Frontiers of Information Technology & Electronic Engineering
www.zju.edu.cn/jzus; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

A social tag clustering method based on common co-occurrence group similarity*

Hui-zong LI†‡1,2, Xue-gang HU†1, Yao-jin LIN†3, Wei HE1, Jian-han PAN†1

(1School of Computer and Information, Hefei University of Technology, Hefei 230009, China)
(2School of Economics and Management, Anhui University of Science and Technology, Huainan 232001, China)
(3School of Computer, Minnan Normal University, Zhangzhou 363000, China)
†E-mail: lihz_aust@sina.com; jsjxhuxg@hfut.edu.cn; yjlin@mnnu.edu.cn; peter.jhpan@gmail.com

Received June 11, 2015; Revision accepted Sept. 11, 2015; Crosschecked Dec. 9, 2015

Abstract: Social tagging systems are widely applied in Web 2.0. Many users use these systems to create, organize, manage, and share Internet resources freely. However, many ambiguous and uncontrolled tags produced by social tagging systems not only worsen users' experience, but also restrict resources' retrieval efficiency. Tag clustering can aggregate tags with similar semantics together, and help mitigate the above problems. In this paper, we first present a common co-occurrence group similarity based approach, which employs the ternary relation among users, resources, and tags to measure the semantic relevance between tags. Then we propose a spectral clustering method to address the high dimensionality and sparsity of the annotating data. Finally, experimental results show that the proposed method is useful and efficient.

Key words: Social tagging systems, Tag co-occurrence, Spectral clustering, Group similarity
http://dx.doi.org/10.1631/FITEE.1500187 **CLC number:** TP311

1 Introduction

Social tagging is an important application in Web 2.0. It is a system of description, organization, management, and classification of Internet resources (Isabella, 2009). In a free and open environment, web users can choose personal terms (i.e., tags) to annotate web resources, according to their own perception or understanding of these resources. At present, there are many social tagging systems on the Internet. For example, Flickr (www.flickr.com) is a picture-sharing system that allows users to upload and tag their pictures. CiteULike (www.citeulike.org) is an academic paper-sharing system, in which researchers can annotate papers using their own scientific terms. Lastfm (www.last.fm) is a famed music community, in which users are permitted to tag songs, albums, and artists. Delicious (www.delicious.com) and Blogmarks (www.blogmarks.net) are social bookmarking systems, in which users can annotate any web page using tags. Bibsonomy (www.bibsonomy.org) provides both bookmarking and paper-sharing services. All of these systems belong to folksonomy, which was first proposed by Vander Wal (2004), who regarded folksonomy as a bottom-up social classification system. Shirky (2004) described folksonomy as a typical and flat name-space. Mathes (2004) viewed folksonomy as a collaborative classification method based on shared metadata in which tags are generated by users. A tag may be a word or a term in an existing vocabulary, or a new word created by users. In general, a tag, as a kind of user-generated

‡ Corresponding author

* Project supported by the National Natural Science Foundation of China (Nos. 61273292, 61303131, 51474007, and 51374114) and the MOE Humanities and Social Science Research on Youth Foundation of China (No. 13YJCZH077)

ORCID: Hui-zong LI, http://orcid.org/0000-0002-1459-989X metadata with uncontrolled vocabulary, has special lexical information, embodies collective intelligence of users, and plays a key role in social tagging systems. Except for the characteristics of keywords and terms in the resources, a social tagging system offers a supplement for web resources to express the user's comments or viewpoints, which is very helpful for information retrieval and personalized recommendation. In the theoretical field, social tagging has aroused a lot of interest, such as the meaning of the tag (Noruzi, 2006), the using motivation of the user (Furnas *et al.*, 2006), the visual interface of the tag (Rivadeneira *et al.*, 2007), and the information and semantic retrieval of the tag (Laniado *et al.*, 2007; Vanderlei *et al.*, 2007). In the application field, tagging techniques have also been widely applied in social media and online service systems; for example, embedded tag knowledge management methods were introduced to web systems to promote context-aware web services (Cuzzocrea and Mastroianni, 2003; Cuzzocrea, 2006).

In some tagging recommender systems, tags and tagging information have been applied to reconstruct users' interest profile or resource summary based on the tag vector space. However, all of these social tagging recommender systems have to face such problems as ambiguity, sparsity, redundancy, and lack of semantics, as the tags are of syntactic nature and in a free form. The first hindrance is that some users with different knowledge backgrounds or preferences may use different words to tag the same resource, which could result in ambiguity. The second hindrance is that some users are not willing to annotate resources with enough tags, which could bring out sparsity. The third hindrance is that a huge number of social tags are 'tag spam', which could lead to redundancy. These problems lower the performance of social tagging systems in discovering ability, sharing capacity, and recommending accuracy. To alleviate these problems, the use of a tag clustering method has been introduced into social tagging systems. Tag clustering could expose the coherence of tags from the perspectives of how users tag and how resources are tagged. Undeniably, the tag cluster can reveal resource topic information or user interest profile in a more concise way, and thus mitigates the above problems to some extent.

The first task for tag clustering is to evaluate the relevance of tags, which can be measured by several methods. At present, the main method to assess the relationship between tags is based on the vector space model (VSM). First, the tag is represented as a weight vector based on resources or users, and the weight can be decided by the TF-IDF formula (Salton, 1983). Second, the tag vector space is established by composing all tags. Finally, Euclidean distance or cosine distance is used to calculate the similarity of tags over the tag vector space (Simpson, 2008). However, this method will suffer from high-dimensionality and sparsity problems when there are a lot of resources or users. In addition, some semantic information will be lost when the ternary annotating relation is transformed to a binary relation based on resources or users. The second method is using the similarity of tag content combined with an external semantic dictionary to find the relation of tags. However, this method suffers from the problem that new words and concepts though used, are not contained in the dictionary (Simpson, 2008). In the third method, the number of co-occurrences is directly used to measure tag similarity (Begelman *et al.*, 2006), which will alleviate the high-dimensionality and sparsity problems to a certain extent. Nevertheless, in the traditional method, tag co-occurrence means that two tags are used to describe the same resource, or two tags are used by the same user, which weakens the semantic relationship of tags, and brings semantic noise. In addition, the co-occurrence similarity between two tag individuals is not accurate enough to measure tag relevance, because their semantic relationships are influenced deeply by other information, such as their common co-occurrence group.

Most research applies the traditional clustering algorithms directly, such as agglomerative hierarchical clustering (Shepitsen *et al.*, 2008), or K-means (Noll and Meinel, 2007) for tagging data, which are designed to deal with the spherical structure. For an arbitrary shape data distribution, these traditional clustering algorithms find it hard to differentiate clusters accurately. In tag clustering, we know that the tagging data has a complex ternary relation, so the semantic relationship between tags is very complex, with the latent topics of tags largely overlapping and mixed. Therefore, the data distribution of tags may be irregular or arbitrary, which makes it difficult to simply apply the above clustering algorithms. As an alternative, spectral clustering can aggregate the objects in an arbitrarily shaped data space, and converges to a global optimal solution.

Hence, in this paper, we first analyze all statuses of tag co-occurrence, and look for a better tag co-occurrence form that maintains the tag semantic relationship perfectly. Then we extend the characteristic of tag co-occurrence and discuss the impact of the tag's common co-occurrence group on tag similarity, and acquire a new method to measure tag similarity based on the common co-occurrence group. After that, we build the tag's similarity affinity matrix and use a spectral clustering algorithm to aggregate the tags.

