Partitioning instead of Clustering: An alternative approach for mining structured content in folksonomies

Bernhard Waltl
Technische Universität München
waltlb@in.tum.de

ABSTRACT

We present an alternative approach to offer a more structured way of navigating through folksonomies where user provided resource objects are assigned freely-chosen text labels (i.e. tags). The main idea behind this new method is to algorithmically determine different facets to enable faceted browsing. By offering facets, the resource collection can be accessed and filtered in a structured way. In contrast to the existing clustering concepts that primarily work with the similarity of tags, the partitioning approach mines facets, whereas the relationship between tags of one facet is that those tags do not appear on the same resource objects. The extensions of tags, i.e. the set of resources having the tag assigned, within the same facet are mutual exclusive, therefore we get disjoint partitions of the resource space. To determine those partitioning facets an algorithmic approach using linear programming was developed. The algorithm was applied to data samples from a real world folksonomy, namely the photo sharing platform Flickr. Although real world folksonomies are very noisy and different users with different vocabularies intensify this noisiness, representative and meaningful facets could be determined by the algorithm.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: Content Analysis and Indexing; H.4 [**Information Systems Applications**]: Miscellaneous; H.5.4 [**Information Interfaces And Presentation**]: Hypertext / Hypermedia, Navigation

General Terms

Folksonomies, facets, partitioning, navigation

Keywords

Social tagging, facet creation, faceted browsing, algorithm

1. INTRODUCTION

Many collaborative information systems use tags, i.e., freely-chosen text labels, as a means for categorizing contents created and uploaded by independent users (e.g. photos, bookmarks, music, podcasts) in order to make these contents accessible. Where hierarchical organization schemes fail, tags offer a way to navigate, search and browse through these collections of resources. The entire set of tag assignments in such a collection is called a folksonomy. These rather chaotic organization schemes exhibit no clear structure since there are no constraints in the usage of tags and users have different vocabularies. In combination with the fact that the number of resources can be very high this implies, that the utility of the system suffers if the user is not provided additional aid in analyzing a set of resources.

Information exploration using grouping is a very common method to design a user interface that supports information seekers. According to Hearst two methods are popular: *clustering* and *faceted categorization* [9].

The usual approach is to analyze the number of co-occurrences of tags and generate clusters of tags being frequently used together. Several algorithms for this purpose are described in literature, sometimes including specific user interfaces displaying the results to the user [2, 18, 20, 23, 25]. Those algorithms focus on determining strong relationships between resources based on the tags that they share with each other.

One of the big issues finding clustered tag sets is deciding whether co-occurrence count is significant or not [2]. Flickr provides so called *Flick clusters*, that group related tags together into several cluster. Looking at the clustering results for the word "Summer"[1], four clusters are proposed. Each cluster is represented by a few tags, that describe the resources of those clusters. Looking closer to those tags, it can be seen that the clusters do have relations within but they do also have relations to photos in some other cluster. For example, the text labels *cloud* and the plural form *clouds* are provided in different clusters but they obviously have a very strong relation with each other. The fact that different clusters are likely to share some resource objects holds for many examples that are generated by the Flickr approach of clustering (e.g. rain, animals). Whereas other clustering examples work better. Considering the provided clusters for

[1]http://www.flickr.com/photos/tags/summer/clusters/, accessed August 26th, 2012

the tag "Jaguar"[2], the suggested clusters represent the animal and the car brand in a meaningful way. The semantical coherence within a cluster is very high, whereas the semantic correlation to the other clusters is hardly given. Although *Flickr clusters* use even more parameters than occurrence (e.g. pageviews, comments left by users, etc. [2]), many of the provided clusters still lack consistency and intuitivity. Hearst furthermore showed that an additional disadvantage of clustering is the conflation of many dimensions simultaneously. Users prefer understandable hierarchies at a uniform level of granularity [9].

The alternative way to explore through folksonomies is using faceted categorization. Faceted browsing is a suitable method to provide exploratory search within a folksonomy. The presence of multiple facets enables a more flexible and enhanced search. To offer multidimensional faceted browsing, the objects of interests have to be classified along with several dimensions of metadata [14]. An online interface that exemplary shows the support through faceted categorization by providing useful different facets, is the Flamenco project[5]. For the example database of Nobel Prize winners many different facets such as gender, country, affiliation, year and prize are provided.

Section 2 reviews the studies of related work. In Section 3 a more detailed view on the idea of deriving facets from the folksonomy as well as on the implementation is given. Furthermore some limitations are briefly discussed. Section 4 presents the experimental results of the algorithm. Finally a conclusion is made in Section 5.

2. RELATED WORK

Discover latent structure from text-labeled objects is a well studied problem and many different methods were developed and exhaustively investigated in experimental setups. However, most of the approaches assume hierarchical structure of the content that is processed via similarity relation or using a lexical database such as WordNet [3] [6, 16, 21, 22]. If no hierarchy is created, the algorithms cluster the objects regarding to their similarity, which is mostly detected using co-occurrence count of the tags that they share [3, 4, 8, 13, 18].

Li et al. suggested a method for semantic browsing by considering similarity between tags [13]. They observed that the semantic relationship between two tags is represented through their co-occurrence. Likewise, Hassan-Montero and Herrero-Solana interpret tag co-occurrence as an indicator of semantic similarity between tags [8]. Based on the work of Heymann and Garcia-Molina [11], Benz et al. proposed an algorithm to derive a semantic structure from folksonomies [3]. Similarity thereby is also measured and determined through the co-occurrence count.

Schmitz et al. proposed a further approach, based on association rule mining, to retrieve hierarchical structures from folksonomies [19]. Those rules can be used to determine subsumption relations or to recognize pairs of tags which occur together very frequently.

