td>
<td>201.5</td>
<td>36.8</td>
</tr>
<tr>
<td>+ All specialist saplings</td>
<td>mean</td>
<td>54.8</td>
<td>589.3</td>
<td>36.3</td>
</tr>
<tr>
<td>+ All specialist saplings</td>
<td>2*mean</td>
<td>56.8</td>
<td>676.5</td>
<td>36.0</td>
</tr>
</tbody>
</table>

Larity, the more similar nodes were merged. In addition, in Table 1 we also report the size of each learned tree and the percentage of specialist nodes in it.

RAP in general leads to few or no structural inconsistencies, however, integrating specialist knowledge into the learning process improves the quality of the learned trees, improving the net similarity, while removing irrelevant concepts. For example, the root node in the “africa” folksonomy has a child node “christmas”, because some people spent the Christmas holidays in Africa. While this is not a problem, “christmas” is linked to other, possibly irrelevant concepts, which are introduced during sapling collection stage. Figure 3(a) shows the result of running the RAP without specialist knowledge. Note that “africa”→“christmas” is linked to nodes “family”, “card”, etc., which are irrelevant to Africa. Figure 3(b) shows the result of using RAP with specialist knowledge. Now the 40 nodes (“xmas,” “family,” “card,” etc.) originally placed under “africa”→“christmas” were moved to “christmas” under “holiday.” Moreover, “table mountain” and other nodes under “africa”→“cape town” were moved under “africa”→“south africa”→“cape town.” As we can see from this illustration, adding specialist knowledge helps produce a better folksonomy.

Another question is whether specialist knowledge alone is sufficient to produce high quality folksonomies. We observe that integrating specialist and folk knowledge leads to more comprehensive trees than using specialist knowledge alone. In the folksonomies learned by RAP, usually less than half of the nodes can be attributed to specialists.

5 Conclusion

In this paper, we propose additional constraint on RAP to learn structures and present the effects of integrating specialist knowledge. Our validation on three seed terms supports the claim that specialist knowledge improve the quality of learned folksonomies. In future work we would like to generalize our specialist identification process to automatically discover specialists.

Acknowledgments

This material is based in part on work supported by the National Science Foundation under Grant No. IIS-0812677.

References