The method presented in this paper has two advantages. One is that we directly use tag co-occurrence to measure the tag semantic relevance, which can alleviate the high-dimensionality and sparsity problems, and shows good performance for a spectral clustering algorithm. The other is that the tag co-occurrence similarity measure method keeps the strongest semantic relevance of tags.

The rest of this paper is organized as follows. We review the related work in Section 2 and introduce the preliminaries in Section 3. The common co-occurrence group similarity between tags and its spectral clustering algorithm are discussed in Section 4. Experimental results are reported in Section 5. Section 6 concludes this paper and outlines future work.

2 Related work

Many studies have been carried out on tags in social tagging systems. Marlow *et al.* (2006) introduced some famous tagging systems and compared their differences. Furnas *et al.* (2006) discussed the using motivation of users and pointed out the working mechanism of the tagging system. Noruzi (2006) and Suchanek *et al.* (2008) discussed the meaning of tags when they were used to describe resources. Bischoff *et al.* (2008) segmented tags into eight types, and then pointed out how different types were used to annotate resources. Michlmayr and Cayzer (2007) considered that two tags had some semantic relation when they were used to annotate the same bookmark. Heymann and Garcia-Molina (2006) indicated that tags have some underlying structures.

Recently, using a tag clustering method to discover the emergent structure in tags has aroused much interest, and a lot of work has been carried out to improve the social tagging system. Deutsch *et al.* (2011) presented clustering tags in a tag cloud for improving user experience, where they conducted a study on the strengths and weaknesses of using tag clouds in common Web-based contexts. Vandic *et al.* (2011) proposed a method to enhance the search ability using social tagging systems; they used cosine similarity to measure semantic relatedness and aggregated syntactic variations of tags with the same meaning. Lehwark *et al.* (2008) employed U-map and emergent-self-organizing-maps (ESOMs) techniques to aggregate and visualize the tagging data, and discovered underlying structures in music collections.

Dattolo *et al.* (2011) proposed a method for detecting clusters of similar tags and relationships among them, and a clustering process was used to find the different classes of related tags. Knautz *et al.* (2010) introduced tag co-occurrence into the retrieval system by using a single-link clustering algorithm to improve the efficiency of information retrieval. Simpson (2008) introduced a divisive method upon a graph, in which tags corresponded to nodes and relevance values of tags were called edge weights; Simpson then divided the graph into unconnected sub-graphs by removing the lowest weighted edges. Begelman *et al.* (2006) proposed a partitioning-based clustering method, in which a spectral bisection was used to split data into two clusters recursively and a modularity function was used to decide whether the partition was acceptable. Cui *et al.* (2011) introduced a novel tag clustering method, TagClus, and they used a random walk-based approach to measure relevance between tags by exploiting the relationship between tags and resources.

Gemmell *et al.* (2008) used an agglomerative hierarchical clustering method to aggregate tags, and demonstrated how tag clusters serving as coherent topics could help in personalized recommendation (Shepitsen *et al.*, 2008) or personalized navigation. Xu *et al.* (2015) proposed a kernel information propagation tag clustering algorithm, in which they exploited the kernel density estimation of the directed k-nearest neighbor (kNN) graph as a start to reveal the prestige rank of tags in tagging data. Van Damme *et al.* (2007) assessed the similarity of two tags through their contents and information from terminological or lexical resources. They used Leo

Dictionary and WordNet to measure the semantic relationship between two tags, and based on this approach, the tags could be transformed to ontology. A similar work was presented by Markines *et al.* (2009), who also used WordNet as the semantic grounding to compute tag similarity; particularly, they ranked tag pairs by using the Jiang-Conrath distance (Jiang and Conrath, 1997), which integrates information theory and taxonomic knowledge.

The above methods can be successfully applied in all kinds of social tagging systems, but there are some problems, such as the lack of semantics (Begelman *et al.*, 2006; Simpson, 2008; Knautz *et al.*, 2010; Cui *et al.*, 2011; Dattolo *et al.*, 2011), high-dimensionality and sparsity (Gemmell *et al.*, 2008; Shepitsen *et al.*, 2008; Xu *et al.*, 2015). For example, the method proposed by Cui *et al.* (2011) is a resource-based co-occurrence approach, which directly transforms the ternary annotating relation to a resource-based binary relation. This approach misses the user's information, and weakens the semantic relationship between tags. In the method proposed by Xu *et al.* (2015), the tag was represented as a vector based on resources, and the cosine function of the angle was used to measure the similarity between tags, which would cause high-dimensionality and sparsity when there are a lot of resources, and would also weaken the semantic relationship of tags. Another method proposed by Van Damme *et al.* (2007) and Markines *et al.* (2009) cannot avoid the confusion of new words or concepts. The tag co-occurrence method presented in this paper considers resources and users simultaneously, and does not need to represent the tags as a vector space model, which can alleviate the above problems.

3 Preliminaries

3.1 Social tagging system model

A social tagging system is an open web application environment. It permits web users to annotate web resources freely by using tags. A social tagging system includes three entity elements (i.e., web users, web resources, and tags) and one relation element (i.e., the ternary annotation relation among them). The model of the social tagging system could be represented as a four-tuple model $F := (U, T, R, A)$, where U is a finite set of users, R a finite set of resources, T a finite set of tags, and $A \subseteq U \times T \times R$ a finite ternary relation set among users, resources, and tags. The element $a = (u, r, t) \in A$ represents an annotating behavior, which denotes user u using tag t to annotate resource r. The social tagging system model is shown in Fig. 1.

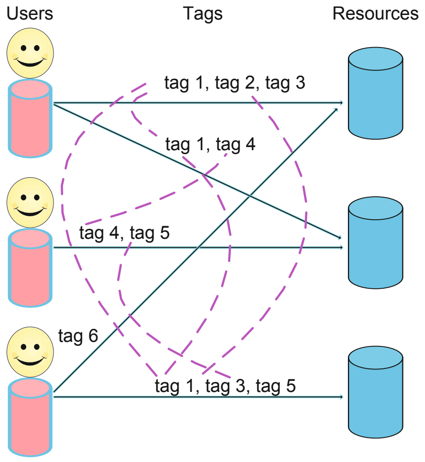

Fig. 1 Model of a social tagging system

3.2 Analysis of tag co-occurrence relationship

As pointed out by Michlmayr and Cayzer (2007) and Szomszor *et al.* (2007), the co-occurrence relationship of tags should be viewed as a kind of semantic relationship. Therefore, in this subsection, we analyze the co-occurrence relationship of tags and discover the most valuable tag co-occurrence status. We firstly define some sets for addressing some concepts:

$T(r_m) = \{t_x | t_x \in T\}$, which represents a tag set of resource r_m, where $r_m \in R$ and $T(r_m) \subset T$;

$T(u_l) = \{t_y | t_y \in T\}$, which represents a tag set of user u_l, where $u_l \in U$ and $T(u_l) \subset T$;

$T(u_l, r_m) = \{t_z | t_z \in T\}$, which represents a tag set that user u_l annotates on resource r_m, where $u_l \in U$, $r_m \in R$, and $T(u_l, r_m) \subset T$.

Hence, there are three statuses of tag co-occurrence:

Status 1: $t_i \in T(r_m)$ & $t_j \in T(r_m)$, which denotes that the two tags t_i and t_j appear in the same tag set $T(r_m)$. We then call t_i and t_j co-occurrence for the same resource.

Status 2: $t_i \in T(u_l)$ & $t_j \in T(u_l)$, which denotes that the two tags t_i and t_j appear in the same tag set $T(u_l)$. We then call t_i and t_j co-occurrence for the same user.