Text labels are organized within a facet in such a way, that they reflect the concepts relevant to a domain. Each facet is composed of an orthogonal set of categories. The main problem by defining such a facet is to specify the semantic relation between meaningful labels and what kind of category the facet describes. To solve this problem, the creation is done manually [9] or by building a semantic hierarchy using a lexical database like WordNet[6, 21, 22]. Providing meaningful facets without manual assigned categories and avoiding the construction of semantic relationships between tags with a lexical database is quite challenging [7].

Lin et al. presented an interface that enables exploratory search called "ImageSieve" [14]. They performed a user study demonstrating that faceted search based systems can help users to explore large collections and find relevant information more effectively. The clustering is done by semantic similarity using the Scatter/Gather method, introduced by Hearst and Pederson [10], to group objects in topically-coherent clusters.

In this paper, we introduce an alternative approach to mine patterns within a folksonomy and to enhance the usability of an exploratory search by providing facets that neither cluster objects nor assume hierarchical relationships between tags.

3. PARTITIONING ALGORITHM

Folksonomies can be represented by tripartite graphs since the structure of those networks are composed of three kinds of nodes, i. e. users, resources and tags [12]. In our approach, as in most other methods, we make no distinction between different users and therefore we consider the network as a bipartite graph consisting of a set of resources and a set of tags. A partition of a set is a division into subsets. Those subsets are mutually exclusive and collectively exhaustive.

As mentioned above the idea behind faceted search is to provide tags, that are semantically orthogonal within a dimension. For example if we consider a color facet, we expect "red", "green", "blue", etc. to be represented in this facet. The same idea holds for many different facets such as time periods, spatial information, etc. However, the main problem is to extract those facets dynamically from the folksonomy. The goal of our algorithm is to partition the set of resources into non-intersecting sets with regard to their tags. The algorithm finds sets of tags of which the extensions are mutually exclusive. No tag of a facet shares a resource with another tag of the same facet.

To illustrate this, Figure 1 shows an example of a possible small folksonomy. The example consists of twelve resources *R1, ..., R12* and seven different tags *t1, ..., t7*. An optimal partitioning is easy to spot. Since *t1, t2, t3* cover the whole set of resources and furthermore are mutually exclusive regarding to their resources they seem to be semantically orthogonal within the same dimension. While the tags *t4, t5, t6, t7* do not cover all resources and since *t5* and *t6* share a resource, namely *R12*, they are not supposed to represent a common category. The same argument holds

[2]http://www.flickr.com/photos/tags/jaguar/clusters/, accessed August 26th, 2012

[3]http://wordnet.princeton.edu/, accessed August 26th, 2012

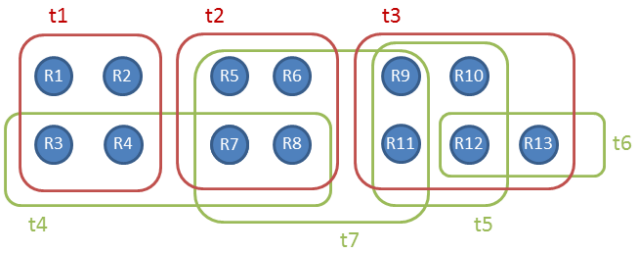

Figure 1: Tag - Resource relationship

for all other combinations of tags. However, searching for similarities would lead, depending on the clustering strategy, most likely to one clusters. Since all tags share many resource objects with some other tag, many algorithms that seek for co-occurrence would detect a strong similarity relationship between those tags. To get a more illustrative example lets assume that the resources are pictures and *t1, t2, t3* contain time information, e.g. 2010, 2011 and 2012. Those would belong to the same facet since they are representatives of the same category. It is not common that a picture is tagged with more than one text label representing the time information.

Figure 1 is an idealized example of a possible dataset. The existence of tags like *t1, t2, t3*, that have completely disjoint resource sets and additionally cover the whole resources space cannot be assumed in general, especially not in folksonomies with thousands of users. Therefore the algorithm tries to extract facets even on subsets of the resources within a folksonomy and can be configured to allow an overlap between the resource sets of different tags. The partitions get fuzzy, i. e. they are no longer necessarily mutually exclusive, and the union of them won't represent the whole folksonomy. Neglecting the two constraints, namely mutually exclusiveness and collectively exhaustiveness, can be accepted as they are very restrictive and in general not lead to meaningful results in noisy real-world folksonomies.

3.1 Implementation as a Linear Program

In order to determine the tags to generate those partitioning facets, an algorithmic way to determine those from the folksonomy is necessary. All the information, that can be used comes from the folksonomy itself: the tags, the resources and the relationship between those objects, i.e. labelling. The output should be a set of tags that are within a facet and therefore represent different categories of the same dimension. Since this can be represented as a binary value, we need a vector x that holds a value for each possible tag that indicates whether it is in the facet or not.

1: The corresponding tag is in the facet.
0: The corresponding tag is not the facet.

Furthermore, the tag selection has to satisfy a set of constraints, those result from the restriction that some tag must not appear along with some other tag within the same facet. To solve a problem of this kind liner programming provides a useful and well studied method. It is a common technique to solve optimization problems in various fields of study. As we will later see, it also allows us to assign a value to each tag, what gives us an opportunity to map the structure of the folksonomy better to the constraints.

Linear programming is a method to maximize a given linear function, subject to certain linear inequality constraints. The canonical form of a linear program is as follows:

$$\begin{aligned} \max \quad & c^T x \\ \text{s.t.} \quad & Ax \leq b \\ & x \geq 0 \end{aligned}$$

x represents the vector of variables, those are going to be determined by the solver. c and b are vectors consisting of known parameters. A is a matrix of coefficients. The expression, that is going to be maximized is called the objective function, which is $c^T x$. The linear inequalities are defined by $Ax \leq b$.