Status 3: $t_i \in T(u_l, r_m)$ & $t_j \in T(u_l, r_m)$, which denotes that the two tags t_i and t_j appear in the same tag set $T(u_l, r_m)$. We then call t_i and t_j co-occurrence for the same user and the same resource simultaneously.

As is shown in status 1, the two tags exist in the same resource's tag set. The user's influence on this co-occurrence status is ignored, and the ternary annotation relationship among users, resources, and tags is separated. Therefore, this co-occurrence status will weaken the semantic relationship of tags when it is used to assess the relevance of tags.

As is shown in status 2, the two tags exist in the same user's tag set. The resource's influence on this co-occurrence status is ignored, and the ternary annotation relationship among users, resources, and tags is separated as well. Therefore, this co-occurrence status will also weaken the semantic relationship of tags when it is used to assess the relevance of tags.

As is shown in status 3, the two tags are used by the same user to annotate the same resource. In this status, there are three situations for the two tags to express their semantic relationship. We give three examples to respectively explain the three situations as follows:

Example 1 Suppose tag python $\in T(u_1, r_1)$, tag programming $\in T(u_1, r_1)$, tag python $\in T(u_2, r_1)$ and tag programming $\in T(u_2, r_1)$, which means the two tags are used by different users to annotate the same resource. In this condition, the two tags reflect the unified understanding of different users to the same resource, so the semantic relationship of the two tags is very close.

Example 2 Suppose tag python $\in T(u_1, r_1)$, tag programming $\in T(u_1, r_1)$, tag python $\in T(u_1, r_2)$ and tag programming $\in T(u_1, r_2)$, which means the two tags are used by the same user to annotate different resources. In this condition, the two tags reflect the unified understanding of the same user to different resources, and disclose the two resources with the same features, so the semantic relationship of the two tags is very close.

Example 3 Suppose tag python $\in T(u_1, r_1)$, tag programming $\in T(u_1, r_1)$, tag python $\in T(u_2, r_2)$ and tag programming $\in T(u_2, r_2)$, which means the two tags are used by different users to annotate different resources. In this condition, the two tags reflect the unified understanding of different users to different resources, so the semantic relationship of the two tags is very close.

Obviously, the third tag co-occurrence status contains the ternary annotation relationship completely, and makes the semantic relationship of two tags very close. Therefore, we make use of this tag co-occurrence status and its extension to assess the semantic relevance of tags.

3.3 Spectral clustering

Spectral clustering is a graph-partitioning algorithm. The most widely used objective function to evaluate graph partitions in spectral clustering is the normalized cut (Shi and Malik, 2000) based on 2-way partitioning. Let $G = (V, E, W)$ be an undirected graph, where V is the set of vertices in the graph and $w_{ij} \in W$ is the affinity of edge $e_{ij} \in E$ between vertices $i \in V$ and $j \in V$. Here, the normalized cut objective function can be defined as follows:

$$\mathrm{NCut}(A, B) = \frac{\mathrm{cut}(A, B)}{\mathrm{vol}(A)} + \frac{\mathrm{cut}(A, B)}{\mathrm{vol}(B)}, \quad (1)$$

where A and B represent two different sub-graphs, $\mathrm{vol}(A) = \sum\limits_{i \in A} \sum\limits_{i \sim j} w_{ij}$, and $\mathrm{cut}(A, B) = \sum\limits_{i \in A} \sum\limits_{j \in B} w_{ij}$.

The multi-way normalized cut (Gu *et al.*, 2001) is a transformation of the normalized cut based on k-way partitioning, and its objective function can be defined as follows:

$$\mathrm{MNCut}(A_1, A_2, \ldots, A_k) = \frac{\mathrm{cut}(A_1, \bar{A}_1)}{\sum\limits_{i \in A_1} \sum\limits_{j} w_{ij}} + \frac{\mathrm{cut}(A_2, \bar{A}_2)}{\sum\limits_{i \in A_2} \sum\limits_{j} w_{ij}} + \ldots + \frac{\mathrm{cut}(A_k, \bar{A}_k)}{\sum\limits_{i \in A_k} \sum\limits_{j} w_{ij}}. \quad (2)$$

It is proven that NCut and MNCut are NP-hard problems. For this reason, the Laplace transformation is used in the spectral clustering algorithm to relax the constraints. Let $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W}$ be a Laplacian matrix, where $\boldsymbol{W}$ is the similarity matrix of graph G, $\boldsymbol{D}$ is a diagonal matrix, and $D_{ii} = \sum\limits_{i \sim j} w_{ij}$. After the Laplacian matrix $\boldsymbol{L}$ is normalized, the optimal solutions of NCut and MNCut exist in the vector space composed of the eigenvectors corresponding to the k minimum eigenvalues of $\boldsymbol{L}$.

The key step of spectral clustering is using eigen decomposition technology to deal with the similarity matrix of sample data. The sample data is then aggregated in a new eigenvector space. Compared with the traditional clustering algorithms, the spectral clustering algorithm does not need to transform the data into an n-dimensional vector space. In addition, spectral clustering is cost-insensitive for the irregular or error data, and can be used to process large-scale sparse data. Hence, we choose spectral clustering to aggregate tags in the social tagging data space. The method ensures the integrity of the tag semantic relation, and alleviates the high-dimensionality and sparsity problems because tags need not be represented as a vector space model.

4 Tag co-occurrence spectral clustering method

4.1 Tag co-occurrence

Definition 1 (Tag co-occurrence) Let $S = (U, R, T)$ be a social annotating data set, where $U = \{u_1, u_2, \ldots, u_l\}$ is a set of users, $R = \{r_1, r_2, \ldots, r_m\}$ a set of resources, and $T = \{t_1, t_2, \ldots, t_n\}$ a set of tags. The binary annotating relation set based on users and resources is decided by A_{t_i}, $A_{t_i} = \{< u_p, r_q > | u_p \in U, r_q \in R, t_i \in T, < u_p, r_q, t_i > \in S\}$, and $A_{t_i} \subseteq U \times R$. If A_{t_i} and A_{t_j} have the same element, namely $(A_{t_i} \cap A_{t_j}) \neq \varnothing$, we call tag t_i a co-occurrence with t_j, signed as $\mathrm{Co}(t_i, t_j)$.

4.2 Individual co-occurrence similarity

Definition 2 (Tag individual co-occurrence similarity) Given two tags t_i and t_j, their binary annotating relation sets A_{t_i} and A_{t_j} have the same element. In this situation, we call the two tags t_i and t_j individual co-occurrence. The symbol $\mathrm{sim_{indiv}}(t_i, t_j)$, used to represent the individual co-occurrence similarity between two tags, can be measured by the Jaccard coefficient. Therefore, the tag individual co-occurrence similarity is defined as follows:

$$\mathrm{sim_{indiv}}(t_i, t_j) = \frac{|A_{t_i} \cap A_{t_j}|}{|A_{t_i} \cup A_{t_j}|}, \tag{3}$$

where $|A_{t_i} \cap A_{t_j}|$ is the number of elements in the intersection set of A_{t_i} and A_{t_j}, representing the individual co-occurrence frequency between tags t_i and t_j, and $|A_{t_i} \cup A_{t_j}|$ is the number of elements in the union set of A_{t_i} and A_{t_j}, representing the total using frequency of tags t_i and t_j.

4.3 Common co-occurrence group similarity

Using the individual co-occurrence similarity to aggregate tags may be a good choice, but the individual co-occurrence cannot be competent in disclosing the semantic relationship of tags. In fact, the co-occurrence similarity of tags is deeply influenced by their common co-occurrence group. Here, we first give the concept of the common co-occurrence group.

Given two tags t_i and t_j, t_i co-occurs with t_j, namely $\mathrm{Co}(t_i, t_j)$, and there is a series of tags $t_v, \ldots, t_z$ co-occurring with t_i and t_j, namely $\mathrm{Co}(t_i, t_v)$ and $\mathrm{Co}(t_j, t_v)$, …, $\mathrm{Co}(t_i, t_z)$ and $\mathrm{Co}(t_j, t_z)$. We call tags $t_v, \ldots, t_z$ the common co-occurrence group of t_i and t_j. Now we give two examples to explain the influence of the common co-occurrence group on similarity measure and clustering:

Example 4 Given a tag set $T = \{t_1, t_2, t_3\}$, t_1 co-occurs with t_2, t_1 co-occurs with t_3, while t_2 does not co-occur with t_3. Suppose $|A_{t_1} \cap A_{t_2}| = |A_{t_1} \cap A_{t_3}| \neq 0$, $|A_{t_2} \cap A_{t_3}| = 0$, and $|A_{t_1} \cup A_{t_2}| = |A_{t_1} \cup A_{t_3}| \neq 0$. According to Eq. (3), we find $\mathrm{sim_{indiv}}(t_1, t_2) = \mathrm{sim_{indiv}}(t_1, t_3) \neq 0$ and $\mathrm{sim_{indiv}}(t_2, t_3) = 0$. How to aggregate these tags into two clusters?

The clustering results are shown in Fig. 2.

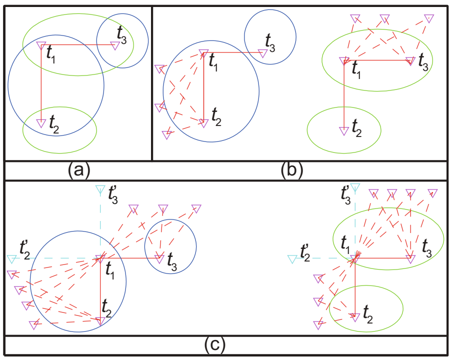

Fig. 2 Clustering results of Example 4: (a) neither (t_1, t_2) nor (t_1, t_3) have a common co-occurrence group; (b) either (t_1, t_2) or (t_1, t_3) have a common co-occurrence group; (c) both (t_1, t_2) and (t_1, t_3) have a common co-occurrence group

If t_1 and t_2 do not have a common co-occurrence group, and t_1 and t_3 also do not have a common co-occurrence group, it is impossible to obtain the unique solution according to their individual co-occurrence similarity, because these tags can be aggregated into cluster $\{t_1, t_2\}$ and cluster $\{t_3\}$, or cluster $\{t_1, t_3\}$ and cluster $\{t_2\}$. The results are shown in Fig. 2a.

If t_1 and t_2 have a common co-occurrence group while t_1 and t_3 do not have a common co-occurrence group, influenced by the common co-occurrence group, the similarity between t_1 and t_2 is much larger than that between t_1 and t_3. Then it is easy to obtain the unique solution. The three tags can be aggregated into cluster $\{t_1, t_2\}$ and cluster $\{t_3\}$. Conversely, the only one clustering result is composed of cluster $\{t_1, t_3\}$ and cluster $\{t_2\}$. The results are shown in Fig. 2b.

If t_1 and t_2 have a common co-occurrence group, and t_1 and t_3 have a common co-occurrence group, the only one clustering result can be decided by comparing the influences of the two common co-occurrence groups. The results are shown in Fig. 2c.

Example 5 Considering an extreme situation, for a tag set $T = \{t_1, t_2\}$, if $(|A_{t_1} \cap A_{t_2}| = |A_{t_2}|) << (|A_{t_1} \cup A_{t_2}| = |A_{t_1}|)$, can the two tags be aggregated into the same cluster?

According to Eq. (3), the individual co-occurrence similarity between t_1 and t_2 can be calculated, namely, $\mathrm{sim}_{\mathrm{indiv}}(t_1, t_2) = \dfrac{|A_{t_1} \cap A_{t_2}|}{|A_{t_1}|} \cong 0$. Obviously, t_1 and t_2 cannot be aggregated into the same cluster. In fact, the two tags may have a hierarchical relation in concepts. For instance, tag t_1 is ‘programming’, and tag t_2 could be a more professional concept ‘python’. In this situation, using individual co-occurrence similarity cannot make a correct judgment. However, we can use the common co-occurrence group to describe their semantic relevance, and decide whether the two tags can be aggregated into the same cluster by calculating their common co-occurrence group similarity.

Definition 3 (Tag common co-occurrence group similarity) Given two tags t_i and t_j, and $\mathrm{Co}(t_j, t_j)$, let $C = \{t_y \,|t_y \in T\}$ be a common co-occurrence group tag set of t_i and t_j, and $C \subseteq T$. We use t_c^k to represent the kth tag in C, and symbol $\mathrm{sim}_{\mathrm{group}}(t_i, t_j)$ to represent the common co-occurrence group similarity between t_i and t_j. The common co-occurrence group similarity between the two tags can be measured by an improved Jaccard coefficient. The tag common co-occurrence group similarity is defined as follows:

$$\mathrm{sim}_{\mathrm{group}}(t_i, t_j) = \frac{1}{|C|} \sum_{k=1}^{|C|} \frac{|(A_{t_i} \cup A_{t_c^k}) \cap (A_{t_j} \cup A_{t_c^k})|}{|(A_{t_i} \cup A_{t_c^k}) \cup (A_{t_j} \cup A_{t_c^k})|}. \tag{4}$$

The physical interpretation of Eq. (4) is that a group of co-occurrence tags are introduced to measure the similarity between two tags. Groups of these tags maintain the very strong semantic relationship. Thus, the differentiation between two tags is improved.

4.4 Combinatorial co-occurrence similarity

In the above discussion, Eqs. (3) and (4) provide two methods to calculate similarity on different aspects. Here, we propose a combination method to address the similarity between tags.

Definition 4 (Tag combination co-occurrence similarity) The symbol $\mathrm{sim}_{\mathrm{comb}}(t_i, t_j)$ is used to represent the combinatorial co-occurrence similarity. Therefore, the tag combinatorial co-occurrence similarity is defined as follows:

$$\mathrm{sim}_{\mathrm{comp}}(t_i, t_j) = \lambda \cdot \mathrm{sim}_{\mathrm{group}}(t_i, t_j) + (1-\lambda) \cdot \mathrm{sim}_{\mathrm{indiv}}(t_i, t_j), \tag{5}$$

where λ ($0 \le \lambda \le 1$) is a weight coefficient. Usually, as an equilibrium coefficient, λ is set to 0.5, which means the individual co-occurrence similarity and the common co-occurrence group similarity are equally important. Eq. (5) can be used to calculate the common co-occurrence group similarity when $\lambda = 1$, and the individual co-occurrence similarity when $\lambda = 0$.

4.5 Tag co-occurrence spectral clustering algorithm

In this subsection, we use a normalized spectral clustering algorithm to cluster tags (Ng *et al.*, 2002). We need to preprocess the data.

Given a social tagging dataset $S = (U, R, T)$, we need to construct a binary annotating relation set A_t for each tag. Then we construct a dataset $A_T = \{A_{t_1}, A_{t_2}, \ldots, A_{t_n}\}$, which is the input of our algorithm. The tag spectral clustering algorithm based on common co-occurrence group similarity is shown in Algorithm 1.