To solve the problem of generating facets with the optimization method of linear programming, the problem has to be formulated in an adequate way. The only information, that the algorithm uses is from the folksonomy, therefore all coefficients and parameters are extracted from it. First of all, the algorithm creates an extended adjacency matrix that represents the connection between two tags. A tag t is connected to some other tag t', if there is a resource that is labeled with both tags t and t'. The adjacency matrix stores the number of resources that the two tags share with each other. The algorithm continues to determine the values of the objective function, namely c, which we call the cost function, since it represents the "value" of a tag. Different tag values lead to different results, which we will discuss in the next section. By default the value of a tag is defined by its total occurrence count within the folksonomy. This information is already available, since it is represented by the elements of the main diagonal of the adjacency matrix.

The next step is formulating the linear inequalities. There are two informations needed: the matrix A that represents the coefficients and the vector b that represents the right side of the inequalities. The algorithm iterates through all pairs of tags and checks if they have resources that they share with each other. This information is stored in the adjacency matrix of the folksonomy. If there is a resource, that holds both text-labels, then they must not appear within the same facet because they most likely do not belong to the same dimension which is the basic idea of the determination process and therefore a requirement for tags to appear in the same facet. To avoid that this two tags end up in the result a proper constraint has to be formulated. A row is added to matrix A that represents this constraint. An example of a row in the matrix:

$$0 \quad \ldots \quad 0 \quad 1 \quad 0 \quad \ldots \quad 0 \quad 1 \quad 0 \quad \ldots \quad 0$$

This gives linear inequality constraints of the following form:

$$x_0, \ldots, x_n \in \{0,1\}:$$
$$0 * x_0 + \ldots + \underbrace{1 * x_i}_{i^{th}tag} + \ldots + \underbrace{1 * x_j}_{j^{th}tag} + \ldots + 0 * x_n \leq 1$$

The position of the 1-elements in the row are according to the tags, that are not allowed to appear simultaneously due

partitioning restrictions. A possible row that conforms to the example in Figure 1 would place the 1-elements in the first and in the fourth position. This states out, that Tag *t1* and *t2* cannot be part of a result of the facet extraction algorithm at the same time, since they both share resources. To complete the linear program every element of the vector b needs to be set to one, to ensure that the constraints defined exclude the co-appearance of two tags.

3.2 Adoption and Improvements

Overlapping Tags

Determining structure in folksonomies using tags can be very challenging through their noisiness and ambiguity [17]. Many times two different tags appear on the same resource, though they cover the same dimensional categories and should therefore appear in the same facet.

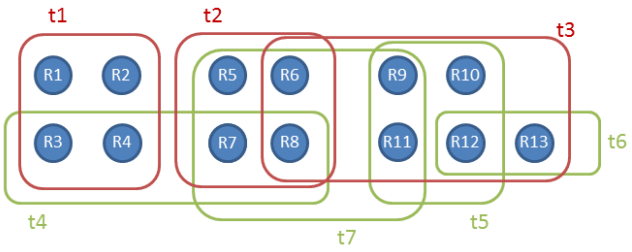

Figure 2: Overlapping tags

Figure 2 illustrates the problem. The example shows in general the same relationship as in Figure 1 but the tag *t3* is now labeled on an additional resource namely *R8*. Although a good facet would still be *t1, t2, t3*. However, the first naive approach would not find those overlapping tags, because of the restrictive constraints, that do not allow for any intersections between resources of tags. So we extended the algorithm by an additional parameter called *allowedOverlap*. This parameter is a percentage and considered while creating the constraints. If the count of intersecting resources of two tags is less than the number of *allowedOverlap* per cent of resources of the tag with less occurrences, then no constraint is generated since this overlap is accepted.

Minimum Appearance

The idea behind collaborative tagging systems is to allow users to assign a resource without any constraints or vocabularies that prescribe which words to use or how to spell them. As a consequence to this unsupervised annotating concept, a huge amount of different tags appear in those tagging systems. The frequency distribution of tags in real-world folksonomies follow the so called *power law* distribution. This means that there is a large number of occurrences in the head and a very low number of tag occurrences in the long tail [13, 20, 24]. Figure 3 shows quantitatively the ordered tag distribution trend for an excerpt dataset of the Flickr platform. The dataset represents the Flickr group "Munich, Germany"[4] and holds more than 30000 different photos.

[4]"Munich, Germany", http://www.flickr.com/groups/munich/, accessed on October 11th, 2012

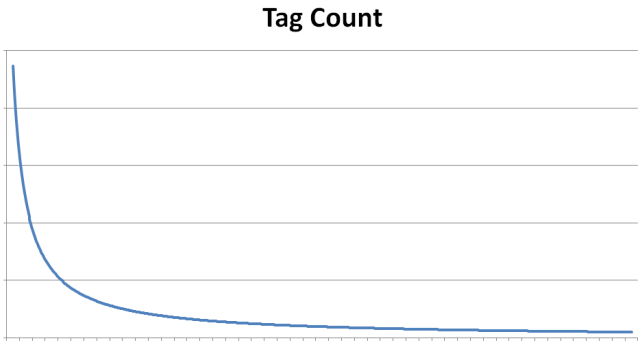

Figure 3: A qualitative overview of tag appearances from a Flickr group in decreasing order

Tags that appear at a very low frequency are most likely tags that belong to a single user or to a small group of users and are not very expressive for users that do not belong to this group. However, those tags are very distinguishing to most of the other tags and therefore they would represent a small set of resources without intersecting with other text labels. That is way those tags are very likely to turn up in the partitioning facet although they are not very relevant to a user that does not have any knowledge of the folksonomy and furthermore do not have a relationship to the very low frequency tags from another user.