In Algorithm 1, $\boldsymbol{W}$ is the similarity matrix of tags, $\boldsymbol{D}$ the diagonal matrix of tags derived from $\boldsymbol{W}$, $\boldsymbol{L}$ the unnormalized Laplacian matrix of tags derived from $\boldsymbol{D}$ and $\boldsymbol{W}$, $\boldsymbol{L}'$ the normalized Laplacian matrix of tags derived from $\boldsymbol{D}$ and $\boldsymbol{L}$, $\boldsymbol{E}$ the eigenvector matrix derived from $\boldsymbol{L}'$, and $\boldsymbol{R}$ the normalized matrix of $\boldsymbol{E}$.

Algorithm 1 Tag spectral clustering based on common co-occurrence group similarity

```
Require: Dataset $A_T$, the number of clusters $k$
Ensure: Clusters $T_1, T_2, \ldots, T_k$
1: Generate the tag similarity matrix $\boldsymbol{W} \in \boldsymbol{\Phi}^{n\times n}$ from $A_T$ based on Eq. (4) and set $W_{ii} = 0$.
2: Compute the unnormalized Laplacian matrix $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W}$, where $\boldsymbol{D}$ is a diagonal matrix satisfying $D_{ii} = \sum_{ij} w_{ij}$.
3: Compute the normalized Laplacian matrix $\boldsymbol{L}' = \boldsymbol{D}^{-1/2}\boldsymbol{L}\boldsymbol{D}^{1/2}$.
4: Compute the first $k$ eigenvectors of $\boldsymbol{L}'$, i.e., $\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_k$.
5: Let $\boldsymbol{E} \in \boldsymbol{\Phi}^{n\times k}$ be the matrix containing vectors $\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_k$ of $\boldsymbol{L}'$ as columns.
6: Form the matrix $\boldsymbol{R} \in \boldsymbol{\Phi}^{n\times k}$ from $\boldsymbol{E}$ by normalizing the rows to norm 1, which is set as $r_{ij} = e_{ij}/(\sum_k e_{ik}^2)^{1/2}$.
7: For $i = 1, 2, \ldots, n$, let $\boldsymbol{t}_i \in \boldsymbol{\Phi}^k$ be the vector corresponding to the $i$th row of $\boldsymbol{R}$.
8: Cluster the points $(t_i)_{i=1,2,\ldots,n}$ with the $K$-means algorithm into clusters $T_1, T_2, \ldots, T_k$.
9: Return $T_1, T_2, \ldots, T_k$.
```

The time complexity of our method is related mainly to steps 1, 4, and 8. For step 1, the time complexity of the similarity matrix construction is $O(n^2)$. For step 4, the time cost of the eigenvector computation is $O(kn^2)$. For step 8, the time cost of the K-means algorithm is $O(tkn)$, where t is the number of iterations. In general, the time complexity of our method is acceptable.

5 Experimental evaluations

5.1 Experimental dataset

Our experiment was conducted on CiteULike, which is a famous academic paper-sharing system. Users in CiteULike can freely annotate their own or others' papers, and create their collections of articles. CiteULike has a very large amount of data and provides some kinds of datasets for researchers, such as 'Who-posted-what data', from which we obtain a subset. After data preprocessing, our dataset contains 4186 users, 1028 resources, and 6905 tags. The experiments were carried out using a workstation with an Intel Core i5 3470 CPU (3.2 GHz) and an 8 GB memory, running Windows 7 (64 bit). All the algorithms were implemented in Matlab 2010. Table 1 displays the statistics of our dataset.

Table 1 Statistics of the experimental dataset CiteULike

Parameter	Value
Number of users	4186
Number of resources	1028
Number of tags	6905
Number of annotations	29 048
Average number of annotations per user	7
Average number of annotations per resource	28

5.2 Evaluation criteria

Evaluation of clustering results can be viewed as a cluster validation. We cannot use precision or recall to assess clustering results, because there are no class labels or external benchmarks in our dataset. In this subsection, we introduce two internal valuations to assess the effectiveness of the clustering result. We first introduce the Silhouette coefficient (SC) (Kaufman and Rousseeuw, 2008), which is a famous metric, to evaluate the clustering result. Next, we introduce the Dunn index (Dunn, 1974) as another way to assess our results.

5.2.1 Silhouette coefficient

Kaufman and Rousseeuw (2008) introduced the SC to assess clustering results. For any tag t_i in clustering results, its SC (represented as $\mathrm{SC}(t_i)$) is calculated as follows:

$$\mathrm{SC}(t_i) = \frac{b(t_i) - a(t_i)}{\max\{a(t_i), b(t_i)\}}, \tag{6}$$

$$\mathrm{SC}(C) = \left(\sum \mathrm{SC}(t_i)\right) / \mathrm{Size}(C), \quad t_i \in C, \tag{7}$$

$$\mathrm{SC}(D) = \left(\sum \mathrm{SC}(t_i)\right) / \mathrm{Size}(D), \quad t_i \in D. \tag{8}$$

Here, we hypothesize that tag t_i is in cluster A. In Eq. (6), $a(t_i)$ is the average distance between t_i and any other tag in cluster A, and $b(t_i)$ is the minimum average distance between t_i and any other cluster B. How can the distance between the two tags be measured? In our approach, we use similarity to assess how near the two tags are. Therefore, the dissimilarity can be used to measure the distance between two tags. Let $\mathrm{sim}(t_i, t_j)$ be the similarity between tags t_i and t_j, so the distance between tags t_i and t_j can be measured by $1 - \mathrm{sim}(t_i, t_j)$. When tag t_i belongs to a singleton cluster, $a(t_i)$ cannot be calculated. According to Eq. (6), $\mathrm{SC}(t_i)$ also cannot be calculated. In this situation, we set any value $v \in [-1, 1]$ as SC for these tags according to applications.

Singleton clusters do not have a semantic structure because there is only one tag in them. To ensure the integrity of our dataset, in this paper, we set $\mathrm{SC}(t_i) = 0$ when the clustering result of tag t_i is a singleton cluster.

If we want to obtain the SC for every tag in a cluster, we can use Eq. (7) to calculate it, which is the average SC of all tags in cluster C. Likewise, Eq. (8) is used to calculate the SC of the clustering result, which is the average SC of all tags. In this study, the average SC of all tags calculated by Eq. (8) is used as an evaluation criterion.

The SC confirms the clustering performance based on the pairwise difference of between-culster and within-cluster distances. Obviously, clusters with higher SC values tend to be good clusters.

5.2.2 Dunn index

Dunn (1974) introduced the Dunn index to recognize well-separated and dense clusters. The Dunn index is the ratio between the minimum inter-cluster distance and maximum intra-cluster distance:

$$\mathrm{Dunn} = \min_{1\le i\le |C|} \min_{1\le j\le |C|} \frac{d(C_i, C_j)}{\max\limits_{1\le k\le |C|} d'(C_k)}, \tag{9}$$

where $|C|$ is the number of clusters, $d(C_i, C_j)$ represents the distance between clusters C_i and C_j, and $d(C_i, C_j) = \min\limits_{x\in C_i, y\in C_j} \{d(x, y)\}$. $d'(C_k)$ represents the intra-cluster distance of cluster C_k, and $d'(C_k) = \max\limits_{x,y\in C_k} \{d(x, y)\}$.

The internal valuation criterion looks for clusters with high intra-cluster similarity and low inter-cluster similarity. Therefore, the higher the Dunn value is, the better the clustering result is.

5.3 Experiments and discussion

5.3.1 Comparisons of spectral clustering results with different similarities

To evaluate the effectiveness of our method, we conducted several experimental comparisons on our dataset. We first compared the spectral clustering results obtained by using different tag co-occurrence similarity measures mentioned in Section 4. Silhouette coefficients and Dunn values are shown in Figs. 3 and 4, respectively. Here, the cluster number (k) is 40, 60, 80, 100, or 120. As introduced in Section 4.4, when $\lambda = 0$, Eq. (5) is equivalent to Eq. (3), and they represent the individual co-occurrence similarity; when $\lambda = 1$, Eq. (5) is equivalent to Eq. (4), and they represent the common co-occurrence group similarity; when $0 \le \lambda \le 1$, Eq. (5) represents the combinatorial co-occurrence similarity.

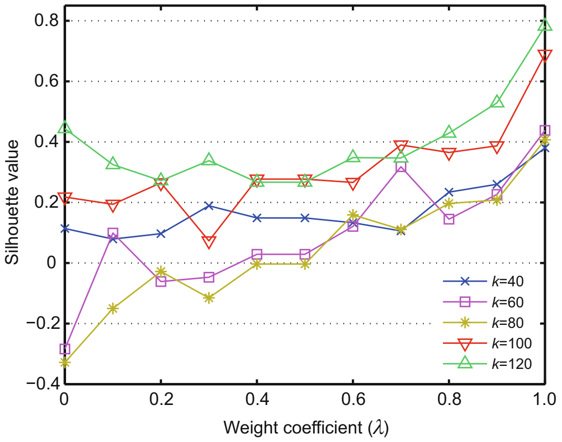

Fig. 3 Silhouette coefficient comparisons by using different tag co-occurrence similarities in the spectral clustering algorithm

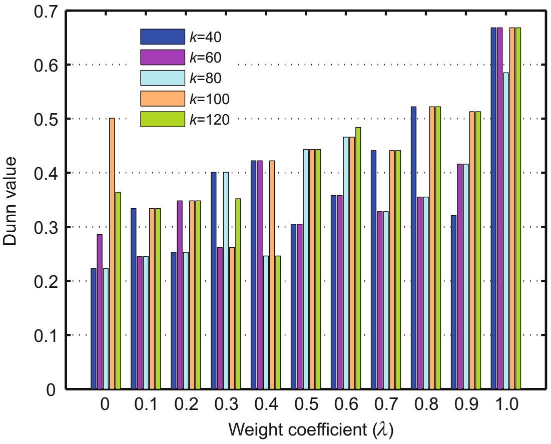

Fig. 4 Dunn comparisons by using different tag co-occurrence similarities in the spectral clustering algorithm

As shown in Figs. 3 and 4, generally the SC values and Dunn values increase with the increase of λ. The results denote that, compared with the individual co-occurrence similarity and the combinatorial co-occurrence similarity, the spectral clustering method based on common co-occurrence group similarity has better performance and provides better clustering results. In our opinion, the most basic reason is that individual co-occurrence similarity cannot address enough semantic features in the context of a large amount of annotating data. In Fig. 3, the SC value at some point is negative, which means that no meaningful cluster structure is found in this situation.

5.3.2 Processing of other contrastive clustering methods

Tag clustering contains two important tasks: one is the relevance assessment of tags, namely acquiring the similarity of tags, and the other is using a clustering algorithm to produce clusters. As analyzed in Sections 1 and 2, the similarity measure methods such as VSM and tag co-occurrence will lose some semantic relationship between tags if they are based only on resources or users, and traditional clustering algorithms such as K-means and agglomerative hierarchical clustering cannot tackle the annotating data of arbitrary shape. To verify the validity of our method, we implement K-means and agglomerative hierarchical clustering with different similarity measures, and then compare their performance with that of our method.

We first use VSM to measure the similarity between tags. Two approaches can be used to measure the similarity between tags based on VSM. One is based on resources, and the other on users. Given two tags t_i and t_j, for each tag, we use a weight vector based on resources to represent it, namely $\boldsymbol{t}_i = [t_{ir_1}, t_{ir_2}, \ldots, t_{ir_m}]$ and $\boldsymbol{t}_j = [t_{jr_1}, t_{jr_2}, \ldots, t_{jr_m}]$. Here, we use the TF-IDF formula to calculate the values of weights. Then we calculate the Euclidean distance of the two vectors t_i and t_j. The distance represents the dissimilarity between tags t_i and t_j. The same way is used for user VSM.

We also use resource co-occurrence or user co-occurrence to measure the similarity between tags. For two tags t_i and t_j, we count the number of individual occurrences and co-occurrences of t_i and t_j in all resources. Then we use the Jaccard coefficient to calculate the similarity between tags t_i and t_j. The same approach is used for user co-occurrence.

Based on the similarity of all tags, the similarity matrix is constructed. For the K-means algorithm, we directly run it on the similarity matrix to generate the clustering results. For the agglomerative hierarchical clustering algorithm, one problem must be solved; that is, how the distance between two clusters is measured. Several approaches, such as single-linkage, complete-linkage, and average-linkage, can be adopted. In this paper, the shortest distance between objects in the two clusters is used to measure its distance. Namely, we use the single-linkage clustering algorithm to cluster tags. For K-means or hierarchical clustering, clustering results may contain some singleton clusters. As mentioned in Section 5, we set the SC value of these singleton clusters as 0.

5.3.3 Comparisons of results obtained by other clustering methods based on VSM with our method

Figs. 5 and 6 show the SC values and Dunn values of K-means based on resource VSM, hierarchical clustering based on resource VSM, K-means based on user VSM, hierarchical clustering based on user VSM, and our method, respectively.

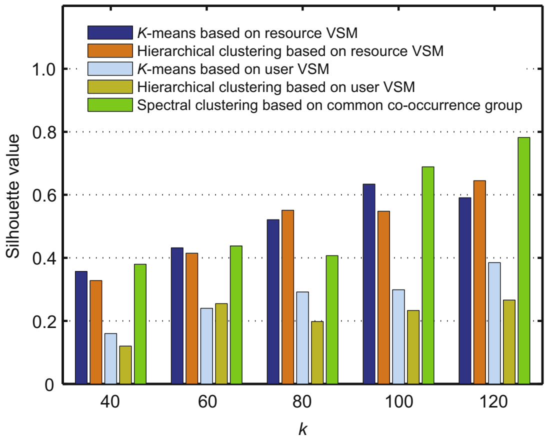

Fig. 5 Silhouette coefficient comparisons by four clustering methods based on VSM with our method

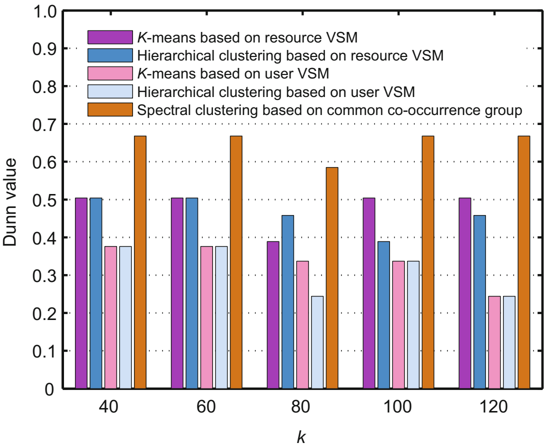

Fig. 6 Dunn comparisons by four clustering methods based on VSM with our method

5.3.4 Comparisons of results obtained by other clustering methods based on tag co-occurrence with our method

Figs. 7 and 8 show the SC values and Dunn values of K-means based on resource co-occurrence, hierarchical clustering based on resource co-occurrence, K-means based on user co-occurrence, hierarchical clustering based on user co-occurrence, and our method, respectively.