To prevent this low frequency tags to appear in the facet, the algorithm was extended by a parameter called *minimum appearance*. The meaning is straightforward. Any tag that is considered by the algorithm to be in the facet needs to occur at least as often as the value of *minimum appearance* demands. Otherwise the algorithm ignores this tag. Beside more meaningful facets, this improvement leads to better performance since less tags need to be considered.

Exclude Tags

Rather than offering only one partitioning facet, that represents one semantical dimension, it should be possible to offer more than one different facet to provide a variety of dimensions, enhancing the possibility of accessing the folksonomy. It can be expected, that there are different dimensions depending on the subject of the analyzed folksonomy, e.g. time, spatial, ... In order to find all these different facet types the algorithm requires several iterations. If a facet was determined all tags within would be excluded in the next repetition of the algorithm, therefore it is necessary to parametrize the algorithm to specify the tags that are not going to be considered.

3.3 The Cost Function

An additional improvement addresses the cost function. The basic idea of the linear program is to maximize (respectively, minimize) the product of $c^T x$. Since the vector x is variable, it represents determined tags that belong to a facet, and the constraints are fixed, the cost function has huge influence on the outcome of the algorithm. The cost function assigns a value to each tag. There are several different ways to quantify the value of a tag.

The cost function assigns a positive value to each tag of a

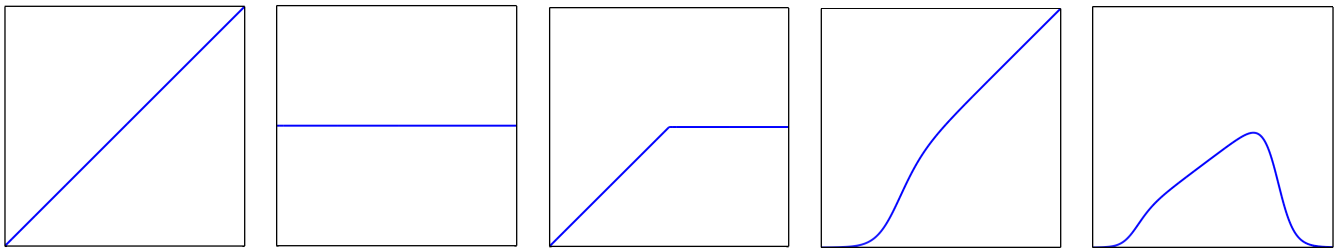

Figure 4: A qualitative comparison between the different cost functions, namely (a) Number of Occurrences (b) Uniformly (c) Capped (d) Logistic (e) Logistic with Cutoff. The x-axis hold the number of occurrences of a tag and the y-axis represents the value of this tag.

given set of tags $\mathcal{T}$:

$$C : t \to \mathbb{R}^+ \quad \text{with} \quad t \in \mathcal{T}$$

Furthermore, let us define the occurrence count of a tag within a folksonomy as

$$occ : t \to \mathbb{R}^+ \quad \text{with} \quad t \mapsto count(t)$$

Number of Occurrences

The most obvious approach mapping a value to a tag is assigning the counted number of occurrences. For this purpose it is necessary to determine the appearances of a tag in the folksonomy without any further processing, i.e. no additional mathematical method is required. This way of determining the value of a tag is quite simple and reflects the significance of tags since more important tags usually appear more often than less important tags. The resulting cost function is defined as follows:

$$C_{occ}(t) = occ(t)$$

Nevertheless there is an issue regarding the usage of this function. Since a few tags do have a very high occurrence (see figure 3), those are most likely to be in a facet because of the value maximization character of the linear program. Other tags, that appear less frequently throughout the folksonomy and share some resources with the high-valuable tag are no longer considered to be in the same facet because of the mutually exclusiveness of the tag-resource sets. Using this function often leads to facets with a very common tag beside a few tags with very low frequency that do not have any semantic relationship to each other.

Uniformly

Another way to assign costs to the tags is uniformly. In this case, every tag has the same value, i.e. one. This assignment neglects the structure of the tagging occurrence within the folksonomy, since the tags with the highest frequency are treated as valuable as those with low frequency. The resulting cost function is defined as follows:

$$C_{uni}(t) = 1$$

The main advantage, in contrast to the cost function that uses the number of occurrences, is that tags with very high occurrence do not prevent other tags, with lower frequency, to be considered by the algorithm. Since every tag is considered to be equal, the result accords more to the structure and relationships between distinct tags and their corresponding resources rather than on the frequency of occurrence. However, from this characteristic also arises the main disadvantage. The high frequency tags, are most likely the most popular ones. Facets created with this cost function may cover one semantical dimension but this dimension can be very specific, since it allows for very low frequent appearing tags to represent a facet. It cannot be assured that the semantic of such a facet can really provide a more structured access since the semantic issue of the facet may not reveals to the exploring user.

Capped

To avoid the problem that arises with the use of the appearance count as cost function and additionally not to loose the structure of the tagging occurrence, the algorithm can be configured to work with a further cost function we called "Capped Costs". This function assigns every tag the number of its appearances within the folksonomy as long as it is below a certain threshold Θ. If the occurrence count of a tag exceeds this threshold value, the value gets capped by assigning the value of the threshold. This leads to a mapping where tags with low occurrence are as valuable as they are regarding to their appearance and tags with very high frequency won't get too valuable since the absolute value is capped. The resulting cost function is defined as follows:

$$C_{capped}(t) = \begin{cases} occ(t), & \text{if } occ(t) \leq \Theta \\ \Theta, & \text{otherwise} \end{cases} \qquad \Theta \in \mathbb{R}^+$$

As a consequence, very popular tags are not going to prevent other tags to be in the facet, since the cost maximization of the linear program does not eagerly try to use this popular tag. This is the main advantage of this cost function. However, it is challenging to find the optimal threshold value since it depends very much on the folksonomy itself. If the threshold is too low, the cost function converges to the uniform distribution of tag values and on the other hand if it is too high, it converges to the cost function that represents the number of occurrences.

Logistic

At an initial stage, the growth rate of the curve is very low but is exponentially increasing until saturation is reached.

From the point of saturation the growth rate is decreasing till it stops. Curves with such an "S" shape are called sigmoid or logistics functions[5] and are defined by the exponential function:

$$\frac{1}{1+e^{-t}}$$

Therefore the resulting cost functions is defined by the formula:

$$C_{log}(t) = \frac{occ(t)}{1+e^{(-\alpha*(occ(t)-\beta))}}$$

A cost function with such a shape allows eliminating tags with low frequency whereas the value of high occurring tags is almost the number of their occurrence. Tags, that do not appear very often but also do not appear very rarely will still be considered, although their value, regarding to their occurrence count, is decreased by the cost function. The decreasing coefficient is determined by the logistic function and therefore a real value between zero, for low frequency tags, and one, for high occurring tags.

The shape of function C is influenced by two parameters α, β. Whereas β determines the point of inflection and α determines the rate at which the logistic function rises.

Although the logistic function provides a smooth transition from zero to the number of occurrences of a tag, the determination of the parameters α, β is challenging.

Logistic with Cutoff

A further development of the logistic cost function is the so called *cutoff* functionality. The basic idea is quite simple. The value assignment to tags is the same as in the logistic function but in this case, high occurring tags are going to get decreased values. If the occurrence of a tag value exceeds a threshold, the value will smoothly decrease, regarding to the difference to the threshold. A high difference will reduce the value down to zero. The threshold is a parameter of the algorithm and the attenuation is determined by a second logistic function that represents the suppression of frequently occurring tags. Therefore additional parameters α', β', that are used to determine the shape of the second logistic function, need to be set. The resulting cost functions is defined by the formula:

$$C_{logCutoff}(t) = \frac{occ(t)}{1+e^{(-\alpha*(occ(t)-\beta))}} - \frac{occ(t)}{1+e^{(-\alpha'*(occ(t)-\beta'))}}$$

The number of parameters to be defined increases and therefore the complexity of the cost function. This may be a disadvantage of this function. In general this method provides the possibility to define some kind of a "frequency window" that specifies what tags are going to be considered by the algorithm since an upper and a lower bound for occurrences can be specified using two logistic functions.

3.4 Limitations

Although the algorithm is simple and allows many improvements and adjustments it has some limitations.

[5]http://en.wikipedia.org/wiki/Logistic_function, accessed August 30th, 2012

Titling the Facets

The algorithm determines tags, that have a certain relationship to each other: they are not or only rarely assigned on the same resource. Those tags are supposed to describe different categories of the same semantical facet, e.g. "red", "green", "blue". The semantical subject described by the facet cannot be detected by the algorithm. Therefore it is not possible to assign a meaningful name to the facet. To solve this problem it is either possible to consult a lexical database or to assume and determine a hierarchical structure within the folksonomy and find a tag subsuming those tags in the facet. Anyway, both mentioned approaches are not implemented in the algorithm as is.

Performance

Depending on the size of the analyzed folksonomy, formulating the linear program, i.e. creating constraints and calculating tag values, is usually negligibly fast, solving, in general, is not. This restricts the usage of the algorithm in web applications. Providing instantly calculated facets can only be achieved on small folksonomies with a low number of resources and tags. A reliable estimation about computing time cannot be given, since it depends on many different factors, i.e. the number of resources, tags, constraints, et cetera.

Cost Function

Although the cost function can be varied in many different ways, it is still limited. The cost functions calculate a value for every tag but do not consider other tags that are determined to be in the facet. This calculation of the values is done independently of those other tags. E.g. if a pair of two tags *t1* and *t2* appear together in the same facet it is not possible to reward this combination with an additional value or to specify a negative value as a deduction.

4. EXPERIMENTAL RESULT

In the following experimental setup we extracted facets from real-world folksonomies using the partitioning algorithm. We chose Flickr as a photo sharing platform that allows users to upload and tag personal photos. Due to it's popularity it provides many photos and tags are an important part of the system and represent a primary navigational tool [15].

4.1 Dataset

For the evaluation we have selected Flickr photos through the Flickr API. The photos are extracted from the public Flickr group "Munich, Germany"[6]. To avoid resources with very few tags, we only imported those photos with more than three tags.

Number of Photos	31711
Number of distinct Tags	30860
Number of Users	3728

Table 1: Basic information from the imported Flickr Group "Munich, Germany"

[6]"Munich, Germany", http://www.flickr.com/groups/munich/, accessed on October 11th, 2012

4.2 Experimental Setup

The import of the photos as well as the implementation of the algorithm was done in Tricia[7], an open-source Java platform used to implement enterprise web information systems but also social software solutions including wikis, blogs, file shares and social networks [1].
Furthermore, Tricia offers a way to navigate and to interactively explore the folksonomy using a web browser. We prototypical implemented the algorithm and offered the partitioning facets as the navigational tool. Additionally to the tags the number of resources on which they are labeled on is displayed in brackets beside the tag (see Figure 5). This information is useful because it supports the navigation process by providing more information about the qualitative structure of the folksonomy.
Using this implementation, we navigated through the folksonomy to search for meaningful partitioning facets. Since the algorithm can be applied to subsets of the folksonomy, we searched for subsets, that are consistently tagged and that fulfill the requirement of mutual exclusion regarding to their resources. During this task, we found meaningful and representative facets on different subsets.