5.3.5 Discussion of experimental results and summary

By examining the experimental results of Figs. 5–8, we can see that the quality of clustering results obtained by our spectral clustering method is considerably better than that of K-means and hierarchical clustering with respect to the SC and Dunn measures. The reason is that our proposed method can not only maintain the strongest semantic relationship of tags but also capture the global semantic information of tags through the common co-occurrence group.

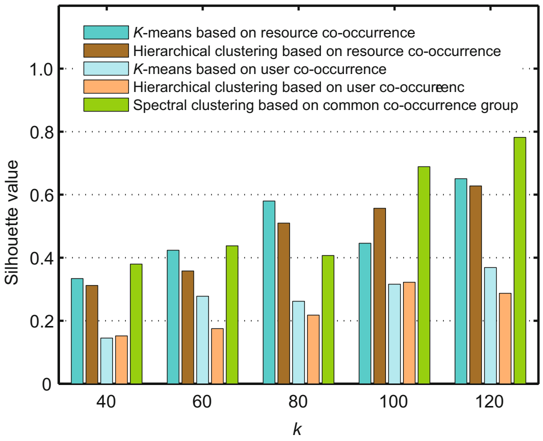

Fig. 7 Silhouette coefficient comparisons by four clustering methods based on tag co-occurrence with our method

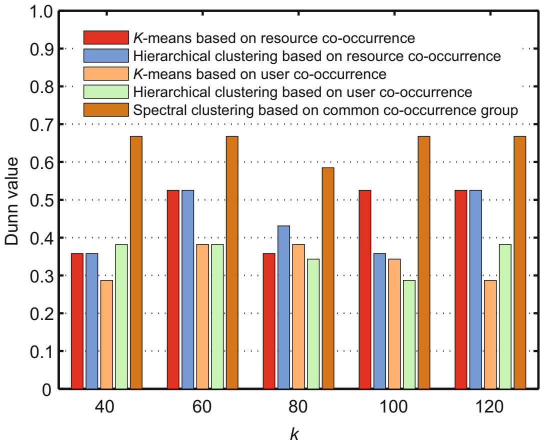

Fig. 8 Dunn comparisons by four clustering methods based on tag co-occurrence with our method

From Figs. 5 and 7, we can see that when the cluster number of k is 80, the SC value obtained by our spectral clustering method is lower than those of K-means and hierarchical clustering based on resource VSM or resource co-occurrence. Observing the inner structure of clustering results, we find that our method generates more singleton clusters than K-means and hierarchical clustering based on resource VSM or resource co-occurrence, which may be the reason why our method achieves a poor performance when $k = 80$.

From Figs. 5 and 6, we find that the clustering results obtained by K-means and hierarchical clustering based on resource VSM are in a similarly poor range. We also see that the clustering results obtained by K-means and hierarchical clustering based on user VSM are in a similarly poor range. The same situations exist in Figs. 7 and 8. Why do the two traditional clustering algorithms produce a similarly poor performance? The reason is that they have the same defects on high-dimensional and large datasets, no matter which method is used to measure the similarity.

From Figs. 5 and 6, the values of the SC and Dunn reveal that the similarity measure method based on user VSM produces the worst performance compared with the method based on resource VSM and our method. From Figs. 7 and 8, the similarity measure approach based on user co-occurrence also produces the worst performance compared with the approach based on resource co-occurrence and our approach. According to the statistics of our dataset in Table 1, we find that the number of users is around 4 times that of the resources. In other words, the user dimension is much larger than that of the resources. We also find that the number of average annotations per user is 1/4 that of the average annotations per resource. When the user VSM or the user co-occurrence is used to assess the semantic relevance of tags, the problem of sparsity or the lack of semantics is more serious than in resource VSM or co-occurrence. Therefore, for our dataset, the semantic relevance evaluation methods based on user VSM or co-occurrence cannot measure the similarity of tags accurately, which results in the clustering results obtained by user VSM or user co-occurrence being the worst. However, our spectral clustering method based on the common co-occurrence group alleviates the above problems, and can generate meaningful cluster structures.

To sum up, the comparisons of the results under different clustering methods are given in Table 2.

6 Conclusions

It is well known that tag clustering can help to find interesting tag clusters embedded in tagging datasets. We used the common co-occurrence group similarity to measure the relevance of tags based on tag co-occurrence, which not only keeps the strongest semantic relationship of tags, but also captures the global semantic information. To support relevance assessment, we first improved the Jaccard coefficient based on the common co-occurrence group to calculate the similarity of tags. After that, we proposed a spectral clustering method for tag clustering, which can alleviate some problems such as the lack of semantic, high dimensionality, and sparsity.

We implemented our relevance measure on a real world dataset from CiteULike, and compared our tag spectral clustering method with other different clustering approaches. Experimental results showed that our method achieves good performance in tag clustering. In this paper, we took into account only the ternary annotation information among users, resources, and tags to assess the relevance between tags. In future work, we will combine the ternary annotation information with resource's contents or user's characteristics to evaluate the relevance of tags.

References

Begelman, G., Keller, P., Smadja, F., 2006. Automated tag clustering: improving search and exploration in the tag space. Proc. 15th Int. World Wide Web Conf., p.15-33.

Bischoff, K., Firan, C.S., Nejdl, W., *et al.*, 2008. Can all tags be used for search? Proc. 17th ACM Conf. on Information and Knowledge Management, p.193-202. http://dx.doi.org/10.1145/1458082.1458112

Cui, J.W., Liu, H.Y., He, J., *et al.*, 2011. TagClus: a random walk-based method for tag clustering. *Knowl. Inform. Syst.*, **27**(2):193-225. http://dx.doi.org/10.1007/s10115-010-0307-y

Cuzzocrea, A., 2006. Combining multidimensional user models and knowledge representation and management techniques for making web services knowledge-aware. *Web Intell. Agent Syst.*, **4**(3):289-312.

Cuzzocrea, A., Mastroianni, C., 2003. A reference architecture for knowledge management-based web systems. Proc. 4th Int. Conf. on Web Information Systems Engineering, p.347-351. http://dx.doi.org/10.1109/WISE.2003.1254509

Dattolo, A., Eynard, D., Mazzola, L., 2011. An integrated approach to discover tag semantics. Proc. ACM Symp. on Applied Computing, p.814-820. http://dx.doi.org/10.1145/1982185.1982359

Deutsch, S., Schrammel, J., Tscheligi, M., 2011. Comparing different layouts of tag clouds: findings on visual perception. *Human Aspects Visual.*, **6431**:23-37. http://dx.doi.org/10.1007/978-3-642-19641-6_3

Dunn, J.C., 1974. Well-separated clusters and optimal fuzzy-partitions. *J. Cybern.*, **4**(1):95-104. http://dx.doi.org/10.1080/01969727408546059

Furnas, G.W., Fake, C., von Ahn, L., *et al.*, 2006. Why do tagging systems work? Proc. Extended Abstracts on Human Factors in Computing Systems, p.36-39. http://dx.doi.org/10.1145/1125451.1125462

Gemmell, J., Shepitsen, A., Mobasher, B., *et al.*, 2008. Personalizing navigation in folksonomies using hierarchical