4.3 Facets

If subsets of the folksonomy are considered the partitioning algorithm is able to extract consistent and meaningful facets, that represent different aspects of the same dimension. Figure 5 shows two facets generated for a subset of the Flickr photos. The facet (a) is extracted from the photos, that are all labeled with the tag "Canon", whereas the photos from facet (b) share the label "Nikon". Both are very common camera manufacturer with a variety of different models. Using those sets of photos the algorithm was able to determine significant and representative facets. The algorithm extracted the models from Canon and Nikon.

(a)

Partitioning Facet	
1000d	(37)
300d	(57)
350d	(50)
400d	(120)
40d	(37)
450d	(44)
500d	(135)
550d	(87)
5dmarkii	(108)
7d	(60)
ixus	(39)
moment	(41)
powershot	(85)
robert	(35)

(b)

Partitioning Facet	
d100	(26)
d200	(39)
d300	(69)
d40	(84)
d50	(83)
d5000	(45)
d700	(97)
d70s	(33)
d80	(97)
d90	(129)
film	(35)
instantfave	(122)
soninka	(23)

Figure 5: Two different facets both using the cost function "Logistic with Cutoff" (a) created on the subset "Canon" and (b) created on the subset "Nikon".

[7]"Tricia", infoAsset AG, http://www.infoasset.de, accessed on October 19th, 2012

Within facet (a) there are two tags that are not related to "Canon", namely "robert and "moment", whereas in (b) the tags "film", "instantfave" and "soninka" are not related to the camera models of "Nikon". Those wrongly detected tags arise, again, from the circumstance that real-world folksonomies are noisy and differ in the tagging behavior of users.

A further example of a meaningful partitioning facet is shown in Figure 6. The facet is generated from the Flickr photos, that are commonly tagged with the label "Museum" and therefore hold many different museums of Munich. Again, the algorithm is able to find very representative tags, that refer to the museum photos within the Flickr Group "Munich, Germany". Almost all tags refer to museum in Munich except the tag "Musèe", which generally describes photos from different museums, that are not tagged with any particular museum.

Partitioning Facet	
altepinakothek	(38)
artgallery	(34)
bmw	(148)
brandhorst	(137)
deutsches	(67)
glyptothek	(39)
musée	(40)
neuepinakothek	(18)
residenz	(17)
theatermuseum	(41)

Figure 6: A facet generated for the "Museum" tag, using the cost function "Logistic with Cutoff".

Figure 7 shows a further facet that is based on the whole set of photos. It uses the cost function "Logistic with Cutoff" but in contrast to the facets shown in Figure 5 and 6, it allows a small overlap of the tags. As mentioned, this overlap affects the extensions of a tag, i.e., photos labeled with the tag. The dilution, that those extensions need no longer be disjoint partitions, leads to facets with much more tags.
Although allowing overlapping extensions is a useful adaption of the algorithm and leads to meaningful facets, the threshold value of the overlap is difficult to determine. As a result of this removed restriction the facet holds many more different tags, that may share some resources.
Therefore, tags that belong to many distinct dimensions are in the facet. Through their variety of dimensions they should not appear within the same facet. The facet in Figure 7 shows date information as well. The tags representing the years from 2006 to 2010 are discovered. Beside the date information, tags are shown, that are not related to the date information, e.g. "5d", "allemagne", "analog", et cetera. Even though 11 tags are shown in the figure, there are in fact 38 tags within this facet.

Anyway, within these 38 tags there are more than just the dimension that represents the date information. A "color" dimension and a "district" dimension are included. Even if the threshold holds a value that allows for extracting the date information, tags that are not related are still determined by the algorithm.

Figure 7: The first 11 tags of a partitioning facet that was generated using all photos, with the cost function "Logistic with Cutoff". Additionally, a small overlap (10%) of the extensions of the tags is allowed.

Many different dimensions are mixed in one single facet, because of the overlap. The usage of the combination of the cost function and the overlap parameter is not sufficient to determine facets that only contain facets of one dimension.

Table 2 gives an overview of the different cost functions that were implemented. It allows a quantitative comparison of the number of tags of the same dimension to the total number of tags that are within a facet. For instance, if we consider the resulting facet for the subset of resources that are labeled with the tag "Canon" using the cost function "Logistic with Cutoff", then a facet with 14 tags in total is the result. 12 of this 14 tags are of the same dimension (see Figure 5). The classification and counting of the tags that share the same dimension was done manually.

Tag	Number of Occurrences	Uniformly	Capped	Logistic	Logistic with Cutoff
Canon	1/2	4/14	4/14	1/2	12/14
Nikon	0/3	7/15	7/15	0/2	10/13
Olympus	4/15	4/16	3/17	0/1	10/12
Museum	0/1	3/17	3/17	0/1	8/10

Table 2: Comparison of the cost functions.

From the results in the table some conclusions can be derived. First of all, there are big differences between the results using the various cost functions. Whereas "Logistic" leads to very few tags within a facet, the usage of "Uniformly", "Capped" and "Logistic with Cutoff" results in facets with much more tags. This is in line with the way that those cost functions assign values to the tags. An explanation for this effect can be found by considering the cost functions "Number of Occurrences" and "Logistic". They enable very frequent occurring tags more likely be in a facet since they have very high values, respectively the other cost functions do not allow for very high values (see Figure 4). It can been seen, that if a cost function allows high values for a tag, the determination of distinct tags of one dimension suffers. Without a maximum value for tags as an upper bound, the algorithm is hardly able to identify meaningful facets.
Furthermore, the table shows, that even if cost functions prevent tags from being too valuable to skew the resulting facets this does not automatically lead to representative facets. Although the cost functions "Uniformly" and "Capped" have a maximum value for tags, the facets that are determined are in general no of one dimension. Furthermore, the tags within those facet seemed to be a chaotic selection without a clear structure.
Meaningful and representative facets are discovered when the algorithm uses the cost function "Logistic with Cutoff". This function excludes tags with very low frequency as well as tags with very high occurrence. This method seems to be the best way to determine the value of tags, since the resulting facet holds the most tags of one dimension in every of the four cases considered in Table 2.

Figure 8 is a graphical visualization of the comparison of the cost functions. Two different diagrams are shown, whereas (a) provides information about the precision and (b) displays the recall of the cost functions. It can be seen, that the function "Logistic with Cutoff" has a high precision value, what can been derived from Table 2 also.
To calculate the recall value, the number of relevant camera models or different museums would be necessary. Counting this values is in practical terms not possible, since there are many different tags throughout the folksonomy, whereas each of them has to be considered and classified. Therefore we took the determined values from the cost function "Logistic with Cutoff". This explains the recall of one for the function "Logistic with Cutoff". The recall value confirms what we have already seen in Table 2. The cost functions "Number of Occurrences" and "Logistic" do not lead to meaningful and representative facets, whereas "Logistic with Cutoff" has a high precision and discovers relevant tags, that no other cost function was able to determine.

4.4 Summary

Although the general principle of the algorithm is simple, meaningful and representative facets could be determined. Many different categories of a dimension could be discovered without any additional preparation of real-world folksonomies. However, every facet that was determined still lacks consistency since no dimension could be extracted purely, i.e., without a tag that does not belong to the facet.

The cost functions have a material influence on the resulting facets. Depending on the assignment of values to tags the resulting facet can be very representative or totally unusable for any further navigation or exploration.

Adapting the algorithm using the parameters *maxOverlap* or *minAppearence* is challenging since it is an issue, that arises from the dataset. Again, noisiness and ambiguity of tags within the folksonomies make it hard to determine values that yield good partitioning facets.

5. CONCLUSION AND OUTLOOK

This paper presents an attempt to offer a more structured way to access and navigate through folksonomies. Providing so called partitioning facets that cover multiple categories of a dimension can be seen as an additional aid to support users that have no or less knowledge of a folksonomy. The set of

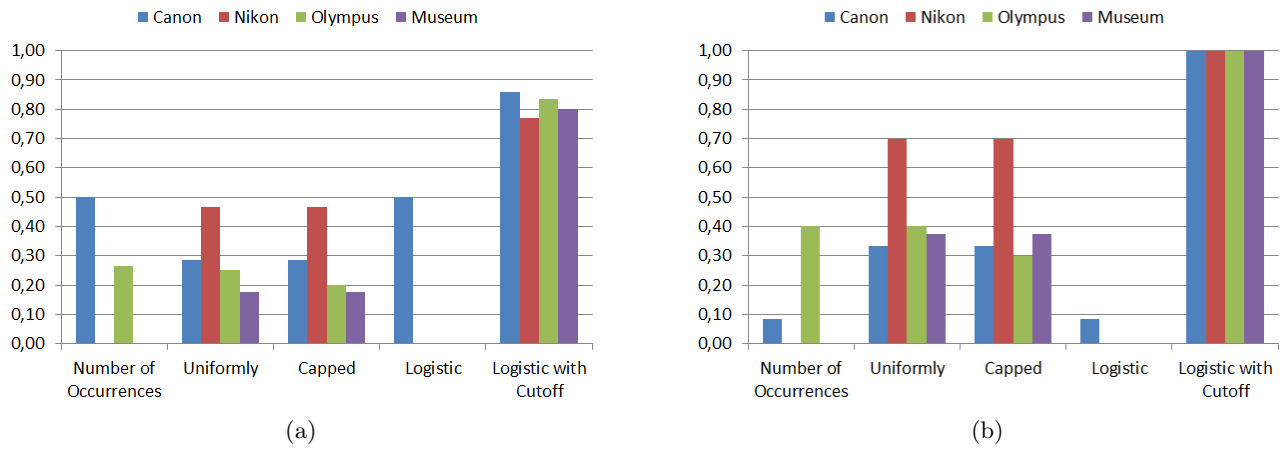

Figure 8: A comparison between the cost functions regarding to (a) precision and (b) recall.

tags, that represents a facet, is algorithmically determined. Since the relation between those tags have to satisfy certain restrictions this approach can be formulated as a linear program.

To obtain partitioning facets, a set of photos with tag labels from the photo sharing platform Flickr was extracted. Using this folksonomy, facets were generated by the algorithm. Although the algorithm was able to determine some very meaningful and representative facets, there remain challenges that arise from the noisiness within the folksonomy. Since users differ in their vocabulary as well as in their tagging behavior the extracted partitioning facets often lacks consistency. This problem arises since the folksonomies are very noisy and consist of resources from many different users with different vocabularies and distinct tagging behavior.
It can be assumed that this issue cannot be solved by adapting the cost functions or the parameters of the algorithm only. Without any additional preprocessing that lowers the noisiness this approach won't produce results that do not longer lack consistency. Harmonizing the tags and structuring folksonomies, as introduced by Matthes et al. [16], would decrease this noisiness and therefore represent an important step towards better results.

Additionally more research is needed to make the approach more applicable. Determining thresholds for the cost functions as well as developing new ones may also lead to better results.

However, this paper shows, that there are ways to automatically determine facets and furthermore adds a new way to discover a latent structure from text-labeled objects. This may be more interesting in collaborative tagging systems, where noisiness within Folksonomies is not as high as in online platforms such as Flickr. Beside existing concepts to enhance navigation and exploring, namely clustering and hierarchical subsumption, the partitioning approach can be an additional method to improve the accessibility of folksonomies.