Table 2 Comparisons of results under different clustering methods

Clustering algorithm	Similarity measure method	SC/Dunn				
		$k = 40$	$k = 60$	$k = 80$	$k = 100$	$k = 120$
K-means algorithm	Tag's resource VSM	0.357/0.504	0.432/0.504	0.521/0.389	0.634/0.504	0.591/0.504
	Tag's user VSM	0.160/0.376	0.240/0.376	0.292/0.337	0.299/0.337	0.385/0.244
	Tag's resource co-occurrence	0.334/0.358	0.424/0.525	0.580/0.358	0.446/0.525	0.651/0.525
	Tag's user co-occurrence	0.145/0.287	0.278/0.382	0.262/0.382	0.316/0.343	0.369/0.287
Agglomerative hierarchical clustering algorithm	Tag's resource VSM	0.328/0.504	0.415/0.504	0.551/0.458	0.548/0.389	0.645/0.458
	Tag's user VSM	0.120/0.376	0.255/0.376	0.198/0.244	0.233/0.337	0.266/0.244
	Tag's resource co-occurrence	0.312/0.358	0.358/0.525	0.510/0.431	0.557/0.358	0.628/0.525
	Tag's user co-occurrence	0.152/0.382	0.175/0.382	0.218/0.343	0.322/0.287	0.287/0.382
Spectral algorithm	Tag's common co-occurrence group	**0.380/0.668**	**0.438/0.668**	**0.407/0.585**	**0.689/0.668**	**0.782/0.668**

tag clustering. Proc. 10th Int. Conf. on Data Warehousing and Knowledge Discovery, p.196-205. http://dx.doi.org/10.1007/978-3-540-85836-2_19

Gu, M., Zha, H., Ding, C., *et al.*, 2001. Spectral relaxation models and structure analysis for k-way graph clustering and bi-clustering. Available from http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.10.2657 [Accessed on Apr. 5, 2015].

Heymann, P., Garcia-Molina, H., 2006. Collaborative creation of communal hierarchical taxonomies in social tagging systems. Technical Report, No. 2006-10, Stanford University, USA.

Isabella, P., 2009. Folksonomies. Indexing and Retrieval in Web 2.0. Walter de Gruyter, Berlin. http://dx.doi.org/10.1515/9783598441851

Jiang, J.J., Conrath, D.W., 1997. Semantic similarity based on corpus statistics and lexical taxonomy. Proc. Int. Conf. of Research on Computational Linguistics, p.1-15.

Kaufman, L., Rousseeuw, P.J., 2008. Finding Groups in Data: an Introduction to Cluster Analysis. John Wiley & Sons, London, UK. http://dx.doi.org/10.1002/9780470316801

Knautz, K., Soubusta, S., Stock, W.G., 2010. Tag clusters as information retrieval interfaces. Proc. 43rd Hawaii Int. Conf. on System Sciences, p.1-10. http://dx.doi.org/10.1109/HICSS.2010.360

Laniado, D., Eynard, D., Colombetti, M., 2007. Using WordNet to turn a folksonomy into a hierarchy of concepts. Proc. 4th Italian Semantic Web Workshop on Semantic Web Application and Perspectives, p.192-201.

Lehwark, P., Risi, S., Ultsch, A., 2008. Visualization and clustering of tagged music data. Proc. 31st Annual Conf. on Data Analysis, Machine Learning and Applications, p.673-680. http://dx.doi.org/10.1007/978-3-540-78246-9_79

Markines, B., Cattuto, C., Menczer, F., *et al.*, 2009. Evaluating similarity measures for emergent semantics of social tagging. Proc. 18th Int. Conf. on World Wide Web, p.641-650. http://dx.doi.org/10.1145/1526709.1526796

Marlow, C., Naaman, M., Boyd, D., *et al.*, 2006. HT06, tagging paper, taxonomy, Flickr, academic article, to read. Proc. 17th Conf. on Hypertext and Hypermedia, p.31-40. http://dx.doi.org/10.1145/1149941.1149949

Mathes, A., 2004. Folksonomies—cooperative classification and communication through shared metadata. Available from http://www.adammathes.com/academic/computer-mediated-communication/folksonomies.html [Accessed on Apr. 5, 2015].

Michlmayr, E., Cayzer, S., 2007. Learning user profiles from tagging data and leveraging them for personal(ized) information access. Proc. 16th Int. World Wide Web Conf., p.1-7.

Ng, A.Y., Jordan, M.I., Weiss, Y., 2002. On spectral clustering: analysis and an algorithm. Proc. Conf. Advances in Neural Information Processing Systems, p.849-856.

Noll, M.G., Meinel, C., 2007. Web search personalization via social bookmarking and tagging. Proc. 6th Int. Semantic Web Conf. and 2nd Asian Semantic Web Conf. on the Semantic Web, p.367-380. http://dx.doi.org/10.1007/978-3-540-76298-0_27

Noruzi, A., 2006. Folksonomies: (un)controlled vocabulary? *Knowl. Organ.*, **33**(4):199-203.

Rivadeneira, A.W., Gruen, D.M., Muller, M.J., *et al.*, 2007. Getting our head in the clouds: toward evaluation studies of tagclouds. Proc. SIGCHI Conf. on Human Factors in Computing Systems, p.995-998. http://dx.doi.org/10.1145/1240624.1240775

Salton, G., 1983. Introduction to Modern Information Retrieval. McGraw-Hill College, New York, USA. http://dx.doi.org/10.1016/0306-4573(83)90062-6

Shepitsen, A., Gemmell, J., Mobasher, B., *et al.*, 2008. Personalized recommendation in social tagging systems using hierarchical clustering. Proc. ACM Conf. on Recommender Systems, p.259-266. http://dx.doi.org/10.1145/1454008.1454048

Shi, J., Malik, J., 2000. Normalized cuts and image segmentation. *IEEE Trans. Patt. Anal. Mach. Intell.*, **22**(8):888-905. http://dx.doi.org/10.1109/34.868688

Shirky, C., 2004. Folksonomy. Available from http://www.corante.com/many/archives/2004/08/25/-folksonomy.php [Accessed on Apr. 5, 2015].

Simpson, E., 2008. Clustering tags in enterprise and web folksonomies. Proc. Int. Conf. on Weblogs and Social Media, p.222-223.

Suchanek, F.M., Vojnovic, M., Gunawardena, D., 2008. Social tags: meaning and suggestions. Proc. 17th ACM Conf. on Information and Knowledge Management, p.223-232. http://dx.doi.org/10.1145/1458082.1458114

Szomszor, M., Cattuto, C., Alani, H., *et al.*, 2007. Folksonomies, the Semantic Web, and Movie Recommendation. Proc. 4th European Semantic Web Conf., p.71-84.

Van Damme, C., Hepp, M., Siorpaes, K., 2007. Folksontology: an integrated approach for turning folksonomies into ontologies. Proc. Workshop on Bridging the Gap Between Semantic Web and Web2.0, p.57-70.

Vanderlei, T.A., Durão, F.A., Martins, A.C., *et al.*, 2007. A cooperative classification mechanism for search and retrieval software components. Proc. ACM Symp. on Applied Computing, p.866-871. http://dx.doi.org/10.1145/1244002.1244192

Vander Wal, T., 2004. Folksonomy. Available from http://vanderwal.net/essays/051130/folksonomy.pdf [Accessed on Apr. 5, 2015].

Vandic, D., van Dam, J.W., Hogenboom, F., *et al.*, 2011. A semantic clustering-based approach for searching and browsing tag spaces. Proc. ACM Symp. on Applied Computing, p.1693-1699. http://dx.doi.org/10.1145/1982185.1982538

Xu, G.D., Zong, Y., Jin, P., *et al.*, 2015. KIPTC: a kernel information propagation tag clustering algorithm. *J. Intell. Inform. Syst.*, **45**(1):95-112. http://dx.doi.org/10.1007/s10844-013-0262-7