6. REFERENCES

[1] T. Büchner, F. Matthes, and C. Neubert. Data model driven implementation of web cooperation systems with tricia objects and databases. volume 6348 of *Lecture Notes in Computer Science*, pages 70–84. Springer Berlin / Heidelberg, 2010.

[2] G. Begelman, P. Keller, and F. Smadja. Automated tag clustering: Improving search and exploration in the tag space. *Proceedings of the Collaborative Web Tagging Workshop (WWW'06)*, 2006.

[3] D. Benz, A. Hotho, S. Stützer, and G. Stumme. Semantics made by you and me: Self-emerging ontologies can capture the diversity of shared knowledge. *Proceedings of the Web Science Conference 2010*, 2010.

[4] C. H. Brooks and N. Montanez. Improved annotation of the blogosphere via autotagging and hierarchical clustering. *Proceedings of the 15th international conference on World Wide Web*, 2006.

[5] B. Chun, A. Elliott, J. English, M. Hearst, K. Li, R. Sinha, K. Swearingen, and K.-P. Yee. Flamenco home. *http://flamenco.berkeley.edu/*, accessed on 30th August, 2012.

[6] W. Dakka, R. Dayal, and P. G. Ipeirotis. Automatic discovery of useful facet terms. *Proceedings of the ACM SIGIR 2006 Workshop on Faceted Search, 2006*, 2006.

[7] S. Giovanni Maria and Y. Tzitzikas. *Faceted Taxonomy-Based Sources Dynamic Taxonomies and Faceted Search*. Springer Berlin Heidelberg, 2009.

[8] Y. Hassan-Montero and V. Herrero-Solana. Improving tag-clouds as visual information retrieval interfaces. *InScit2006: International Conference on Multidisciplinary Information Sciences and Technologies*, 2006.

[9] M. A. Hearst. Clustering versus faceted categories for information exploration. *Magazine Communications of the ACM - Supporting exploratory search*, 49(4):59–61, 2006.

[10] M. A. Hearst and J. O. Pedersen. Reexamining the cluster hypothesis: scatter/gather on retrieval results. *Proceedings of the 19th annual international ACM*

SIGIR conference on Research and development in information retrieval, pages 76–84, 1996.

[11] P. Heymann and H. Garcia-Molina. Collaborative creation of communal hierarchical taxonomies in social tagging systems. Technical report, Stanford InfoLab, 2006.

[12] R. Lambiotte and M. Ausloos. Collaborative tagging as a tripartite network. *Lecture Notes in Computer Science*, 3993:1114 – 1117, 2005.

[13] R. Li, S. Bao, Y. Yu, B. Fei, and Z. Su. Towards effective browsing of large scale social annotations. *Proceedings of the 16th international conference on World Wide Web*, pages 943–952, 2007.

[14] Y. Lin, J.-w. Ahn, P. Brusilovsky, D. He, and W. Real. Imagesieve: exploratory search of museum archives with named entity-based faceted browsing. *Proceedings of the 73rd ASIS&T Annual Meeting on Navigating Streams in an Information Ecosystem - Volume 47*, pages 1–10, 2010.

[15] C. Marlow, M. Naaman, D. Boyd, and M. Davis. Ht06, tagging paper, taxonomy, flickr, academic article, to read. *Proceedings of the seventeenth conference on Hypertext and hypermedia*, pages 31–40, 2006.

[16] F. Matthes, C. Neubert, and A. Steinhoff. Structuring folksonomies with implicit tag relations. *Proceedings of the 23rd ACM conference on Hypertext and social media*, pages 315–316, 2012.

[17] A. Plangprasopchok, K. Lerman, and L. Getoor. Growing a tree in the forest: constructing folksonomies by integrating structured metadata. *Proceedings of the 16th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 949–958, 2010.

[18] D. Ramage, P. Heymann, C. D. Manning, and H. Garcia-Molina. Clustering the tagged web. *Second ACM International Conference on Web Search and Data Mining (WSDM 2009)*, 2008.

[19] C. Schmitz, A. Hotho, R. Jäschke, and G. Stumme. Mining association rules in folksonomies. *Data Science and Classification: Proceedings of the 10th IFCS Conference, Studies in Classification, Data Analysis and Knowledge Organization*, pages 261 – 270, 2006.

[20] B. Sigurbjörnsson and R. v. Zwol. Flickr tag recommendation based on collective knowledge. *Proceedings of the 17th international conference on World Wide Web*, pages 327–336, 2008.

[21] E. Stoica, M. Hearst, and M. Richardson. Automating creation of hierarchical faceted metadata structures. *Proceedings of the Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics*, pages 244 – 251, 2007.

[22] E. Stoica and M. A. Hearst. Nearly-automated metadata hierarchy creation. *Proceedings of the Human Language Technologies 2004: The Conference of the North American Chapter of the Association for Computational Linguistics*, pages 117–120, 2004.

[23] P. Venetis, G. Koutrika, and H. Garcia-Molina. On the selection of tags for tag clouds. *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 835–844, 2011.

[24] H. Wu, M. Zubair, and K. Maly. Harvesting social knowledge from folksonomies. *Proceedings of the seventeenth conference on Hypertext and hypermedia*, pages 111–114, 2006.

[25] R. v. Zwol, B. Sigurbjornsson, R. Adapala, L. G. Pueyo, A. Katiyar, K. Kurapati, M. Muralidharan, S. Muthu, V. Murdock, P. Ng, A. Ramani, A. Sahai, S. T. Sathish, H. Vasudev, and U. Vuyyuru. Faceted exploration of image search results. *Proceedings of the 19th international conference on World wide web*, pages 961–970, 2010.